A Review of Clustering Schemes for Mobile Ad Hoc Networks

Fraol Bekana, J. Mehedi

Abstract- Due to its vast application, maintaining the connectivity and forwarding the information in mobile ad hoc network (MANET) is very crucial to increase the efficiency as well as the performance of the system. One way of guarantying this performance to a large and dynamic network is through clustering. A number of researchers came up with a variety of approaches and performance metrics for ad hoc clustering. In this paper, we have presented a comprehensive review of various proposed clustering schemes for MANET. The classification and analysis of these schemes are done depending on their cluster formation. Descriptions of their approaches, evaluations of their performance, discussions of their advantages and disadvantages of each clustering schemes are presented. We believe that this paper will enable readers to get more understanding of ad hoc clustering and indicate research trends in the area.

Key words: Clustering, CDS, Mobile ad hoc Networks

I. INTRODUCTION

Advancements in wireless technology now a days makes our day to day life more easier and simpler than ever. Mobile ad hoc network (MANET) is one of the major contributor to this achievement as it enables us to get communicate at anytime anywhere unlike infrastructure-based networks. Thus, MANET is a dynamic, self-organizing and infrastructure less network. Each node can act as a router to receive and/or forward packets as well as all nodes can randomly move around, leaving or joining the network at any time. Due to these fact, it bears many applications in areas like emergency service, battle fields, group of researchers in forest, group of mountain climbers, rescue operations and so on. Thus, maintaining connectivity and forwarding the information in MANET is very crucial to increase the efficiency as well as the performance of the system. In MANET, communication between mobile nodes might be done in a multi-hop fashion due to transmission power limitation and radio channel utilization. In this case, several intermediate hosts participate in relaying the packets to its destination node. In order to transfer packets to all the nodes, broadcasting is the easiest and most frequently used one. Special attention is needed here not to overload the network with traffic jam while conveying the information to all the nodes. Otherwise, the network might be field with some redundant broadcasting packets resulting in congestion and packet drop due to collision. These will consume the limited battery power and available bandwidth of the mobile nodes. All these problems are called “broadcast storm problem” [1]. Broadcasting storm problem can be alleviated either by reducing the number of rebroadcasting hosts or differentiate the timing of rebroadcasting [20].

Revised Version Manuscript Received on May 21, 2015.

Fraol Bekana, Electronics and Communication Engineering, National Institute of Technology, Silchar, India.

Dr. Jeshan Mehedi, Electronics and Communication Engineering, National Institute of Technology, Silchar, India.

The latter is an NP-hard problem to assign an optima slot to each node in the network which also requires timing synchronization [1]. The former can be achieved through clustering which limits the number of nodes involved in forwarding the packets through the entire network. This will inhibit some of the nodes from forwarding the packets in which they become only passive receivers. Clustering is partitioning a larger and complex network into many smaller and simpler virtual groups called clusters. This enhances the performance of the network by making the network management more easier and simpler. It also helps to propagate data packets at a faster rate through the network thereby minimizing the transmission delay. Consequently, many researchers focus on presenting an efficient and effective ways of clustering for MANETs. As far as we know, there have been no in depth reviews of MANET clustering issues presented recently. In this paper, we have presented a comprehensive review of various proposed clustering schemes for MANET. We have classified and analyzed these schemes based on their cluster formation.

A. Clustering

As stated above, clustering is a process of partitioning the entire bigger network into smaller and scalable groups called clusters. Within each cluster, the members choose one node to be a boss to manage the broadcasting and other intra-cluster and inter-cluster activities. These selected nodes, bosses, are connected together to form a connected virtual backbone (CVB) for the network. Many researchers follow different bases for clustering the network into different groups. Although the distance between the nodes is the most factor, there are also other rules and factors to be considered. Based

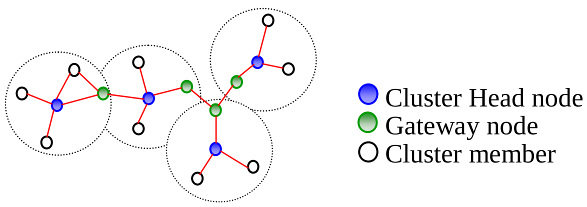


Fig. 1. Typical Clustering of the network into different cluster groups

on these factors and rules, a larger network can be grouped into different isolated clusters as shown in fig. 1.

The above network is divided into different cluster groups depending on some rules as indicated by dotted circles. In a clustered network, the nodes function may vary depending on their positions and other factors. Clusterhead is a group boss and is used to monitor packet forwarding in the cluster. Gateway node is a non-clusterhead node used to perform inter-cluster packet forwarding in the network. Gateway nodes are used to connect two neighbouring clusterheads either directly or indirectly through other intermediate

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication Pvt. Ltd.

nodes. Cluster members are ordinary nodes which are not involved in packet forwarding. They send and/or receive messages through their clusterhead. They do not have any connection by themselves with other clusters.

B. Why Clustering?

A clustering architecture in mobile ad-hoc network is crucial for many reasons. Firstly, it helps to propagate data packets in the network at a faster rate than ever. This is because of the reduced number of nodes participating in packet forwarding. Only clusterheads and gateway nodes are responsible while ordinary nodes are not required. This will increase the performance of the network by mitigating the redundancy in re-broadcasting which in turn reduces collision while increasing network bandwidth. Secondly, it provides easier network management and maintenance as the larger whole network is represented by a newly reduced and smaller virtual backbone network. This will increase the scalability of the network as well. Moreover, clustering creates an environment for increasing the capacity of the system by spatial frequency reuse mechanism [2], [3], [12], [13]. In a multi-clustered network structure, two or more non-overlapping clusters can use the same frequency or code set provided that there is no frequency interference between them [4]. Lastly, in the case of some link failure, there is no need for the whole network to participate on it but locally fixed reducing significantly the network overhead.

C. The Cost of Clustering

Although clustering increases the network performance, it is not easy to cluster a network in to different groups. Because constructing and maintaining cluster in a dynamic environment is more complex and expensive than simply flat-based MANETs. It needs an explicit message exchange between two mobile nodes consuming a significant network bandwidth [2]. This also drains the limited battery power of a mobile node resulting in a network disconnection. This may create havoc in sensitive areas like military and/or rescue operations where connectivity is immiscible. Therefore, unlike the flat structure, the maintenance cost, the unique control message exchange, the ripple effect of re-clustering, the message complexity, the stationary assumption for cluster formation, and the time complexity required are the main costs of a cluster-based MANET [5].

Table 1. Description of cost terms for clustered-based MANET

Cost of Clustering	Description
Maintenance cost	In a large and dynamic network where frequent cluster head exchange or link failure is common, a significant amount of information exchange is required to maintain connectivity. This is critical in MANET which leads to a huge network bandwidth and power consumption in mobile nodes.
Unique control message for clustering	Unlike data packets or routing information exchange, cluster formation requires a unique clustering-related information exchange between mobile node pairs. This is because of the fact that clusters cannot be formed or maintained by non-clustering-related messages which in turn increases the cost.
Ripple effect of re-clustering	Re-clustering due to node mobility and/or link failure of a single cluster head may affect the stability of its neighboring clusters and hence expands to the whole network. This is critical for network topology change which in turn consumes node's energy and bandwidth degrading its performance.
Message complexity	Clustering requires a significant amount of explicit message exchange in cluster formation phase. The overall and unique clustering-related message exchanged during clustering is called message complexity. The message complexity of re-clustering in maintenance of networks with ripple-effect is equal to that of cluster formation phase while much lower for the one with no ripple effect. Hence, networks with ripple-effect of re-clustering suffers high communication complexity than others.
Stationary assumption for cluster formation	In dynamic MANETs, it is difficult for a mobile node to obtain the exact information about its neighbors. However, clustering requires mobile nodes to be static while cluster-related information exchange is carried out in order to get accurate neighbor information. Hence, it is mandatory for a mobile nodes to be assumed stationary while cluster formation is going on.
Time complexity	Time complexity is the number of times an iteration is needed to run so as to complete cluster formation process. This might not be constant for some clustering schemes leading to unbounded time complexity. Consequently, those clustering schemes that need a constant period of motion assumptions are expected to consider this metric into account.

D. Classification of Clustering

Broadly, MANETs can be classified into centralized and decentralized based on their structure. In the former case, the whole network is divided into hierarchical structure where only few nodes are set to control the network function. This is called clustering the network into different clusters. In the later case, there is no centralized management whereby each node is equally responsible for the proper functioning of the network. This can be said flat network structure. Clustering schemes in MANET can be done based on different criteria. Some may classify based on hop-distance between cluster pairs into single-hop clustering [6], [7], [8] and multi-hop clustering [9], [10]. Y. Yu and H.J Chong [5], divided clustering into six based on their objectives. These are DS-based clustering, low maintenance clustering, energy efficient clustering, mobility aware clustering, load-balancing clustering, and combined metric-based clustering. In this paper, we have classified clustering into five based on the metrics used for clusterhead formation as shown in fig.2 below. Namely, node ID-based clustering, node density-based clustering, energy aware clustering, mobility-based clustering, and weighted metric-based clustering. In node ID-based clustering, a mobile node identification number is used to form a cluster. Maximum

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication Pvt. Ltd.

ID node or minimum ID node can be used to cluster the network into different groups. In node density-based clustering, the density of a mobile node with its neighbour is used to divide a network into different clusters. In weighted metric-based clustering, a mobile nodes use multiple metrics for clustering; nodal mobility, distance, degree, battery power, cluster size etc. Researchers may combine and adjust these metrics based on the situation and scenario used for. In energy-aware clustering, a mobile node is used for clustering in such a way that its remaining energy level is relatively high or equal to that of its members. This might help to prolong the network lifetime and cluster stability. In mobility-aware clustering, the mobility behaviour of a mobile node is used for cluster formation. A typical classification of clustering in MANET is depicted in fig. 2.

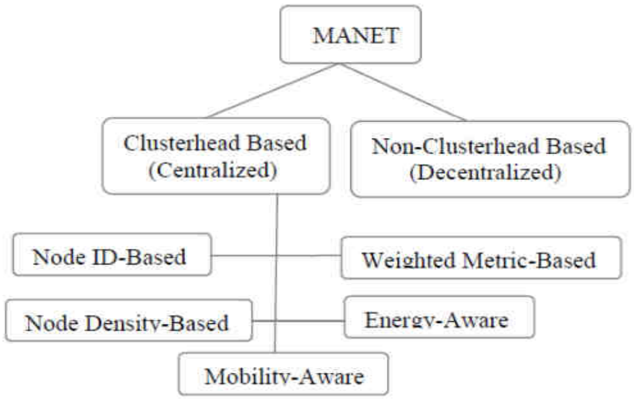


Fig. 2. Classification of clustering based on clusterhead formation metrics

II. NODE ID-BASED CLUSTERING

Node ID-based clustering is one of the most widely used methods of clustering in MANET due to its simplicity [14]. Nodes are assigned a unique identification number so as to uniquely identify every host in the network. This can usually be done either by using its MAC address or IP address as well as by assigning a unique identification number in rare cases. This number is used to group the network into different

Table 2. Summary of clustering schemes

Clustering Type	Definition	Objectives
Node ID-Based Clustering	An identifier-based clustering that heuristically assigns a unique id to each node and chooses a node with either the minimum or maximum id as a cluster head based on the scenario used.	To reduce the complexity and cost of clustering algorithm without affecting the performance of the network.
Node Density-Based Clustering	A connectivity-based clustering in which the density of a node is computed based on its distance from other neighboring nodes.	To generate minimum CDS nodes and increase network scalability.
Energy-Aware Clustering	A type of clustering that takes mobile node's energy consumption in to account.	To assign a relatively high residual energy nodes as cluster head in order to prolong the lifetime of mobile terminals and a network.
Mobility-Aware Clustering	Utilizes the mobility characteristics of a mobile node for cluster formation and maintenance	To enhance connectivity via selecting a relatively low speed mobile node among all its neighborhood
Weighted Metric-Based Clustering	Considers multiple metrics in cluster formation by adjusting their weighting factors for different application scenarios	To obtain an efficient CDS nodes for specific application requirement by properly selecting the metrics and its respective weighing factors

clusters based on different scenario. In a mobile environment, the system performance is better compared with the density-based heuristic in terms of throughput [14], [29], [38]. Although it is the easiest to implement, it suffers from two major draw backs. One is power drainage problem as it is biased to certain nodes which will affect severely the network lifetime because mobile nodes are battery powered. Secondly, loads are not uniformly distributed throughout the network and hence load-balancing problem is inevitable. The node ID-based clustering is done based on either nodes' minimum ID or maximum ID among its all neighbours as follows.

A. Minimum ID-Based Clustering

The minimum-ID, also called identifier-based clustering, was first proposed by Baker and Ephremides [11], [12], [13]. A node that hears its ID is a minimum ID number among all of its neighbours becomes a clusterhead. A node which hears two or more clusterheads serves as a gateway node connecting the neighbouring clusters together. Otherwise, a node becomes an ordinary node. On the other hand, the ID of all neighbours of a clusterhead in the network is higher than that of its clusterhead. Whenever a relatively lower ID node enters into the vicinity of other clusters and identify that its ID is minimum, soon it overtakes the clusterhead role. In a dynamic MANETs, this may results in a frequent clusterhead change hampering the stability of the network. Moreover, minimum ID nodes suffer from power drainage problem as they are more exposed ones to serve as a clusterhead in the network.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication Pvt. Ltd.

Ephremides [12] proposed minimum ID clustering algorithm called linked clustering algorithm (LCA). A unique ID is assigned initially to all the nodes in the network in order to uniquely identify them. A node with a minimum ID form the whole network is elected as a clusterhead as a starting node. All neighbouring nodes covered within the transmission range of the elected node are its clustermember nodes. Among the uncovered nodes in the network, a mobile node with the lowest ID number is arbitrarily chosen to be the next clusterhead. The nodes which lies within the coverage area of the elected clusterhead are all included to be its member nodes. This process will continue until all the mobile nodes available in the network are either elected as a clusterhead or covered to be a member node.

Once a cluster election is over, suitable intermediate node(s) called gateway node(s) is/are elected to form a connected network. In the case of overlapping clusters, any node that lies within the common coverage areas of the two or more clusters can be elected as a gateway node. If more than one node is there, then the one with smaller ID prevails. In the case of non-overlapping clusters, a pair of node one from each cluster reaching one another is elected to form a combined gateway. The combination of clusterheads and gateway nodes form a connected network.

Although the proposed algorithm is simple to implement, node Ids are arbitrarily assigned numbers without considering any qualification of a node to be elected as clusterhead. Moreover, in LCA, minimum ID nodes are prone to power drainage as they are exposed to serve more than other nodes in the network. Thus, it is unlikely that minimum ID nodes' energy level deplete at more faster rate because of excessive tasks they are needed to accomplish as a clusterhead. This in turn reduce the lifespan of the nodes and hence the network, degrading its performance. Besides, in a mobile environment, it is unlikely that the network topology remains the stable resulting in a frequent re-clustering due to high node re-affiliation. Experiments reveal that the performance of the system is better in terms of throughput than the maximum density based clustering algorithm [14], [29], [38].

Let β represents the number of directly connected neighbours that a mobile node can have, then each node is required to send out β messages. Every node is also needed to sendout at least one message indicating about its neighbouring status for election. Because of overlapping cluster structure of LCA, a mobile node may send m such messages on average. Hence, each mobile node is required to send out $(\beta + m + 1)$ messages during cluster formation process. Thus, the overall communication complexity of LCA can be specified as $O(\beta + m + 1)|V|$ where V is the number of nodes in the network.

In [14], D. Gavalas proposed Lowest-ID with Adaptive ID Reassignment (LIDAR) clustering algorithm for MANET. It is introduced to improve the drawbacks of lower ID algorithm by introducing new ID re-assignment at the end of each hello period (HP). The aim is to optimize the scalability and extend the network's lifespan as well as catering for a balanced power consumption among mobile nodes. This is achieved by identifying and electing the most suitable nodes as clusterhead (CH). Suitability of a node here is in terms of having sufficient power level and low mobility rate. Two phases of clustering are there; one is the initial LID based clustering and the second one is clustering in the maintenance phase. In the initial cluster formation phase, a node with lowest-id among all its neighbours is elected a clusterhead. In the cluster maintenance phase, all nodes calculate their weight based on power level and mobility of mobile nodes as follows, $W_i = xB_i - yM_{ip}$ where B_i and M_{ip} are the battery power level and the mean mobility rate during the last period of mobile node i respectively. The sum of $x + y = 1$. Accordingly, CHs sort them in descending order and re-assign new node IDs so that small IDs are assigned to nodes with larger W_i values and large IDs to nodes with smaller W_i values. Hence, lower-ID nodes are elected in the election process and continues like this at the end of each HP duration.

In the presented algorithm, although the idea is to maintain the advantages of lowest-ID algorithm (fast, simple and low-cost clustering process), assigning IDs at the end of each HP period is difficult especially for large MANET. Once we get W_i easily one can elect the clusterhead rather than going for re-arranging and re-assigning new IDs to choose the same node which needs extra resources; bandwidth, power and time. Moreover, the stated way of estimating mobility may not be as such easy because it needs to know the exact location of a node at each interval. Otherwise, we need speed sensing devices that may not be cheap as well. The remaining battery power level estimation is also not realistic. This is because some nodes may be with a high remaining energy level while others are with very low power level. Taking the average as a whole may not lead us to know the exact remaining energy level status of a node.

The communication complexity of Gavalas's algorithm is estimated as follows. In the initial LID clustering phase, first message is sent to assign node ID. Let n represents a node with specific ID, it is required to send 1 identifier message for individual node n in the network. If δ represents the number of directly connected neighbours of a mobile node, then each node is needed to send δ messages in order to detect its neighbours. Based on the received information, every node is required to send at least 1 message to compete for clusterhead. Hence, the message complexity of initial LID clustering is $O[(\delta + 2)|V|]$. In the maintenance phase, every node calculates its weight W_i and unicast it to its CH. Accordingly, a CH receives δ messages from its members. Based on the received value of W_i the CH re-arranges, assign new ID and sent back δ messages to all its cluster members. Now depending on the received new IDs, the above initial LID clustering algorithm run for the election of most appropriate new CH. Thus, during maintenance period, the total message exchange is estimated to be $O[(3\delta +$ *1V.*

However, its time complexity is unbounded. This is due to the fact that in Gavalas's algorithm, a mobile node cannot declare itself as a clusterhead until it finds that all direct neighbours with a lower ID are involved in some clusters as cluster members. As a result, the time complexity for those algorithms is difficult to determine. In the worst case, the number of rounds for completing the cluster formation procedure is equal to the number of clusters, which indicates that only one clusterhead is decided in each round [5].

B. Maximum ID-Based Clustering

It is an algorithm whereby a node with a maximum ID number among all of its neighbours becomes a clusterhead [15]. All the neighbouring mobile nodes have a lower ID

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication Pvt. Ltd.

number as compared to its own clusterhead. This may produce a relatively better stable CDS nodes in the network as compared to highest connectivity clustering algorithm [29]. However, it lacks load-balancing as well as higher ID nodes suffer from power drainage.

D.J. Baker's Maximum ID Algorithm: in [16], D.J Baker and A. Ephremides introduced maximum identity-based clustering for the first time. They came up with a clustering-rule that a node with the maximum identity number among a group of nodes is the first candidate to claim clusterhead status. Node i checks whether a node with a highest identity number greater than his own is there or not among all of its neighbours. If there is no neighbour with higher identity number, node i becomes a clusterhead. In addition to highest ID number, i must also satisfy the condition that it is the "highest" neighbour of some other node(s). This can be done by checking the received connectivity rows from the lower numbered neighbours [16]. In case a clusterhead is covered by another clusterhead, then one of it drop its role becoming ordinary node.

In order to connect clusterheads together so as to form connected network, an intermediate nodes called gateway nodes are used. Any non-clusterhead node is a candidate to be gateway node. Two cases are there; the case of overlapping and non-overlapping clusters. In the former case, any node that hears from two clusterheads and examining that they are unconnected, then it links them together. If more than one nodes sense it, then the one with higher ID is chosen to be a gateway. In the later case, at least one node from each cluster must become a gateway. Each node checks every possible pair of nodes, the first member of which is its own clusterhead and the second member of which is a neighbouring clusterhead. Creation of redundant gateways are avoided by checking for the existence of such a link beforehand.

The proposed algorithm's main attention is on the maximum-ID which is arbitrarily assigned number without considering any speciality for election. To avoid the conflict that a node is without a member, node density (at least one) is considered as a second qualifications of a node to be elected as a clusterhead. This will reduce the size of CDS nodes in the network. However, node IDs do not change over time, those with maximum IDs are more likely to become clusterheads than nodes with minimum IDs. Thus, certain nodes are prone to power drainage due to serving as clusterheads for longer periods of time. This may degrade the network performance by reducing the lifespan of the mobile nodes.

The communication complexity of Baker's algorithm is computed as follows. Let β be the number of directly connected neighbour of a mobile node, each node is required to send β messages to inform its neighbours about its existence. Then every mobile node sends out 1 message to claim for clusterhead. As non-overlapping clusters are required, the next message is sent to eliminate redundant clusterheads if any. Finally, in order to link the clusters together message exchange is needed to select appropriate gateway nodes. Suppose n is the number of nodes common to two or more clusterheads in the network and δ represents the message sent by a node. Then, there will be $n * \delta$ message communication needed to connect clusterheads together. Therefore, the overall message complexity of Baker's algorithm is estimated to be $O[(n * \delta + \beta + 1)|V|]$ Where, V is the number of nodes involved during each period of message exchange in the network. However, its time complexity is unbounded.

Fig. 3. Illustration of Node ID-Based Clustering

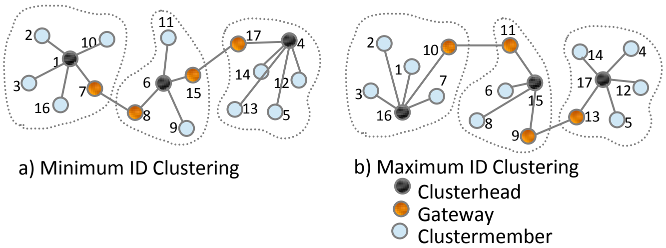


Fig. 3 shows a typical graph of Node ID-based clustering. There are (17) nodes with unique IDs and identical transmission range which form a connected graph. In (a), after the Minimum-ID clustering algorithm is executed, three clusters are formed, as depicted by the three dotted lines. The three dark-coloured circles indicate clusterhead nodes (1, 4, and 6) with minimum IDs among all its respective neighbours. The orange-coloured circles are gateway nodes (7, 8, 15, and 17) connecting the neighbouring clusters together. The light blue-coloured circles are ordinary nodes. Similarly, in (b), Maximum ID-based clustering algorithm is executed forming three clusters. Accordingly, nodes (15, 16, and 17) are elected as a clusterhead nodes while (10, 11, 9, and 13) form gateway nodes. Although nodes (7 and 8) hear each other, they are not chosen to be gateway nodes as depicted in (b). Rather, nodes (10 and 11) are selected because of their greater IDs than the two.

III. NODE DENSITY-BASED CLUSTERING

Node density-based clustering algorithm is based on the density of uncovered neighbouring nodes. The density of a node is simply represented as the number of uncovered neighbours. Nodes broadcast "hello message" periodically to collect information from its neighbours. A mobile node receiving minimum/maximum hello messages (based on the algorithm used) than all its uncovered neighbouring nodes has a priority to be added to CDS node list.

A. Minimum Node Density-Based Clustering

This algorithm is based on minimum density of uncovered neighbours of a mobile node. It starts with a node having minimum number of connectivity among its uncovered neighbours and continues until all nodes in the network are covered. It ascends from a node with lower uncovered neighbours to nodes with higher uncovered neighbours.

In [17], C.S Victor and G. Amalanathan proposed a strategic minimum density minimum velocity CDS (ST-MD-MV CDS) algorithm. It is a clustering algorithm based on the density of uncovered neighbours of a mobile node. A locally suitable node can be elected as an initiator node and then the density of covered nodes are calculated. A mobile node with a minimum number of uncovered nodes among its neighbours (at least one) is included to CDS node list and goes up until all nodes are covered. In case two nodes face the same density, minimum velocity is used to break a tie. The proposed algorithm's principle is similar to that of maximum density based clustering. The difference between the two algorithm is that clusterhead election is done in either descending or ascending order of density of

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication Pvt. Ltd.

uncovered neighbouring nodes and complexity. In the case of minimum density-based clustering, clusterhead election is done first starting from less uncovered neighbouring nodes and goes to the highest uncovered neighbouring nodes sequentially while the second one is vice versa.

In the proposed minimum density-based algorithm, a link failure of even a single node, may lead to clusterhead re-calculation process in the whole network. This consumes network bandwidth and battery power degrading system performance. Moreover, it generates high communication overhead in the cluster formation phase. The communication complexity of the algorithm is about $O(n^4)$ where n represents the number of nodes in the network.

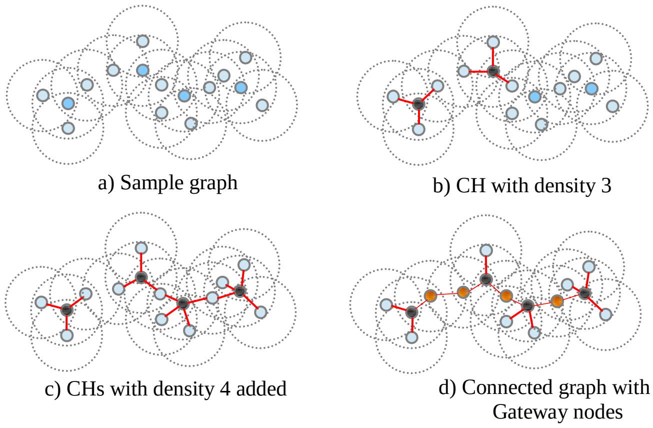


Fig. 4. Illustration of Minimum Node Density-Based Clustering

Fig. 4 depicts a typical minimum node density-based clustering. In the graph, there are sixteen mobile nodes with equal transmission radii. In (a) the sample network graph is shown before running the algorithm. After a minimum node density- based clustering algorithm runs, three clusters are elected based on the following steps. The clustering starts with the minimum density nodes elected first and goes up until all nodes are covered. As indicated in (b), two clusterheads each with three neighbouring nodes are chosen first and dark-coloured. Then in (c) two more clusterheads with density of four are elected and added to the previous ones. Finally, with the help of gateway nodes, the clusterheads are connected together forming a connected network as shown in (d). However, in the case of maximum node density-based clustering algorithm, the steps are vice versa.

B. Maximum Node Density-Based Clustering

A Mobile node having maximum node density than all its neighbours is elected to be a clusterhead [15], [13]. In [18], [19], the researchers proposed a greedy algorithm whereby 2-hop neighbour information is needed to elect DS node. A node with maximum number of 2-hop uncovered neighbours among all is elected.

In [13] M. Gerla and J. Tsai presented a multi-cluster, multi-hop packet radio network by using a distributive clustering algorithm. They considered maximum density based clustering algorithm to choose a suitable node for their network scenario. The density (connectivity) of a node is computed based on its distance from others. Each node broadcasts the list of a 1-hop neighbouring nodes it can hear periodically to show its connectivity. A mobile node having the highest uncovered nodes among all its neighbours is elected as a clusterhead. In case of a tie, the lowest ID node is given priority. All covered nodes become members of a cluster and are not allowed to participate in clusterhead election anymore. A node common to two or more clusterheads becomes a gateway node facilitating inter-cluster communications. Any two clusterheads are at 2-hop distance from each other as they are not directly linked.

In the proposed algorithm, as the number of nodes in a cluster is increased, the throughput of the system gradually degrades. This is because the algorithm is focused on generating minimum size of the network, CDS nodes. But a clusterhead node can handle only a limited number of nodes at a time and hence it is upper bounded. Moreover, in a dynamic network where the number of node affiliation is high, clusterhead changes frequently which increases network instability. Even in the presence of one link change due to node movement, it may fail to be re- elected as a clusterhead. This may also increase the network overhead and battery power consumption.

Assume β be the number of direct neighbours of nodes that a mobile node can have at any instant of time t then, each mobile node sends β message to detect its connectivity with its neighbours. Every mobile node is also needed to send out at least one message to claim its information. Thus, each node is expected to send out $(\beta + 1)$ messages for cluster formation. Hence, the communication complexity of M. Gerla's clustering algorithm is $O[(\beta + 1)|V|]$ where V is the total number of mobile nodes in the network.

In [20], a Timer-Based CDS protocol is proposed by K. Sakai, aiming at producing a lesser number of dominating set. The algorithm is distributive and applies defer timer protocol at each mobile node to reduce network overhead. The defer timer is set at every node and a mobile node experiencing higher connectivity among all of its neighbours be marked as DS node. Three phases are there for cluster formation. In the initiator election phase, a number of local minimal (LM) nodes are elected to be initiators distributively in multi-initiator (MI) case while only one in single-initiator (SI). In the tree construction phase, a disjoint dominator trees are established using the set defer timer. A defer timer is set inversely proportional to the number of uncovered neighbours granting a higher priority to nodes having more uncovered neighbours. In other words, a mobile node with a maximum density of uncovered neighbours is set with a relatively low value of defer timer giving higher priority to be elected as a DS. In tree connection phase, the initiator node sends a control message to border leaf nodes of its own so as to connect to other neighbouring tree in MI while SI do not need this step as it is already connected. Dominator trees plus those nodes connecting trees together form CDS nodes.

Although communication overhead is reduced during tree construction phase, it is tedious and time taking to set defer timer at each node. In SI, the algorithm produces a minimum possible CDS nodes. But it consumes a significant amount of time for tree formation as it is based on single initiator node acting as a root for whole network. In maintenance phase, it may create a lot of overhead as it sends messages back to initiator node. When initiator nodes fail, it starts from the scratch which increases the cost of maintenance. On the other hand MI requires less time for tree construction phase as it uses multi-initiator nodes simultaneously opposed to SI. However, many initiators may increase the number of CDS nodes generated in the network as many intermediate nodes are required to connect

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication Pvt. Ltd.

the trees together. This in turn increases the overall power consumption of the system which is in fact a limited resource, degrading the network lifespan.

In the SI CDS construction algorithm, there is no need for tree connection phase. Thus, its message complexity is the grand sum of those messages sent during initiator election and tree construction phases. Accordingly, a message is sent to its neighbours from the initiator initialization node so as to elect the lower ID node among all nodes in the network to be initiator, $O(|V|)$ messages are sent where V is the number of nodes in the network. When a node finds that it is the initiator, it immediately switches its state to dominator and its initiator to itself while sending beacon announcement for tree construction whereby $O(|V|)$ messages are sent. Hence, the overall message complexity in SI CDS construction algorithm is $O(2|V|)$ Its time complexity is also $O(2|V|)$ as in each phase, every node in the network is needed to sequentially process the steps.

In the MI CDS construction algorithm, the message complexity is equal to that of SI plus the tree connection phase. In tree connection phase, each border (leaf) nodes send message (contains its own id and initiator id of its neighbouring tree) to its root. On receiving messages from its neighbours, it covers about the neighbouring trees, it only forwards one copy of the messages if the initiator id contained in the messages are the same. Otherwise, it forwards different messages for each of the different initiator id's it has received. Here, its message complexity becomes $O(b * d)$ where b is the number of boarder nodes in the network and d is the number of neighbouring dominator trees. Consequently, the overall message complexity of MI CDS construction algorithm is $O[(2|V|) + (b * d)]$ which is higher than that of SI. Its time complexity is that of SI divided by the number of initiator nodes used in initiator election phase which is lesser time than that of SI. However, the overall time complexity is unbounded because that of tree construction phase is not bound.

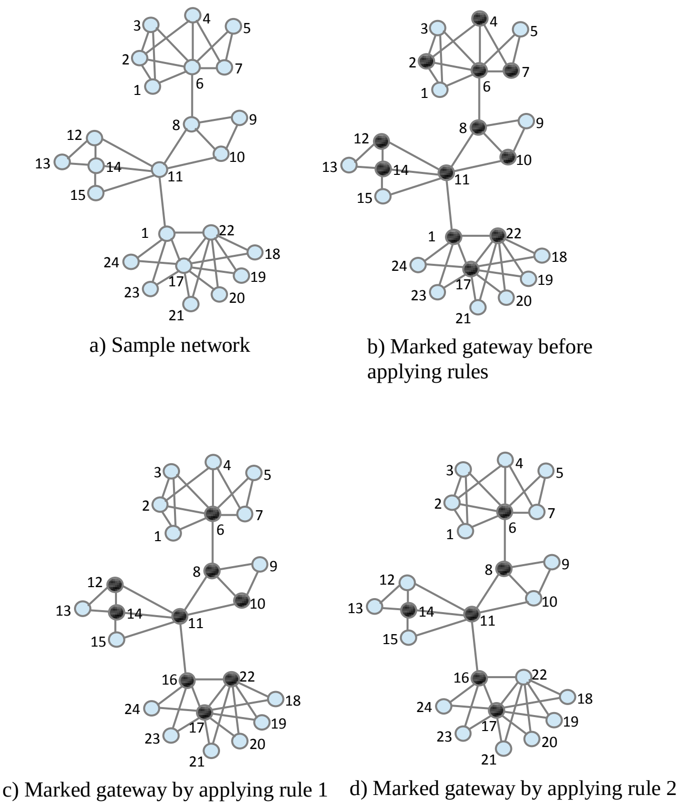
IV. ENERGY-AWARE CLUSTERING

In MANET, nodes are battery powered and needs a close attention how to use this limited resource so as to prolong the network life. For the past few decades, this has been remained a challenge for researchers, negatively influencing the network performance [21], [22]. Thus, strive has to be done in order to come up with a better design that will increase the performance of the network by reducing the energy consumption. In MANET, clusterhead nodes are more exposed to power drainage than ordinary nodes because of many duties assigned to them. This may cause some nodes to go off rapidly resulting in the network disconnection. One way of prolonging the lifespan of such nodes is by using load-balancing method whereby load is shared among mobile nodes uniformly. On the other hand, when clusterhead election is executed, the algorithm should be done in such a way that the remaining energy level of nodes are taken into consideration. Choosing a relatively higher residual energy node every time clustering algorithm runs not only balances the load among the nodes but also prolong the lifetime.

Wu's Algorithm: in his previous work [23], Wu and Li proposed a simple and fully distributed algorithm for the construction of CDS nodes in MANET. Depending on this work, in [6], Wu proposed an extension which is energy-aware algorithm for CDS formation that is based on a marking process. It marks every elected node to be clusterhead while ordinary nodes are left unmarked. As the number of nodes in CDS determines the lifespan of the network, this algorithm is focused on reducing the size of CDS by inhabiting some DS nodes depending on its energy level without altering the network performance. In the process of unmarking redundant DS nodes, the remaining energy level of a node is the main factor while node degree and id are used to break a tie. The DS node is removed when all of its neighbours are covered fully by one or more other nearby connected dominating sets as well as having less power.

In the proposed energy-aware algorithm, although better efforts have been done to remove some DS nodes based on remaining energy level, the great energy consumption difference between DS and ordinary nodes are not yet seen. Its main target is to trim down the size of CDS nodes in the network so that the overall high energy consumption is reduced. Hence, CDS nodes are likely deplete their energy at faster rate than others thereby shortening the network lifespan. However, it reduces the communication overhead while DS nodes exchanges periodical updates as well as insures a faster packet transfer through the network. From its cluster structure, we can say that Wu's algorithm is more effective if it is applied to networks with high density and low mobility behaviour.

Fig. 5 illustrates Wu's marking process to identify CDS nodes. Every mobile node finds its 2-hop neighbours information periodically through sending hello message. In Fig. 5(a), depicts the sample network. In (b), the graph indicates marked gateway nodes without applying any marking rules. Node 2 is marked gateway node because its neighbours (1) and (4) are not directly connected. A total of 12 gateway nodes are there.



a) Sample network
b) Marked gateway before applying rules
c) Marked gateway by applying rule 1
d) Marked gateway by applying rule 2

Fig. 5. Illustration of Wu's marking process

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication Pvt. Ltd.

In (c), nodes (2, 4 and 7) unmark themselves and become ordinary node as the three are all subsets of node (6) as well the id of (2 and 4) are lower than that of the group (2, 4, 6 and 7) nodes. By applying rule 1, the number of gateway nodes reduced to 9. In (d), $N(10) \subseteq N(8) \cup N(11)$ but $N(11) \nsubseteq N(8) \cup N(10)$ and $N(8) \nsubseteq N(11) \cup N(10)$ then node (10) will be unmarked. Similarly, for node (12), $N(12) \subseteq N(14) \cup N(11)$ and $N(14) \subseteq N(11) \cup N(12)$ but $N(11) \nsubseteq N(14) \cup N(12)$. Moreover, the degree of node (12) is less than that of node (14). Hence, node (12) unmark itself and becomes ordinary node. The same logic applies for the rest based on rule 2 of marking process. A further application of marking rule 2 reduced the total number of gateway nodes to 6.

The communication complexity of Wu's power-aware algorithm is similar to that of his previous CDS formation algorithm [23]. First marker assignment message is sent to all nodes. Then every node exchanges its directly connected neighbour set in order to mark if any two unconnected neighbour exists. Thus, the communication complexity of this scheme is $O(2|n|)$ where n is the number of all nodes in the network. Wu's algorithm is efficient in time as it is an extension of his earlier CDS algorithm and can be done in two rounds. That means each of the two rules of the marking process requires one round to be completed resulting in a total of two rounds.

Xiong's Algorithm: Xiong [24] proposed an ordered sequence list to find an efficient minimum CDS for WSNs. They used pruning methods to remove redundant DS nodes to achieve a possible minimum CDS size. As nodes in CDS are relaying nodes their energy depletes more rapidly than other nodes. This may cause some CDS nodes to die out resulting in network partitioning. To reduce this effect they have proposed a Minimum Energy-consumption Broadcast Scheme (MEBS) with a modified version of EMCDS, aimed at providing an efficient scheduling scheme with maximized network lifetime. This helps minimum energy level CDS nodes to be removed prior to creating network disconnection.

The proposed algorithm is better energy-aware clustering during maintenance phase than the available CDS construction as it ignores low power nodes from packet relaying duty. But their algorithm removes active CDS node only when it becomes unable to forward a packet before network disconnection happened. This may not be good as the node gives no more service in the network due to its very low power. That means it is not focused on balancing the great energy consumption difference between CDS and non-CDS nodes rather on avoiding network disconnection by replacing low energy level CDS nodes right before dying-out. On the other hand, once a node is elected as CDS, it will continue serving until its battery gets keen to die. This does not guarantee the presence of some nodes in the network until the end which may result in even more network partitioning after some period of time. Hence, the performance of the network is severely affected as there might be many partitions created in the network unless a dying-out nodes are get replaced specially in the case of static MANET. Moreover, their algorithm creates a lot of communication overhead during construction phase, consuming a significant amount of bandwidth and battery power.

Let x and α be the number of nodes in the network and the node's number of hop from the source node respectively. Each node sends $O(x * \alpha)$ messages in the breadth first search (BFS) algorithm where nodes identify their neighbours. Then, in the process of building up the ordered sequence list, all layers and all nodes in the same layer are needed to be traversed which needs $O(xlogx)$ messages to be sent by every node. During MIS construction, all nodes in the network need to be traversed requiring $O(x)$ messages to be sent out. While constructing dominating set, each node sends $O(x^2)$ messages in order to check their candidate parents. Lastly, all CDS nodes and their parents are traversed to avoid the redundant CDS nodes which needs $O(x^3)$ to be sent. Consequently, the overall message complexity of Xiong's algorithm is $O(x^3)$. Similarly, the time complexity of Xiong's EMCDS algorithm is all rounds needed in the (BFS) algorithm, in the ordering and layering sequence, in constructing the connected dominating set, and in removing the redundant CDS nodes which can be specified as $O(x^3)$.

Rai's Algorithm: Rai [25] proposed power-aware minimum CDS algorithm. Three phases are there in his cluster formation algorithm; first, dominating nodes are identified based on the maximum number of uncovered neighbouring nodes. A node with a maximum number of uncovered neighbouring nodes among all its neighbours is added to DS list. If two nodes have the same maximum number of uncovered neighbours, the one having the lowest node id is given priority to be elected. In the second phase, the selected DS nodes are connected together using Steiner tree by adding some intermediate nodes so as to form CDS. In the third phase, the redundant DS nodes are removed from the CDS resulting in MCDS nodes. CDS nodes consume more energy than ordinary nodes and their failure may result in a topology change which will consume huge network bandwidth as well. For these reason, they tried to propose an energy efficient local repair using only neighbouring nodes than running total CDS re-calculation.

The proposed algorithm is aimed at constructing a MCDS nodes through which energy saving is achieved by minimizing the number of CDS participants. Furthermore, Rai's localized clustering while maintaining a broken link, greatly reduces energy consumption. However, balancing the energy consumption among nodes in the network to prolong its lifetime is not yet seen.

The message complexity of Rai's algorithm is the total message communication in each of the three phases. First the message containing id of each node is sent, $O(|V|)$. Each node need to send a message containing its own id so as to identify its direct connectivity, C and at least one information to claim for clusterhead, $O(C+1)$. In the tree connection phase, each gray node broadcasts its maximum connectivity with dominating sets, $O(|V|-|DS|)$. In the pruning phase, each dominating set send a message about its direct connectivity to claim for its presence in CDS, $O(|DS|)$. Thus, the overall communication complexity of Rai's algorithm is $O[(3+C)|V|]$ where C is the number of direct neighbours of a node and V is the total number of nodes in the network.

V. MOBILITY-AWARE

Mobility of a mobile node is one of a major contributor for

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication Pvt. Ltd.

the network topology change in MANET. Because of this fact, mobility-aware clustering is a type of clustering scheme in which the mobility behaviour of a mobile node is considered. The stability of a cluster depends on how fast the mobile nodes are moving in the network. As the mobility of mobile nodes increases in the network, the change in topology increases too and vice versa. This affects the network performance severely if not treated well. Thus, mobility-aware clustering takes this behaviour of nodes into account so as to produce a better stable network.

Basu's Algorithm; in [26], P. Basu proposed lowest mobility clustering algorithm called MOBIC for MANET. The algorithm is almost similar to lowest ID clustering in execution except that the ID information is replaced here by mobility. Clusterhead selection is based on the mobility behaviour of a mobile node relative to its local neighbours. A mobile node experiencing a relatively low aggregate local mobility among all its 1-hop neighbours has a priority to be chosen as a clusterhead. MOBIC points out that clusterhead election is a local activity whereby only its neighbours and itself can make a decision.

In MOBIC, by calculating the variance of a mobile node’s speed relative to each of its neighbours, the aggregate local mobility of a mobile node can be estimated as follows [26].

$$M_{Yag} = var_0[M_Y^{rel}(X_1), M_Y^{rel}(X_2), \dots, M_Y^{rel}(X_n)]$$
$$M_{Yag} = E[(M_Y^{rel})^2]$$

On the other hand, a mobile node with low variance value indicates that it is relatively less mobile with respect to its neighbours [26]. Consequently, a mobile node with a low variance value in its locality takes the clusterhead role. In case two mobile nodes have the same variance value, the one with lower node ID is considered.

The cluster maintenance of MOBIC is similar to that of least cluster change (LCC) [27] except that MOBIC uses a cluster-contention-interval (CCI) timer to avoid unnecessary clusterhead changes while two clusterheads comes in each other's transmission rage for only a short period of time. That means for two incidental contacts of passing clusterheads, there is no need of re-clustering and hence, the two remains in their own state. Re-clustering is invoked only if the two remains in contact even after CCI timer has expired whereby the one with the lower mobility metric over takes the clusterhead role. This reduces the frequent changes in the network topology due to re-clustering which enhances stability.

In the proposed algorithm, the selection of a low mobility node improves the network performance by reducing the rate of re-clustering. However, its performance degrades if the mobility of cluster member nodes changes dynamically. Hence, MOBIC is more effective and useful if it is applied to a group mobility where the group members are moving with similar speed and direction. Otherwise, the mobility behaviour of its members forces the clusterhead to change its role frequently which may miss the intended target of providing stability.

Let n represents the number of 1-hop neighbours of a mobile node. During clustering, every node sends two consecutive messages to each of its direct neighbours in order to identify their relative speed. Consequently, each mobile node communicates $2n$ messages. Every node broadcasts its own information containing aggregate local mobility to claim for clusterhead. Because of overlapping cluster structure in MOBIC, there might be a probability of more than one cluster-related messages to be sent while clustering. Each mobile host is required to send m such messages on average. Thus, each mobile node is needed to send out a total of $(2n + m + 1)$ messages. Therefore, the message complexity of MOBIC for cluster construction phase is specified as $O(2n + m + 1)|V|$ where V represents the number of mobile nodes in the network.

Zhang's Algorithm; in [28], Yan Zhang proposed a distributed group mobility adaptive (DGMA) clustering algorithm for MANETs. The researchers selected and used Reference Region Group Mobility (RRGM) model as it supports re-clustering (partitioning and mergence). DGMA introduced the mobility metric linear distance based spatial dependency (LDSD) to measure the physical displacement changes of a mobile node. LDSD is calculated based on the speed and direction information exchanges by neighbouring mobile nodes as; $LDSD(x, y, t) = RD(x, y, t) * SR(x, y, t)$ Where, RD and SR are the relative direction and speed ratio of nodes x and y at time t respectively. Higher $LDSD$ shows that the nodes speed and direction of movement are more correlated. On the other hand, less $LDSD$ means the two nodes are moving away from each other.

The proposed algorithm improves most of the draw backs of LID clustering and MOBIC [26] as it has considered the spatial dependency of nodes (i.e. nodes' related speed and direction) for clustering. It is very useful and applicable in groups having similar characteristics like related mobility and direction as in highway traffic. However, it may not represent a general MANET characteristics as the nodes' movement is dynamic and random in both speed and direction. In the worst case, all the mobile nodes in the network might vary in both speed and direction resulting in no clusters. That means all mobile nodes serve as a clusterhead individually which may degrade its performance.

In DGMA, let Δ represents the directly connected neighbours that a mobile node can have, each node is required to send out its own information in order to determine its spatial dependency with its directly connected neighbours. Then, every node sends Δ messages. Each mobile node sends out at least 1 message containing its total spatial dependency to claim for clusterhead. Since DGMA supports an overlapping cluster structure, mobile nodes may send such information more than 1 times. Let's say n messages are sent on average. Therefore, every node is required to send out a total of $(\Delta + n + 1)$ messages for cluster construction phase in DGMA. Thus, the overall message complexity of Zhang's algorithm can be specified as $O(\Delta + n + 1)|V|$ where V stands for the number of mobile nodes available in the network.

Suppose in the worst case, there is only a single clusterhead in the network where all the other nodes are considered to be neighbour of it. In cluster formation phase, this needs a processing time of $2(n-1)$. Similarly, in maintenance phase, it requires a maximum of $(n-1)$ processing time at each red and yellow nodes and $2(n-1)$ at white nodes. Hence, the overall time complexity of DGMA algorithm is $O(n)$ where n is the number of mobile nodes in the network.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication Pvt. Ltd.

VI. WEIGHTED METRIC-BASED CLUSTERING

This is a type of clustering algorithm that incorporates more than one factor whereby the resultant effect of the metrics are used to determine the would be cluster leader among its neighbouring nodes. It is also called Combined-metrics-based clustering (CMBC) as it takes a number of metrics into account for cluster formation [5]. Some of these metrics include node degree, remaining energy level, velocity of nodes, distance between mobile nodes and so on. These metrics are adjusted and combined flexibly for different scenario to choose the most suitable clusterhead among all of its neighbours. The type and weighing factor varies depending on the situation and specific application needed [29]. On the other hand these parameters may not be useful and available at the same time for every scenario as it alters the performance of the network. Hence, one is expected to appropriately select it based on the situation and scenario used.

M. Chatterjee's Algorithm: in [29] Chatterjee proposed an on-demand distributed clustering algorithm for multi-hop packet radio networks. It is a weight-based clustering algorithm (WCA) that considers the following factors in to account for cluster formation. Node degree, transmission range, energy and mobility of the nodes are the metrics used for clustering. The algorithm, WCA, combines each of the above system parameters with certain weighing factors chosen according to the system needs. More precisely, a specific application may give more emphases for only one or two of them while the rest are given less. Accordingly, more weight is assigned for the one given more emphases than the lesser one.

The combined weight factor, W_v, is given as; $W_v = a_1C_v + a_2D_v + a_3M_v + a_4T_v$ Where, C_v, D_v, M_v and T_v represents node connectivity difference, total distance to all neighboring nodes, mobility and battery power respectively while a_1, a_2, a_3 and a_4 are the weighing factors and the sum is normalized according to predefined values (one). i.e, $\sum_{i=1}^{4} a_i = 1$ and $C_v = |d_v - \delta|$ where d_v is node connectivity and δ is constant.

Based on the calculated combined weight, a mobile node with a minimum value of W_v among all its neighbors will be elected to be clusterhead. This process goes on till all nodes become either a clusterhead or member of it.

The proposed algorithm performs better than the available ones in that period. It considered the dynamic nature of MANET as it uses the combined weight metrics which deals with various aspects of a mobile node. Load-balancing issue is also tried to be addressed better than other approaches. However, clusterhead election algorithm is not frequently invoked unless there is a relative change in distance between nodes and their clusterhead. This may not be true even though a relative change in distance is there, as far as the nodes are within the transmission range of its clusterhead, no need of initiating a clusterhead re-election process. On the other hand, in the absence of any relative change in distance, the battery power of a clusterhead may get depleted and/or a node may face some problem which needs its clusterhead role to be handed over to other suitable node. Moreover, the difference in power consumption between ordinary nodes and clusterhead is another problem that severely affects the network performance in shortening node's existence.

Besides, clusterhead serving time T_v alone cannot guarantee a good assessment of energy consumption because data communication consumes a large amount of energy and varies greatly from node to node depending on their load. On the other hand, the value of δ is not well estimated quantitatively so as to balance the load distribution among clusterhead nodes. As node degree varies from one node to another locally, δ cannot be globally represented. That is subtracting a constant number from all nodes may not contribute any in load-balancing as δ needs to change so that the effect will be seen.

Let the maximum number of 1-hop neighbours of a mobile node be represented by Δ. Then, every mobile node is expected to send Δ messages in order to identify all its neighbours. Moreover, each mobile node broadcasts at least one message to claim its own information, containing the degree difference, average speed, distance sum to all direct neighbours, and clusterhead serving time, to its 1-hop neighbours. Because of an overlapping nature of cluster structure, a mobile node may broadcast more than one message about its cluster-related status. Suppose that each mobile node sends out x such messages on average. Accordingly, the overall messages that each mobile node requires to send out for clustering is at least $(\Delta + x + 1)$. Hence, the communication complexity of Chatterjee's algorithm can be represented as $O[(\Delta + x + 1)|V|]$ where V is the total number of mobile nodes available in the network. Although the initial clustering in WCA depends on the stationary assumption of clustering, it requires a non-constant number of rounds to complete its cluster formation indicating that the time complexity is unbounded [5].

M. Aissa's SLWCA Algorithm: In [30], M. Aissa has proposed Stable Load balanced Weighted Clustered Algorithm (SLWCA) with two new modifications; node stability and load balancing models. The former modification was done on the previous works of M. Amine [31] which says the node stability is a function of distance between the two mobile nodes (transmission range) while the later is based on the works of M. Chatterjee [29] who proposed cluster size bound δ in WCA for the purpose of load balancing. The modified stability is a function of node degree as follows;

$$\varphi(n_i) = \sum_{i=1}^{N(n)} \varphi(n_i, n_j) = \deg(n_i)$$

From this the stability weight $S(n_i)$ of node n_i is given as; $S(n_i) = \ln \deg(n_i)$. He concluded that directly connected links are more stable than others. Stability can also be achieved via minimizing the number of clusters formed and the number of re-affiliations under various situations.

In a clustered topology, clusterhead nodes are usually assigned various tasks than ordinary nodes. In addition to guarantying its members with radio resources, has to involve in other numerous inter-cluster duties. In fact the load handled by a clusterhead depends on the number of nodes supported by it. As a result, it is neither desirable to have a clusterhead over loaded nor easy to maintain a perfectly load-balanced system at all times due to frequent out and in of the nodes from and to a cluster.

In [29], [32], [33], [34], [35], the authors proposed that the maximum number of a cluster member is limited by a value δ and the degree-difference is calculated as: $\Delta(n_i) =$

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication Pvt. Ltd.

$\deg(n_i) - \delta$. However, M. Aissa [30] pointed out that the above authors did not specify how to choose δ as improper selection of it may lead to generate many clusterheads resulting in high power consumption. Furthermore, setting a global bound may not be realistic as node degree is not uniformly distributed and rather it varies from one node to another throughout the network. In [36] the authors assumed a uniform distribution of nodes and concluded that the number of members of each clusterhead on average should be half of its directly linked neighbours. As nodes are not uniformly distributed, M. Aissa [30] proposed the following inequalities to overcome the inefficiencies; $\frac{\deg(n_i)}{2} < tdeg(n_i) \le \deg(n_i)$ Where $tdeg(n_i)$ is the number of nodes that a typical clusterhead incorporates. Consequently, a load balanced cluster should contain on average the three fourth of its expected size of direct linked neighbours. On the other hand, a cluster that is too small, may produce a large number of clusters increasing the length of hierarchical routes, resulting in longer end-to-end delay. Accordingly, the node degree limit is set as; $\frac{\deg(n_i)}{2} < \delta \le \frac{3}{4}\deg(n_i)$

Unlike to the previous works, δ is a local degree bound which varies from one node to another and thus is more flexible. Accordingly, when a cluster size exceeds the predefined limits mentioned above, re-clustering procedures are invoked to adjust the number of mobile nodes in that cluster. The researchers also introduced the relative typical degree of a node n_i is compared to the network link and is given by; $rtdeg(n_i) = tdeg(n_i)\frac{1}{|E|} = \frac{3}{4}\deg(n_i)$

Based on the above modifications, their proposed combined weight for clusterhead selection in SLWCA is; $W_i = a_1M_i + a_2Rbl_i + a_3S_i + a_4rtdeg_i$ Where, a_1, a_2, a_3 and a_4 are the weighing factors corresponding to the relative parameters whereby; $\sum_{i=1}^{4} a_i = 1$, and $M_i, Rbl_i, S_i, rtdeg_i$ are mobility of a mobile node i, the remaining battery level of a node i, the link stability of node i, and the relative typical degree of node i, respectively.

The proposed algorithm dealt better in quantifying both the link stability and load-balancing for better performance. However, the stability is affected much more by the mobility of the nodes than the number of clusterheads in the network. Besides, the way of estimating the remaining battery level of a node is not yet quantified. Although it improves its performance, the one used for illustration may not be realistic due to some main logics are missed in his clustering. A great improvement is shown in reducing the number of clusterheads than in WCA [29]. But there is no enough justification presented on how the clusterheads (1 and 8) communicates with its 2-hop neighbours (node 11 and 6) respectively. They might need to use the intermediate nodes (10 and 7) which will increase the number of clusterheads formed than expected one.

The message complexity of SLWCA is similar to that of on-demand WCA while its time complexity is unbounded too. Suppose the maximum number of a directly linked neighbours of a mobile node be represented by β through which a node can identify its two hop neighbours, then every mobile node is required to send β messages in order to identify all its 2-hop neighbours. Moreover, each mobile node broadcasts at least one message to claim its own information, containing its weight, relative typical degree, speed, stability, and remaining battery level of a node to its neighbours. Because of an overlapping nature of cluster structure, a mobile node may forward more than one message about its cluster-related status. Let's say that each mobile node sends out m such messages on average. Consequently, the overall messages that each mobile node needs to send out for clustering is at least $(\beta + m + 1)$. Hence, the communication complexity of Aissa's algorithm can be represented as $O[(\beta + m + 1)|V|]$ where V is the total number of mobile nodes available in the network.

Yang's Algorithm; in [37], [38] Yang proposed a weight-based clustering algorithm (WBCA) which takes the mean connectivity and battery power of mobile nodes into consideration. The mean connectivity degree M_c depends on the mobile node's own connectivity and its 1-hop neighbours as well.

$M_c = \frac{\sum_{i=1}^{|N(n)|} C_{ni} + C_n}{C_n + 1}$, Where C_{ni} is the connectivity of i^{th} neighbor of node n. Accordingly, the connectivity difference of a mobile node n is calculated as the difference between the connectivity of node n and its mean connectivity with neighbors; $D_n = |C_n - M_c|$. The battery power consumed by a clusterhead node, B_n, during the period of its supervision is calculated as; $B_n = \sum_{i=1}^{t} C_{ni}T_{ni}e$, Where, t is the time period node n acts as a clusterhead, C_{ni} is the connectivity, T_{ni} is the time of n acts as a clusterhead at i^{th} time, e is the consumed battery power per node.

The proposed algorithm estimates the degree difference in terms of its mean connectivity of clusterhead locally. Even though it varies from one clusterhead to another, it may not help in over all load balancing as it is not estimated based on the network size. Rather it is limited to local area which may not have any relation with the others in the network. Although the consumed battery power of a clusterhead on service is tried to be estimated better relative to the available ones, some facts are still remains untouched. Basically, the battery consumption of clusterhead node is related to all the activities it performs. Consequently, all cluster member nodes may not be expected to send the same data all the time. Rather, some may need to send while others are not having any to do so. During periodical updating and cluster formation, all cluster members might be required to send probably identical data. For this fact, Yang's way of estimating the consumed energy for a clusterhead node during its supervision may not be a representative one. The consumed energy per node, e is also not yet quantified. Moreover, it deals with that of clusterhead but not general so as to include ordinary nodes. There might be an ordinary node that is not yet chosen to serve as a clusterhead so far but it could be the best potential one to be elected.

Suppose z be the number of 1-hop neighbour of a mobile node, each mobile node needs to send up to z messages to identify its connectivity. In addition to this, every mobile node is expected to send at least one message to claim its information, including the mean connectivity degree, its weight, and its consumed energy. It is unlikely to say that a mobile node broadcasts only once cluster-related messages due to overlapping cluster structure. As a result, each mobile node on average broadcasts m such messages. Consequently, each mobile node requires to send out at least $(z + m + 1)$ messages for cluster formation. Hence, the communication complexity of Yang's algorithm can be

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication Pvt. Ltd.

represented as $O[(z+m+1)|V|]$ where V is the total number of mobile nodes available in the network.

S. Leu and R.S Chang [39], proposed a weight-value algorithm for clustering in MANET. The algorithm is used to select a dominating set based on the received radio signal strength. Each node sends a "Hello Message" to gain its 1-Hop neighbour periodically. The variation of radio signal strength between the nodes is measured as; $\Delta S_{ij} = \frac{s_{ij}^{t+1} - s_{ij}^{t}}{t}$. Then, the link weight between two neighbouring mobile nodes is calculated as; $l_{ij} = \frac{1}{(1-\Delta S_{ij})}$. And the weight-value of a node i is given as; $W_i = \sum_{i=1}^{n} l_{ij}$. Accordingly, a node with the highest weight value is selected as a dominator node.

The proposed algorithm performs better in addressing both static and dynamic nature of MANETs. However, the size of CDS nodes might be large as the aim of the researchers are on producing strongly connected cluster nodes. That means if two 1-hop neighbouring nodes, one with highest connectivity and moderate signal strength while the second with high signal strength and less connectivity are there, then their algorithm chooses the later node as a clusterhead. But if the first node is chosen, its signal strength is good as well as it reduces greatly the size of the network. This may reduce the battery power and bandwidth consumption of the network. Although the algorithm improves the link quality, the computational overhead increases as the weight value of each mobile node changes with the movement of neighbour nodes resulting in a re-calculation of a dominating set. This might result in a huge battery consumption and bandwidth degrading network performance.

Let Δ be the number of directly linked neighbours that a mobile node can have, then each node needs to send out Δ message in order to detect the radio signal variation with its 1-hop neighbours. Each mobile node is expected to send its own information at least once to compete with its neighbours. However, due to the overlapping cluster topology, nodes may send such messages more than one time and let's say on average m. In the tree connection phase, minimum spanning tree (MST) protocol is referred to construct CDS which requires $O(nlogn)$ messages to be exchanged. Thus, the overall communication complexity of Leu's algorithm can be specified as $O[(\Delta+m+1+logn)|n|]$ where n represents the number of nodes in the network. Similarly, its time complexity is almost identical to its message complexity.

Table 3. Summary of objectives, advantages and disadvantages of clustering schemes

Clustering Scheme		Size of CH	Specific Objectives	Advantages	Disadvantages
Node ID-Based	LIDAR [14]	Low	To overcome the drawbacks of LID by re-assigning lower ID to the most suitable node at the end of each HP based on mobility and energy level of a node	Chooses a node with low mobility and high energy level to be cluster head and hence prolonged network lifetime	Frequent ID re-assignment and re-clustering at the end of each HP
	Baker [16]	Low	To propose a connected network with self-starting capability distributively	Reducing clusterhead overlapping and self-organizing network	Maximum ID nodes are prone to battery drainage and lack of load-balancing
Node Density-Based	Victor [17]	Low	To elect a strategically important node as a starting node depending on the application and need unlike maximum density approach	Produce small size of network by starting from the most suitable node based on the scenario needed	Frequent clusterhead re-calculation due to high node re-affiliation which degrades network stability
	Gerla [13]	Low	To reduced the size of the network as much as possible	Generates a reduced size of CDS nodes	Low throughput and frequent clusterhead re-election
	SI [20]	Lowest	To generate the smallest possible CDS nodes with mobility handling capability	Generates the smallest size of CDS nodes and avoids tree connection phase opposed to MI	Single-point failure at a single-initiator leads to whole re-calculation
	MI [20]	Low	To avoid single-point failure at a single-initiator replacing it with a multi-initiator	Localized and low maintenance overhead and avoids single-point failure at SI	Increased number of initiators may increase the size of the network
Power-Aware	Wu [6]	Low	To provide the minimum connected CDS nodes by removing the redundant DS nodes with low energy level thereby decreasing the overall energy consumption of the network	A relatively higher residual energy node remains in CDS list, good for densely populated with low mobility	Unbalanced energy consumption between DS and ordinary nodes and unbalanced load distribution
	Xiong [24]	Low	To reduce energy consumption of the by minimizing the	Replacing an energy exhausted CDS nodes	High communication overhead during CDS construction as

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication Pvt. Ltd.

			number of CDS nodes and providing efficient scheduling scheme (MEBS) to avoid network partition	prior to creating network partition thereby maintaining connectivity	well as energy consumption difference between CDS and ordinary nodes
	Rai [25]	Low	To reduce the overall energy consumption of the network by minimizing the number of CDS nodes and introducing energy aware local maintenance	Energy-efficient repairing of a broken link	High network overhead during cluster formation and unbalanced energy consumption between mobile nodes
Mobility-Aware	MOBIC [26]	Low	To provide a stable cluster topology by reducing the rate of re-affiliation and re-clustering depending on the relative mobility of mobile nodes in the network.	Reduces the rate of re-clustering by choosing a relatively low speed mobile node	High mobility of clustermember nodes may lead to frequent topology change degrading network performance
	DGMA [28]	Medium	To guarantee a more stable cluster structure by reducing re-affiliation and re-clustering using group mobility behaviour based on mobile node's speed and direction	Has a better performance in networks with similar mobility behaviour	Its performance degrades in case the speed and direction of nodes are highly uncorrelated
Weighted Metric-Based	WCA [29]	Medium	To Choose the most suitable clusterheads in local areas by considering several metrics and maintaining stability of the network by reducing the rate of re-clustering	Flexible and applicable in different scenario as needed, re-clustering is not frequently invoked	Unbalanced power consumption between nodes and poor delta and remaining energy level estimation
	SLWCA [30]	Medium	To choose suitable clusterheads based on weight metric in order to increase network stability and provide load-balanced network	Better cluster member estimation and hence better load-balancing	Mobility is not considered in stability formulation, and the remaining energy level estimation is not quantified
	WBCA [37, 38]	Medium	To elect a suitable clusterhead based on its weight and providing a balanced energy consumption among mobile nodes	A relatively better consumed power estimation for serving cluster head	The estimated power formula may not be representative, and unbalanced load distribution
	S.Leu [39]	Medium	Electing the most suitable clusterheads locally by considering its weight value thereby guaranteeing strongly connected clustermembers	Strongly connected neighbours and hence, high link quality	High overhead due to frequent clusterhead re-election at the end of each "Hello message" which may result in a frequent topology change

Table 4. Summary of cost comparison of clustering schemes

	Maintenance cost	Ripple effect of re-clustering	Explicit control message for clustering	Stationary assumption for clustering	Message complexity	Time complexity
LIDAR[14]	High	No	Yes	Yes	$O[(n+\delta+1)\|V\|]$	unbounded
Baker[16]	Medium	Yes	Yes	Yes	$O(n*\delta+\beta+1)\|V\|$	unbounded
Victor[17]	High	Yes	Yes	Yes	$O(n^4)$	unbounded
Gerla[13]	High	Yes	Yes	Yes	$O(\beta+1)\|V\|$	unbounded
SI [20]	High	Yes (in case of initiator failure)	Yes	No	$O(2\|V\|)$	2-rounds
MI [20]	High	Yes (in case of initiator failure)	Yes	Yes	$O(2\|V\|+b*d)$	unbounded
Wu [6]	Medium	Yes	Yes	Yes	$O(2\|n\|)$	2-rounds

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication Pvt. Ltd.

Xiong[24]	High	No	Yes	Yes	$O(x^3)$	$O(x^3)$
Rai [25]	Low	No	Yes	Yes	$O[(3+C)\|V\|]$	unbounded
MOBIC[26]	High	Yes	Yes	No	$O(2n+m+1)\|V\|$	2-rounds
DGMA[28]	Lower	Yes	Yes	No	$O(\Delta+n+1)\|V\|$	$O(n)$
WCA [29]	Medium	Yes	Yes	Yes	$O(\Delta+x+1)\|V\|$	unbounded
SLWCA[30]	Medium	Yes	Yes	Yes	$O(\beta+m+1)\|V\|$	unbounded
Yang [37, 38]	Medium	Yes	Yes	Yes	$O(z+m+1)\|V\|$	unbounded
Leu [39]	High	Yes	Yes	Yes	$O(\Delta+m+1+logn)\|n\|$	$O(n+nlogn)$

VII. CONCLUSION

In the proposed energy-aware clustering algorithms, energy consumption is compared in two phases. In the cluster formation phase, Wu, Xiong and Rai are all aimed at constructing minimum number of CDS nodes through which the overall reduced energy consumption is achieved. This means that minimizing the number of CDS participants will greatly reduce power consumption of the network as CDS nodes consumes more energy because of many duties assigned to them. In the maintenance phase, Wu tried to remain a relatively high residual energy nodes in CDS list while Rai localized maintenance of a broken link which in turn reduces energy consumption. But Xiong's cluster maintenance is focused on avoiding energy depleted nodes prior to creating network disconnection. Although the three algorithms are focused on maximizing network lifespan, balancing the great energy consumption difference between CDS and ordinary nodes are not yet seen.

In node ID-based clustering algorithm, the system performance is better as compared with density-based heuristic in terms of throughput in a mobile environment. Although Gavalas's algorithm outperforms that of Baker's, it is not easy to assign new node IDs at the end of each hello period. However, their algorithm suffers from two major draw backs. One is power drainage problem as it is biased to certain nodes will affect severely the network lifetime. Secondly, loads are not uniformly distributed throughout the network and hence load-balancing problem is inevitable.

Although Victor's and Gerla's algorithm is focused on generating minimum size of the network, the throughput of the system degrades as the number of nodes increased. Moreover, their algorithm may overload clusterheads which cannot basically handle more than a limited number of nodes at a time and hence it is bounded. Moreover, in a dynamic network where the number of node affiliation is high, clusterhead changes frequently which increases network instability. Even in the presence of one link change due to node movement, it may fail to be re- elected as a clusterhead. This may also increase the network overhead and battery power consumption. In MOBIC, the selection of low mobility node improves the network performance by reducing the rate of re-clustering. However, its performance degrades if the mobility of cluster member nodes changes steadily. On the other hand Zhang's algorithm improves the drawback of MOBIC in that it is useful and applicable in groups having similar characteristics like related mobility and direction as in highway traffic. But its performance degrades in case a mobile nodes' related speed and direction are highly uncorrelated. Although M. Chatterjee introduced δ for load-balancing among clusterheads, its value is not estimated in a convincing manner (he assumed it to be ideally 2). Yang tried to estimate δ which varies from one node to another based on local connectivity. This may not be realistic as it is not related to that of the whole network. However, M. Aissa tried to quantify it better which varies from one clusterhead to another which is inter-related by pointing out the fact that nodes are not uniformly distributed throughout the network. Hence, load-balancing is handled better in Aissa's algorithm than in Chaterjee's and Yang's algorithm. In Chaterjee and Aissa, the remaining energy level of a node is not yet estimated while Yang's consumed power estimation for serving clusterhead is a better approach, yet it is not representative. S.Leu's Algorithm provides a strongly connected neighbourhood and hence guaranteed link quality. But frequent re-clustering at the end of every hello message increases network overhead. However, in the case of Chaterjee's algorithm, re-clustering is not invoked frequently compromising for link quality.

REFERENCES

1. Sze-Yao Ni, Yu-Chee Tseng, Y.S. Chen, and J.P. Sheu, The broadcast storm problem in a mobile ad hoc network, In Proc. of the 5th ACM/IEEE Int. Conference on Mobile Computing and Networking (MobiCom '99), pages 151–162, New York, NY, USA, 1999.
2. C.R. Lin and M. Gerla, "Adaptive Clustering for Mobile Wireless Networks," IEEE JSAC, vol. 15, Sept. 1997, pp. 1265–75.
3. C.R. Lin and M. Gerla, "A Distributed Control Scheme in Multi-hop Packet Radio Networks for Voice/Data Traffic Support," Proc. IEEE ICC'95, pp. 1238–42, June 1995.
4. T.C. Hou and T.J. Tsai, "An Access-Based Clustering Protocol for Multihop Wireless Ad Hoc Networks," IEEE JSAC, vol. 19, no. 7, pp. 1201–10, July 2001.
5. Yu .J.P and Chong P.H.J, A Survey of Clustering Schemes for Mobile Ad Hoc Networks, IEEE Communications Surveys and Tutorials, Vol. 7, No. 1, pp. 32-48, 2005.
6. J. Wu, Ming Gaol, and Ivan Stojmenovic, "On Calculating Power-Aware Connected Dominating Sets for Efficient Routing in Ad Hoc Wireless Networks," J. Commun. and Networks, vol. 4, no. 1, pp. 59–70, Mar. 2002.
7. J.H. Ryu, S. Song, and D.H. Cho, "New Clustering Schemes for Energy Conservation in Two-Tiered Mobile Ad-Hoc Networks," in Proc. IEEE ICC'01, vo1. 3, pp. 862–66, June 2001.
8. M. Chatterjee, S.K. Das, and D. Turgut, "An On-Demand Weighted Clustering Algorithm (WCA) for Ad hoc Networks," in Proc. IEEE Globecom'00, pp. 1697–701, 2000.
9. A.B. McDonald and T.F. Znati, "Design and Performance of a Distributed Dynamic Clustering Algorithm for Ad-Hoc Networks," in Proc. 34th Annual Simulation Symp., pp. 27–35, Apr. 2001.
10. T. Ohta, S. Inoue, and Y. Kakuda, "An Adaptive Multihop Clustering Scheme for Highly Mobile Ad Hoc Networks," in Proc. 6th ISADS'03, Apr. 2003.
11. D.J. Baker and A. Ephremides, A distributed algorithm for organizing mobile radio telecommunication networks, in: Proceedings of the 2nd International Conference on Distributed Computer Systems, pp. 476–

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication Pvt. Ltd.

483, April 1981.
12. A. Ephremides, J.E. Wieselthier and D.J. Baker, A design concept for reliable mobile radio networks with frequency hopping signaling, in: Proceedings of IEEE, Vol. 75(1) (1987) 56–73.
13. M. Gerla and J.T. Tsai, “Multiuser, Mobile, Multimedia Radio Network,” Wireless Networks, vol. 1, pp. 255–65, Oct. 1995.
14. D. Gavalas, G. Pantziou, C. Konstantopoulos, Basilis Mamalis, Stable and Energy Efficient Clustering of Wireless Ad-Hoc Networks with LIDAR Algorithm, P. Cuenca and L. Orozco-Barbosa (Eds.): PWC 2006, LNCS 4217, pp. 100–110, 2006.
15. CH.V. Raghavendran, G. Naga Satish, P. Suresh Varma, I.R. Krishnam Raju , Enhancing the Performance of Routing in Mobile Ad Hoc Networks using Connected Dominating Sets, Special Issue of International Journal of Computer Applications, pp. 0975 – 8887, 2012.
16. D.J. Baker and A. Ephremides, The architectural organization of a mobile radio network via a distributed algorithm, IEEE Transactions on Communications COM-29 11 (1981) 1694–1701.
17. C.S. Victor and G. Amalanathan, Construction of Strategic Connected Dominating Set For Mobile Ad Hoc Networks, Journal of Computer Science 10 (2): 285-295, 2014.
18. Yongsheng Fu, Xinyu Wang, Shanping Li, Construction K-Dominating Set with Multiple Relaying technique in Wireless Mobile Ad Hoc Networks, IEEE Computer Society; International Conference on Communications and Mobile Computing, 2009.
19. Ling Ding et al, Distributed Construction of Connected Dominating Sets with Minimum Routing Cost in Wireless Networks, IEEE Computer Society;International Conference on Distributed Computing Systems, 2010.
20. Kazuya Sakai, Scott C.H. Huang, Wei-Shinn Ku, Min-Te Sun, and Xiuzhen Cheng, Timer-Based CDS Construction in Wireless Ad Hoc Networks, IEEE Transactions on Mobile Computing, Vol. 10, No. 10, October 2011.
21. R. Zheng and R. Kravets, “On-demand Power Management for Ad Hoc Networks,” in Proc. IEEE Infocom’03, vol. 1, pp. 481–91, Mar.–Apr. 2003.
22. J. Gomez et al., “PARO: Supporting Dynamic Power Controlled Routing in Wireless Ad Hoc Networks,” Wireless Networks, vol. 9, pp. 443–60, 2003.
23. J. Wu and H. Li. On calculating connected dominating set for efficient routing in ad hoc wireless networks. Proc. of the 3rd Int ’1 Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications, pages 7-14, 1999.
24. Naixue Xiong, Xingbo Huang, Hongju Cheng, and Zheng Wan, “Energy-Efficient Algorithm for Broadcasting in Ad Hoc Wireless Sensor Networks” Sensors 2013, 13, 4922-4946; doi: 10.3390/s130404922.
25. Mritunjay Rai, Shekhar Verma, Shashikala Tapaswi, A Power Aware Minimum Connected Dominating Set for Wireless Sensor Networks, Journal of Networks, Vol. 4, No. 6, August 2009.
26. P. Basu, N. Khan, and T.D.C. Little, “A Mobility Based Metric for Clustering in Mobile Ad Hoc Networks,” in Proc. IEEE ICDC-SW’01, pp. 413–18, Apr. 2001.
27. C.C. Chiang, H.K. Wu, W. Liu, M. Gerla, “Routing in Clustered Multihop, Mobile Wireless Networks with Fading Channel,” Proceedings of SICON 1997.
28. Yan Zhang, Jim Mee Ng, Chor Ping Low, A distributed group mobility adaptive clustering algorithm for mobile ad hoc networks, Elsevier, Computer Communications, Vol. 32, pp. 189–202, 2009.
29. M. Chatterjee, S. Dasand D. Turgut, “WCA: a weighted clustering algorithm for mobile ad hoc networks,” Journal of Cluster Computing (Special Issue on Mobile Ad hoc Networks), 5, pp.193-204, 2002.
30. Mohamed Aissa, Abdelfettah Belghithb and Khalil Drira, New strategies and extensions in weighted clustering algorithms for mobile Ad Hoc networks, Procedia Computer Science 19 (2013) 297 – 304.
31. M. Amine Abid, Abdelfettah Belghith, Stability routing with constrained path length for improved routability in dynamic MANETs, The International Journal of Personnal and Ubiquitous Computing, Springer; Volume 15, Issue 8 (2011).
32. S.K.B. Rathika and J. Bhavithra, An Efficient Fault Tolerance Quality of Service in Wireless Networks Using Weighted Clustering Algorithm; Bonfring International Journal of Research in Communication Engineering; Vol. 2, Special Issue 1, Part 4; February 2012.
33. W. Jin,et al. A load-balancing and energy-aware clustering algorithm in wireless ad-hoc networks; Embedded and Ubiquitous Computing EUC 2005 Workshops; Lecture Notes in Computer Science Volume 3823; pp 1108-1117; 2005.
34. Xi'an Jiaotong, et al. WACHM: Weight Based Adaptive Clustering for Large Scale Heterogeneous MANET; Communications and Information Technologies; ISCIT '07; 2007.
35. Hui Cheng, Jiannong Cao, Xingwei Wang, Sajal K. Das, Shengxiang Yang, Stability-aware Multi-Metric Clustering In Mobile Ad Hoc Networks With Group Mobility, Wireless Communications and Mobile Computing; 9:759 771, Published online 21 April 2008 in Wiley InterScience.
36. Christian Bettstetter, The Cluster Density of a Distributed Clustering Algorithm in Ad Hoc Networks, IEEE International Conference Communications; Page(s): 4336 - 4340 Vol.7; 2004.
37. Wei-dong Yang, Guang-zhao Zhang, A Weight-Based Clustering Algorithm for mobile Ad Hoc network, IEEE Computer Society, 2007.
38. Yang Wei-dong, Weight-Based Clustering Algorithm for Mobile Ad Hoc Network, IEEE: Cross Strait Quad-Regional Radio Science and Wireless Technology Conference; July 26-30, 2011.
39. Seiven Leu, Ruay-Shiung Chang, A weight-value Algorithm for Finding Connected Dominating Sets in a MANET, Journal of Network and Computer Applications, Vol. 35, pages 1615-19, March 2012.

N. Fraol Bekana, received Bachelor degree in Electrical-Electronics Technology from Adama University in 2004 and M.Sc. in Telecom Engineering from College of Telecommunications and Information Technology, Addis Ababa in 2009. He has been a faculty member in Electrical and Computer Engineering department, Arbaminch University, since 2004. He worked as head of department and Engineering consultancy in Arbaminch Institute of Technology. Currently Ph.D student in NIT, Silchar, India. His research interest includes wireless ad hoc networks and PLC.

Jeshan Mehedi, received the B.Tech degree in Electronics and Communication Engineering from University of Kalyani in 2002 and the M.E degree in Electronics and Telecommunication Engineering from Jadavpur University in 2007. He worked as lecturer in Dumkal Institute of Engineering and Technology from 2002 to August, 2008. Now he is working as Ass. Prof in ECE department at NIT, Silchar. His research interest includes Wireless Sensor Networks, Digital System Design.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication Pvt. Ltd.