J Netw Syst Manage (2009) 17:183–207
DOI 10.1007/s10922-009-9123-7

A Survey on One-Hop Clustering Algorithms in Mobile Ad Hoc Networks

Suchismita Chinara · Santanu Kumar Rath

Published online: 9 May 2009

Abstract Clustering in mobile ad hoc network (MANET) play a vital role in improving its basic network performance parameters like routing delay, bandwidth consumption and throughput. One-hop clustering scheme adopts the simple mechanism to make the logical partition of the dynamic network where the network topology changes constantly resulting an unstable clustering. This paper makes a comprehensive survey of some bench-mark one-hop clustering algorithms to understand the research trends in this area. The literature provides the logic of cluster formation for different algorithms in achieving a linked cluster architecture and an intensive simulation survey of their performance on the cluster maintenance aspects such as cluster density, frequency of cluster reelection, frequency of cluster changes by the nodes and the granularity of cluster heads. This paper should facilitate the researchers as well as practitioners in choosing a suitable clustering algorithm on the basis of their formation and maintenance overhead, before any routing scheme is adopted in the mobile ad hoc network.

Keywords Linked cluster architecture · Cluster head · Re-affiliation · Cluster stability · Cluster maintenance

1 Introduction

With the advent in wireless communication and reduction in price of personal communication devices like laptops and Personal Digital Assistants (PDAs), the

S. Chinara (✉) · S. K. Rath
Department of Computer Science and Engineering, National Institute of Technology, Rourkela, Orissa 769008, India
e-mail: suchi.nitrkl@gmail.com

S. K. Rath
e-mail: rath.santanu@gmail.com

internet services have reached every corner of the world. The demand of instant networking services has been tremendously increased in the areas of education, entertainment, business centers and emergency services. Moreover, people want to be in touch with each other while even on-the-fly. Most of the wireless communications depend on a wired backbone, where the fixed base stations are bridged to provide the services within a constrained geographical boundary. But the deployment of such network requires cost and time which may not be possible to provide in utmost emergency. As a result the availability and reliability of internet is not guaranteed everywhere and every time. This led to the development of peer-to-peer network or ad hoc network where people can communicate among each other without the support of any fixed infrastructure even while in motion [1]. The mobile ad hoc network (MANET) needs no time and cost for its deployment because every node plays the role of a router along with its job as an ordinary host. Some of the challenges involved in MANET are the scarce in bandwidth, limited battery power of the nodes and frequent topology change of the networks [2]. Various management solutions for such a complex and dynamic environment of ad hoc network are described in [3].

Dynamic routing being a network layer issue is an interesting research area in MANET. Several routing protocols [4] have been developed so far to handle the multi hop, self organizing network. For both the proactive and reactive scheme, routing can take place either in a flat structure or in a hierarchical structure [2]. It has been proved that the packet delivery delay is reduced in a hierarchical structure with that of in the flat structure [5] resulting in a better routing efficiency. In large networks, the flat routing structure produces excessive information flow which can saturate the network. Hierarchical routing is an interesting solution for handling such a scalable network where only selected nodes take the responsibility of routing [6]. Thus, topology management plays a vital role prior to the actual routing in MANET. Cluster based structure, which is a typical synonym of the hierarchical structure in network topology has become popular since last decade to improve the routing efficiency in a dynamic network. It proceeds with the formation of a virtual cellular backbone (VCB), [7] where the cluster heads map to the base stations in cellular architecture. A literature survey of some clustering protocols (both one-hop and multi-hop) is presented in [8] where the classification is made on the basis of their different objectives. Another survey of clustering algorithms in MANET as well as in wireless sensor networks (WSN) has been presented in [9] in terms of the graph theoretic approach. The algorithms are grouped according to the use of different types of dominating set approaches like independent dominating set (IDS), weakly connected dominating set (WCDS) and connected dominating set (CDS) in setting up the virtual back bone in MANET. Both of the survey literatures make a theoretical study of the algorithms according to their categories, but do not provide any simulation survey to study the performance results of the existing algorithms.

This paper presents a survey for the basics of one-hop cluster formation algorithms in MANET and an intensive simulation study on their performance that decide the overhead of cluster formation as well as network maintenance. The organization of the literature is as follows. Section 2 describes the cluster control structure in detail with illustrative examples. Section 3 starts with the history of clustering algorithms followed by the overview of existing one-hop clustering algorithms and provide the simulation results of individual algorithms as well. An overall comparative simulation study of the algorithms and their flaws and strengths are described in Sect. 4. Finally, Sect. 5 concludes the paper.

2 Clustering in Mobile Ad Hoc Network

2.1 Definition

Clustering in mobile ad hoc network can be defined as the virtual partitioning of the dynamic nodes into various groups. Groups of the nodes are made with respect to their nearness to other nodes. Two nodes are said to be neighbor of each other when both of them lie within their transmission range and set up a bidirectional link between them. Depending on the diameter of the clusters there exist two kinds of cluster control architectures, known as *one-hop* clustering and *multi-hop (d-hop)* clustering. In one-hop clustering every member node is at most 1-hop distance away from a central coordinator called as the cluster head. Thus all the member nodes remain at most two hops distance away from each other within a logical cluster. But in multi hop clustering, [10–12] the constraint of immediate neighborhood of members from the head is eliminated by allowing the nodes to be present at most d-hop distance away from each other to form a cluster. Typical mobile ad hoc network is shown in Fig. 1 with flat and hierarchical structure.

The small circles in the figure represent the wireless nodes in the network. The lines joining the circles denote the connectivity among the wireless nodes. Every node is identified with an ID number (i.e.1–14) along with a numeral within parenthesis. The numbers in the parenthesis indicates weights of the nodes. These weights are calculated with respect to various node parameters and take part in the selection of cluster heads. The weight calculation basis for different algorithms is discussed later in the article. It is visible from the figure that in the flat architecture of MANET every node bears equal responsibility to act as a router for routing the

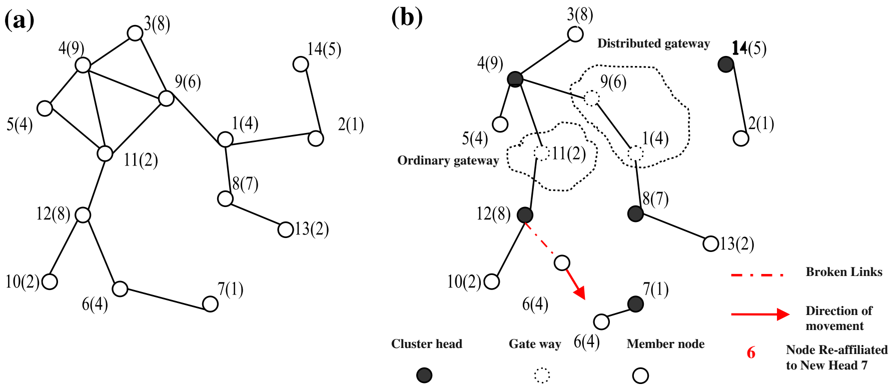


Fig. 1 Nodes in flat and hierarchical structure. **a** Flat structure of nodes. **b** Hierarchical structure of nodes

packets to every other node. So a great amount of message flooding takes place in order to obtain better routing efficiency. In return such message flooding reduces the MAC layer efficiency to certain extent. Cluster control structure can be one possible solution to improve such MAC layer efficiency [2]. In the hierarchical structure nodes are assigned with different functionalities while acting as a cluster head or a cluster gateway or a cluster member as shown in the figure.

Cluster control structure forms the virtual backbone of communication where cluster heads are the communication hotspots. The cluster head works as the local coordinator for its member nodes and does the resource management among them similar to a base station of cellular architecture. These cluster heads are responsible for inter cluster and intra cluster communication. A gateway node is a node that works as the common or distributed access point for two cluster heads. When a node remains within the transmission range of two cluster heads as the node *11* in figure it is called as the ordinary gateway for two corresponding clusters. And a node having one cluster head as an immediate neighbor in addition to which it can reach a second cluster head in two hops as node *9* or *1* is a distributed gateway that is linked to another distributed gateway of other cluster. Both of the distributed gateways provide the path for the inter-cluster communication. The ordinary nodes of the cluster are the immediate neighbors of the cluster heads. They have the capability of serving as either a head or a gateway whenever selected to do so.

2.2 Clustering Requirement

Information flow among various nodes is the prime goal of any communication network. This includes the control information that is exchanged between the source and the destination prior to the actual data transmission and the data information that is routed by various routers to reach the destination. The overhead of information passing increases considerably when the routing nodes are mobile and the network topology changes frequently. This causes instability in the pre-established routes. By the cluster control structure, the topology updating information and routing information can be reduced to the exchange of aggregated information between various clusters and the exchange of detailed topology information to single clusters [13]. This concept localizes the traffic of updating and control messages leading to efficient bandwidth utilization in the ad hoc network.

In cellular network the mobile nodes communicate directly with the fixed base station reducing the wireless part of communication to a single hop problem. Many good solutions have been proposed for handling of mobility of the nodes by this base station. Thus the mapping of cellular architecture into peer to peer network leads to the concept of clustering [14]. In such a virtual cellular architecture (VCA) cluster heads are selected to play the role of base stations of the cellular structure. The cluster head along with its one-hop members form the virtual cells retaining the merits of cellular architecture. Scalability and bandwidth utilization in an ad hoc network have considerable effect on routing efficiency. Keeping track of neighbors in a dynamic network for efficient routing further increases the storage cost in a flat structure. The spatial reuse of major resources like the bandwidth is also reduced in case of a scalable network. Clustering is a proven solution for such situations. The cluster heads being responsible for storing and forwarding the packets need to keep the information regarding their neighbor heads or gateways within their close proximity. This enables to increase the scalability of the network. The bandwidth utilization is also increased as limited nodes take part in transmitting the packets. Energy issue is still an unsolved problem for wireless nodes. Specially, in a wireless sensor network [13] limited battery power is a major constraint. The cluster control structure provides solution for efficient energy management [15]. Cluster heads being the communication centers consume more battery power than non-cluster heads [16]. Further, the non-cluster heads may enter into sleep mode during their idle state reducing the energy consumption to a considerable amount.

2.3 Cluster Formation and Maintenance

The process of clustering can be visualized as a combination of two stages, i.e. cluster formation and cluster maintenance. The cluster formation phase deals with the logical partition of the mobile nodes into several groups and selection of a set of suitable nodes to act as heads in every group. In mobile ad hoc network where the topology changes frequently, selection of optimum number of cluster heads is a NP-hard [17] problem. Hence, there exists some representative algorithms (most of them are found to be greedy) that use the parameters like node identity number, mobility, battery power, degree of connectivity etc. as the factors to decide its suitability for cluster head. Even some researchers combine multiple parameters to select these set of routers [18, 19] in an efficient manner. These selected nodes are responsible for routing as well as to do node management in the mobile network and collectively called as the dominant set in graph theory terminology [20].

The objective of cluster maintenance is to preserve the existing clustering structure as much as possible. In one hop clustering since every node is directly connected to a cluster head, the mobility of either the member node or the cluster head may drive them away from each other. There exists a bidirectional link between these two nodes till both of them are within their transmission range. When any of them moves away from the other, there occurs a link failure and the member node searches for another new head within its transmission range to get affiliated. This kind of situation is called as re-affiliation to a new head node. A typical example is shown for node 6 in Fig. 1. Here, the node 6 moves (shown in the arrow mark) away from its current cluster head 12. When it is out of the transmission range of head 12, it finds the new head node 7 within its transmission range and gets itself affiliated to node 7 and becomes its new member. Thus the head nodes 12 and 7 update their member lists accordingly. A single re-affiliation causes several update messages to flow between both the old and new cluster heads.

The requirement for the reelection of cluster heads arises when the current heads fail to cover all the nodes in the network. Sometimes a node may move away from the transmission range of all the current cluster heads and becomes an orphan node. This demands a reelection of cluster heads. Even at times any of the cluster heads may drain out of energy or may even fail to work due to any fault occurrence and needs a head reelection process. However, such an unavoidable reelection increases the computation cost and the message complexity.

3 Clustering Algorithms and Their Simulation Results

The concept of partitioning the dynamic network into logical clusters was initially proposed by Baker and Ephremides [21]. They designed a self-starting, distributed algorithm to establish and maintain a connected architecture even with the node mobility or node failure. This algorithm is best suited for High Frequency Intra Task Force (HF ITF) communication network along with other communication networks such as packet radio network (PRNET), advanced mobile phone service (AMPS) network and battle field information distributed system (BID). The HF ITF network is a mobile and widely dispersed general purpose network that provides extended line of sight (ELOS: 50–1000 km) communication for naval task force units. Here the nodes are linked via radio waves from the HF band (2–30 MHz). This connectivity among the nodes is sometimes hampered by the variation in antenna pattern of the nodes, ground wave/sky wave interference, node movement, node failure as well as new node addition into the network. So there is a need for a self organizing reliable network structure that can be maintained under changing connectivity without the support of a central controller. Thus the authors of [21] proposed a new architecture called the linked cluster architecture where the network is organized into a set of node clusters and each node belongs to at least one cluster. Cluster heads of every cluster remain in direct communication range of member nodes resolving the hidden terminal problem by the existing busy tone multiple access (BTMA) technique [22].

The major five tasks associated with the implementation of linked cluster architecture [23] are *topology sensing, cluster formation, cluster linkage, link activation and finally routing*. To meet the above tasks without relying on a central coordinator, three algorithms were developed as:

1. Linked Cluster Algorithm (*LCA*),
2. Link Activation Algorithm (*LAA*) and
3. Routing Algorithm.

Linked Cluster Algorithm performs the job of initial three tasks such as topology sensing cluster formation and cluster linkage where as LAA and the routing algorithm cover the link activation and routing operations. We intend to focus on LCA as it describes the basis of neighborhood detection in changing topology and cluster formation.

Topology sensing is the initial step in implementing the linked cluster architecture where the nodes discover their neighbors by the method of probing as described in [23]. The authors indicate that cluster heads can be connected to each other directly or via the gateway nodes. For every sub-band that is formed based on the communication range and connectivity, LCA works in *two* TDMA frames where every TDMA frame is subdivided into N slots. Each node is identified by a unique integer from 1 to N and is allocated to transmit control channel messages during its own slot in each frame. For example, during the *i*th slot of first Frame, node *i* broadcasts its probe message (may be announcing its own identity) and acknowledges the receipt of previously transmitted probe messages that it has heard (by announcing the IDs of those nodes it has heard from so far) during the earlier slots of this frame. A binary CONNECTIVITY matrix stores a value of 1 in the (*i*, *j*) position to indicate the existence of a link between *i* and *j* where as a 0 indicates the lack of connectivity. Thus, at the end of this frame, node *i* fills in some of the entries of its CONNECTIVITY matrix. During second Frame, each node broadcasts acknowledgements for probe messages that it has received since the transmission of its own slot of first frame. Thus by the time of *i*th slot of second frame transmission, node *i* fills in the *i*th row of the connectivity matrix and has the complete knowledge of its neighbors.

After the topology sensing, the cluster formation phase of *LCA* uses the rule that the node with the highest identity number among its neighbors is the first candidate to claim cluster head status. A node first checks its connectivity row in the matrix and if no neighbor with higher identity number is found, it becomes the cluster head. If one or more neighbors exist with higher identity number, the highest numbered neighbor becomes the cluster head. Thus at the end of the second frame the nodes know their NODESTATUS and broadcast it for the implementation of the next task of link cluster architecture, i.e. linking of clusters by *LAA*.

The LCA uses a very simple cluster head selection strategy for its implementation. However, this TDMA approach suffers from the fact that every node needs to maintain an accurate global time among them which is possible only by the presence of a central timer. Moreover, partitioning the frames into number of slots is typically not possible as the node numbers in a dynamic network can not be known a priori. The identity of the nodes being the deciding factor for the cluster heads, the method of number assignment to the nodes become very crucial part of the proposed organization. The higher numbered nodes have a greater tendency to become heads unless it is completely covered by another cluster head. Further, the situation of node/link addition and deletion in the clusters (i.e. the cluster maintenance) is not being considered by the authors. In a nutshell, LCA could not meet certain criteria of the ad hoc network, but could become the *base algorithm* for other benchmark algorithms.

This paper focuses on the cluster formation phase of the linked cluster architecture while explaining the algorithms with their basis of cluster formation as well as the cluster maintenance in the presence of node mobility. The algorithms are described in the order of their development.

We simulate the existing algorithms with *N* nodes on a 100 × 100 grid. The transmission range T_{range} for all the nodes are kept equal for individual experimental setup. The nodes are assumed to move with a constant speed *Disp* and follow the Random Walk mobility pattern during the simulation. This mobility model [24–26] reflects the most erratic and unpredictable movement of an entity. Here, a mobile node moves from its current location by choosing a random speed *Disp* that varies between (*speed* min, *speed* max) and a random direction between $(0, 2\pi)$ respectively. In random walk model when a mobile node reaches a simulation boundary, it bounces back with an angle determined by the incoming direction. This is a memory less mobility pattern as it retains no knowledge about its past location and speed value. The value of *speed* min and *speed* max are set to 0 and 5, respectively.

As discussed earlier, cluster maintenance in the ad hoc networks play a vital role similar to the cluster formation. Once the clusters are formed with one cluster head and zero or more members, the next job is to ensure the availability of a well connected network for routing. A node must get connected to nearby cluster head so that the network resources can be made available to it all the time. If it does not find any head node within its transmission range, then it declares itself as the cluster head. The characteristics of any clustering algorithm can be analyzed by some performance metrics. In order to have a simulation study on the existing weight based one-hop clustering algorithms of MANET, we consider some metrics like Mean Cluster Population (MCP), Rate of cluster changes by a node (RA) and Rate of reelection of the cluster heads (RE). Out of these metrics *MCP* is responsible for packet delivery delay and Quality of the Service (QoS) of the network where as *RA* and *RE* decides the maintenance over head. The definitions of the resulting metrics are as:

- *Mean Cluster Population (MCP)* This defines the average number of logical partitions formed in the network with the mobile nodes. Every cluster is headed by a single cluster head. This defines the cardinality of the dominant set and it ranges as $1 \leq \mathrm{MCP} \leq \mathrm{N}$, where N is the number of nodes in the network. A *lower* MCP is desirable so that the routing delay is reduced [27].
- *The rate of cluster changes (RA)* This defines the average number of times a node switches from its existing cluster head to another cluster head. This is also called as the *re-affiliation rate* [18] by the nodes. A *lower RA* reduces bandwidth consumption during message exchanges for member updation.
- *The rate of Reelection (RE)* This defines the average number of times an updation takes place to select the members of the dominant set (i.e. cluster heads) [18]. A lower reelection rate *RE* implies a better *cluster stability*.

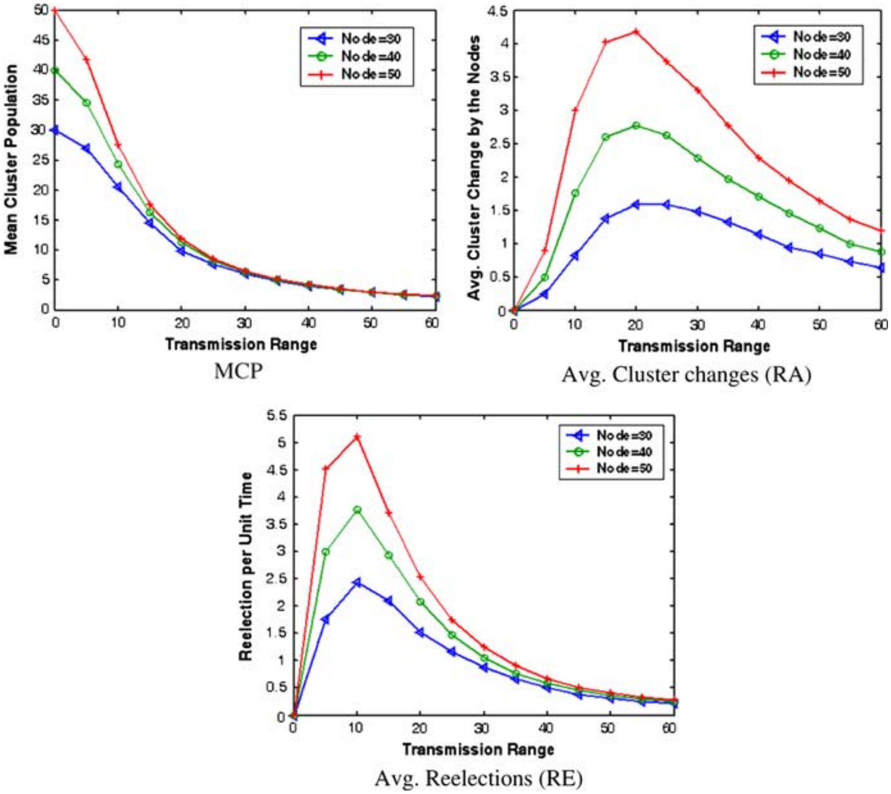
3.1 Lowest ID (LID) Algorithm

A small variation to LCA [21] was proposed by Ephremides, Wieselthier and Baker in [28] and was named as LID algorithm. In this algorithm, every node is assigned with a unique non-negative identification number ID which is the deciding factor for the status of a node. In a mobile packet radio network, a node has no a priori knowledge of the locations of other nodes, the connectivity of the network, nor even of its own neighbors. So as a first task when the network comes up, the connectivity among the nodes is discovered by every other node. This is accomplished as every single node broadcasts it's ID to its neighbors and receives the same from its neighbors. (The term neighbor defines the set of nodes that can hear the messages directly.) If a node listens to all the neighbors' IDs that are higher than its own ID, then it declares itself as the cluster head among its immediate neighbors. And the neighbor nodes whose status is not yet decided become the members of the newly selected head. This process is repeated till all the nodes are assigned with the role of a head or a member of a cluster. This algorithm does not allow two cluster heads to be neighbors, so the gateway nodes provide the path for inter cluster communication. TDMA based probing message exchange is used here to ensure a contention free communication. It does not target to achieve minimum number of clusters, but ensures to produce a connected network.

LID retains its utility as a benchmark, for producing reasonably stable cluster control architecture as discussed by Gerla and Tsai in [29] where the stability of clusters is much better than that of the contemporary algorithms [5]. However, as node ID is the only deciding factor for a node to be a cluster head, the lower ID nodes are biased to become the heads all the time resulting in faster battery drainage that perturbs the cluster stability.

The performance results for Lowest ID (LID) algorithm are shown in Fig. 2. Mean Cluster Population (MCP) depicts the average number of clusters formed with the variation of the transmission range from 0 to 60. Here, the N is varied between 30, 40, and 50. *MCP* reduces with the increase in transmission range. Two cluster heads are not allowed to remain neighbors of each other, so one node is forced to resign from its current status. This increases the number of reelections *RE* and the number of Re-affiliations *RA* by the member nodes.

As in figure there occurs considerable rise in *RA* for the transmission range of 10–20 and decreases slowly with increase in transmission range. This happens because the lower transmission range reduces the size of the cluster zone keeping most of the nodes at the boundary, so that even for a slight movement they may leave the current cluster boundary and join other clusters as their new members.



MCP

Avg. Cluster changes (RA)

Avg. Reelections (RE)

Fig. 2 Performance results of lowest ID algorithm (LID)

With the increase in the range the cluster head periphery is increased to accommodate more members within it. So the *RA* is reduced with increased transmission range. The average number of reelections *RE* that takes place at lower transmission range is more in comparison to that at high transmission range because at lower transmission range MCP is high and the probability that two cluster heads will come closer to each other is quite high. Thus a considerable amount of node head resignation (due to the non-neighborhood constraint) takes place that results more reelections RE of the heads.

3.2 Highest Connectivity Algorithm (HC)

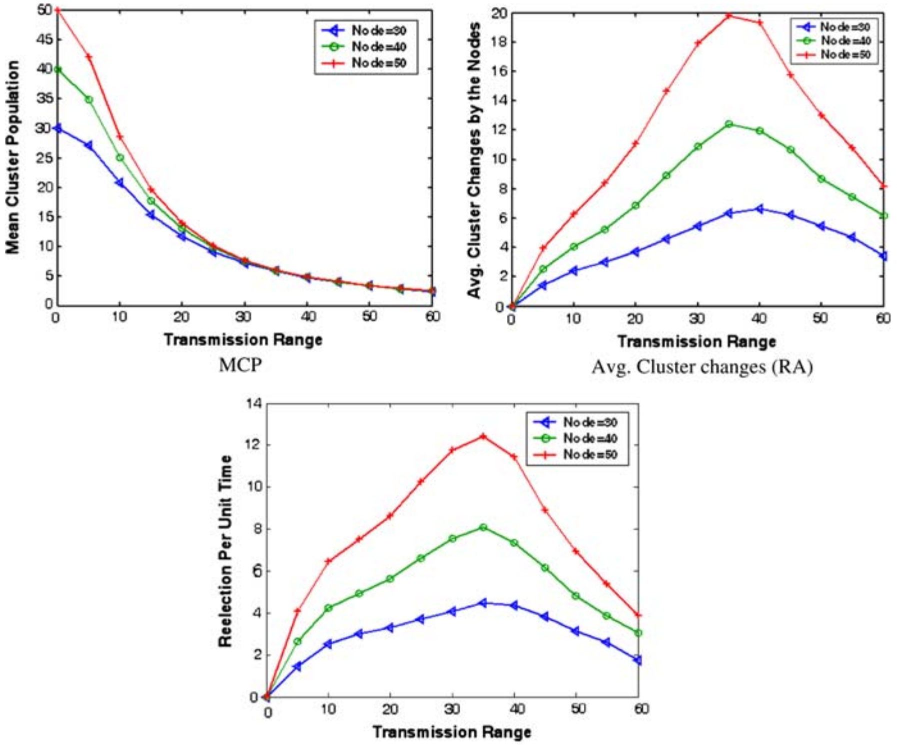
A modified version of LCA named as Highest Connectivity Algorithm was proposed by Parekh [30] with an aim to reduce the number of clusters in the network. If $\{N_i\}$ represents the set of adjacent nodes of a particular node i, then the degree of connectivity of i is represented as $D_i = |N_i|$, where $|N_i|$ is the cardinality of $\{N_i\}$. In this algorithm a node having highest degree of connectivity D_i is selected as the cluster head. And the adjacent node whose status is not yet decided becomes the member of the selected cluster head. A higher degree of connectivity ensures efficient service to the member nodes by minimizing the number of heads. Here the efficiency means lowering the delay in communication through the head nodes.

However, this algorithm results in low throughput. This happens because every cluster is assigned with some resources that are shared among all the nodes in the cluster in a near equal manner. So an increased number of nodes in a cluster reduce the throughput and finally the system performance is degraded. Moreover, the mobility of nodes changes the degree of connectivity of the node very frequently leading to more number of cluster head reelections as well as link updations. Thus the maintenance cost of the clusters is worse than the configuration counterpart. The poor cluster stability of the algorithm decreases its application in real world situation in spite of its reduced delay in data communication.

The performance result for the highest connectivity algorithm (HC) has been depicted in Fig. 3 where the population of clusters *MCP* reduces as the transmission range of the nodes increases. This happens because the higher transmission range enhances the degree of connectivity of the nodes and more number of nodes can emerge within a single cluster. Being connectivity based algorithm, HC suffers from greater reelection rate *RE* and the number of cluster changes *RA* by the nodes as the mobility of the nodes changes their degree of connectivity. Even for a single connectivity change the weight of the head may change enforcing a reelection to occur.

3.3 Mobility Metric Based Algorithm (MOBIC)

A mobility metric based version of lowest ID algorithm MOBIC was proposed by Basu, Khan and Little [31]. The algorithm uses mobility based metric as cluster formation basic and calculation of weights of the nodes in the network. In order to model the mobility, the authors in this paper use the ratio of two consecutive signal



MCP

Avg. Cluster changes (RA)

Avg. Reelections (RE)

Fig. 3 Performance results of highest connectivity algorithm (HC)

strengths received by a node to know its relative motion with respect to its neighbors. Thus the relative mobility metric denoted as

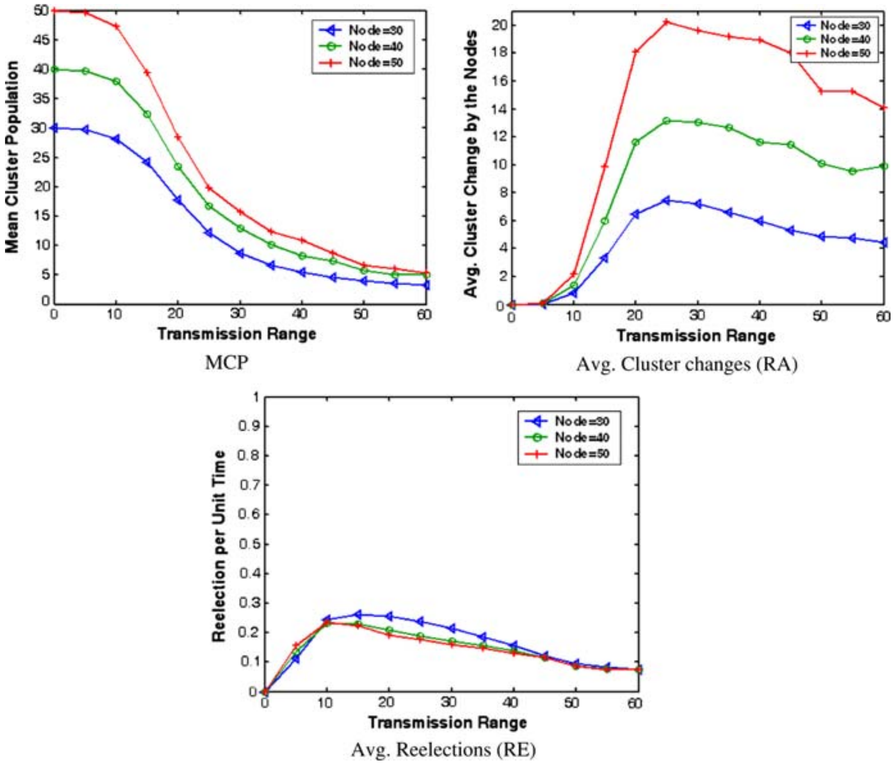
$$M_Y^{\text{rel}}(X) = 10\log_{10}\frac{R_xP_r^{\text{new}}X \rightarrow Y}{R_xP_r^{\text{old}}X \rightarrow Y}$$

at a node Y with respect to X gives either a positive or negative value depending on the value of the numerator. Here R_XP_Y is the received signal power received from node X. When $\left(R_xP_r^{\text{new}}X \rightarrow Y\right) > \left(R_xP_r^{\text{old}}X \rightarrow Y\right)$ the result gives a positive value indicating that both the nodes are approaching each other. Similarly, when $\left(R_xP_r^{\text{new}}X \rightarrow Y\right) < \left(R_xP_r^{\text{old}}X \rightarrow Y\right)$ the logarithm of the ratio gives a negative value indicating that the nodes are moving away from each other. Thus a node having N number of neighbors will have N such values of M_Y^{rel}. Thus the aggregate local mobility M_{Y} of a node Y is calculated by taking the variance of the entire set of relative mobility values. That is $M_Y = \text{var}_0\left(M_Y^{\text{rel}}(X_1),\ M_Y^{\text{rel}}(X_2),\ \ldots,\ M_Y^{\text{rel}}(X_1)\right)$ where the variance is taken with respect to 0. The motivation behind calculating the variance of relative mobility metric with respect to each neighbor is that a low value of M_Y indicates Y to be less mobile with respect to its neighbors. Thus choosing a relatively low mobile node to act as a cluster head yields a better cluster stability.

However, this logic is most applicable in a group mobility model where the nodes move in a group and is easy to find the relative mobility of a node with respect to its neighbors. But the situation where nodes move independently, it is difficult to find the relative mobility metric for every node. The relative mobility needs two consecutive “hello” messages to be transmitted. This further degrades the MAC efficiency by increasing the message exchanges.

Once the relative mobility metric for every node is decided, MOBIC is called upon the nodes. MOBIC works almost same as the Lowest ID algorithm, where the node IDs are replaced by the relative mobility metrics of each node. A node with the lowest value of M_Y amongst its neighbors becomes the cluster head or it becomes a cluster member. When two cluster heads accidentally come within their transmission range, re-clustering is deferred for Cluster_Contention_Interval (*CCI*) period as per the LCC [5] algorithm. If they remain within the range even after the CCI period, then re-clustering is invoked and the node with higher mobility metric resigns from its present status. In MOBIC the need of collecting the relative speed information from the neighbors degrades its performance, as continuous movement of the nodes in MANET may provide inaccurate mobility information during cluster setup.

The result of the MOBIC algorithm is depicted in Fig. 4. The result for population of clusters *MCP* is almost same as that of the LID algorithm except for



MCP

Avg. Cluster changes (RA)

Avg. Reelections (RE)

Fig. 4 Performance results of mobility metric algorithm (MOBIC)

the rate of decay in values which is slower in comparison to that of the former one. This is because the two neighboring cluster heads are allowed to retain their role for a Cluster Contention Interval (*CCI*) period. The number of reaffiliations *RA* by the nodes to the cluster heads is quite high like HC algorithm. And it converges to that of LID only when the transmission range is very high [31]. Being a LCC variant algorithm and as the reelection is deferred till the expiry of a *CCI* time period the reelection rate RE is reduced to a great extent.

3.4 Distributed Mobility Adaptive Algorithm (DCA, DMAC)

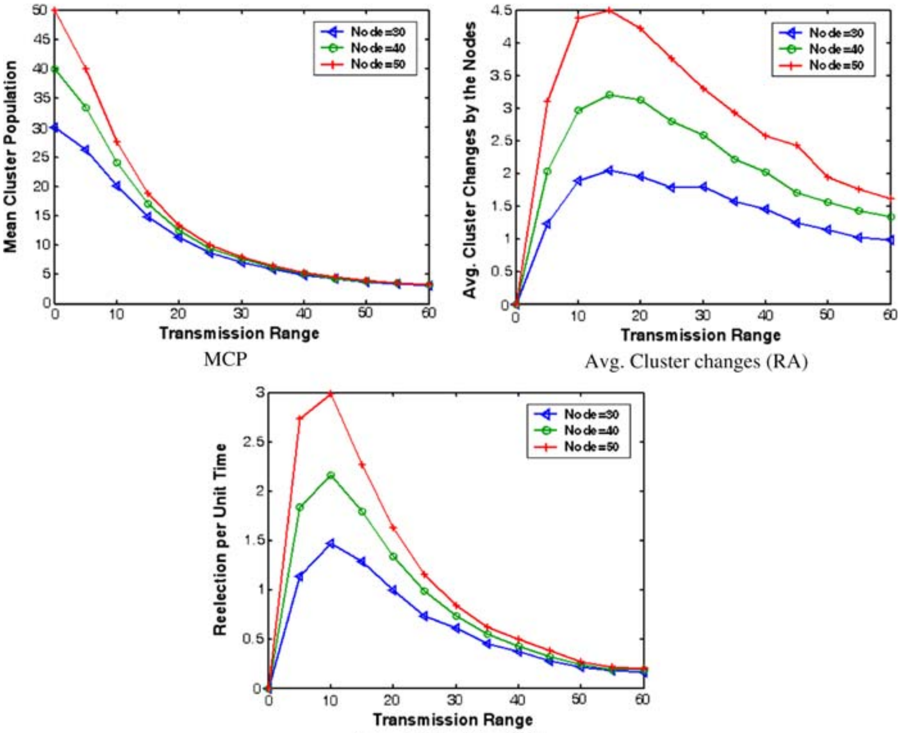
Basagni et al. presented an extension of [14] in [32] as a distributed clustering algorithm (DCA) that is mobility adaptive and truly distributed in nature. This algorithm is a generic weight based cluster formation algorithm. Here each node is associated with a unique parameter called the weight (i.e. a real number $\geq$0) that decides the role of a node. The weights may be the function of node transmission range or node mobility. DCA does not allow the change in network topology during the execution of the algorithm. A node having bigger weight among all its one-hop neighbors is selected as the cluster head (ties are broken by using LID). Any ordinary node opts to join a cluster head with the biggest weight when it comes across several other heads in its proximity. This algorithm explains well for the cluster formation where as the maintenance of the clusters in the presence of node mobility is not specified clearly for which DCA is mostly applicable for a static or a quasi static network.

In the distributed and mobility adaptive clustering algorithm DMAC [33] proposed by the same authors, the cluster formation process is almost same as that of DCA. However, the non-mobility of nodes during the execution of the algorithm is eliminated here, making it truly mobility adaptive. DMAC claims to be the most suitable algorithm for the cluster formation and maintenance in the presence of node mobility. It starts with the assumption that every node knows the ID, weight and role of itself as well as its one-hop neighbors. This proves that the cluster head is selected only with the knowledge of its local topology.

The major weakness of this algorithm lies with the lower weighted nodes. A lower weighted node decides its role only after all the 1-hop neighbor nodes with higher weight have decided their role. The worst case scenario occurs when the network topology contains a chain of nodes whose weights are in sorted order. Vilzmann [34] explains the chain activity in DMAC when a higher weighted node is added to the network or a link failure occurs between a cluster head and one of its member nodes.

The simulation result for DMAC is shown in Fig. 5. The average cluster population *MCP* reduces at a faster rate in comparison to that of MOBIC for a small transmission range. But the value is almost flat when the range is higher than 30. The following two criteria of DMAC increase the reaffiliation and reelections per unit time:

- No two cluster heads can be neighbors of each other.
- When a member node comes within the transmission proximity of another head node whose weight is more than its current head, then it joins the new head resulting in further re-affiliation.



MCP

Avg. Cluster changes (RA)

Avg. Reelections (RE)

Fig. 5 Performance results of distributed mobility adaptive algorithm (DMAC)

3.5 Weighted Clustering Algorithm (WCA)

Das, Mukherjee and Turgut have proposed a weighted clustering algorithm WCA [18, 35, 36] where the reelection takes place with the occurrence of certain event i.e., when there is a demand for it. Node parameters like degree of connectivity, mobility, transmission power and available battery power are considered for selection of a cluster head and are given different weights depending on the network scenario. For example, sensor networks where energy is a major constraint, battery power can be given higher weight. The combined weight of every node is calculated as$W_c = w_1 \times D_v + w_2 \times M_v + w_3 \times T_v + w_4 \times P_v$, where D_v is the degree difference of the cardinality of the neighbors of a node v represented as $|N_v|$ and a scenario based threshold δ which limits the upper bound for the number of members in a cluster, i.e. $D_v = ||N_v| - \delta|$. M_v is the average speed of a node since the last reelection taken place and $T_v = \sum dist(v, \{v'\})$, where $\{v'\}$is the set of neighbor nodes of v. P_v is the cumulative time for which the node v remains as the cluster head. All these parameters are normalized to a predetermined value and the weighing factors are chosen so that

$$\sum_{k=1}^{4} w_k = 1.$$

Initially, the node having the smallest weight is selected as the initial cluster head and its 1 hop neighbors become the members of the cluster. These covered set of nodes are exempted from taking part in the further selection [17]. This process is repeated till all the nodes are allocated a status of either a head or an ordinary member. Fair distribution of load among the cluster heads are made by restricting the upper limit for number of member nodes within a cluster. Such restriction of member nodes also improves the MAC layer efficiency.

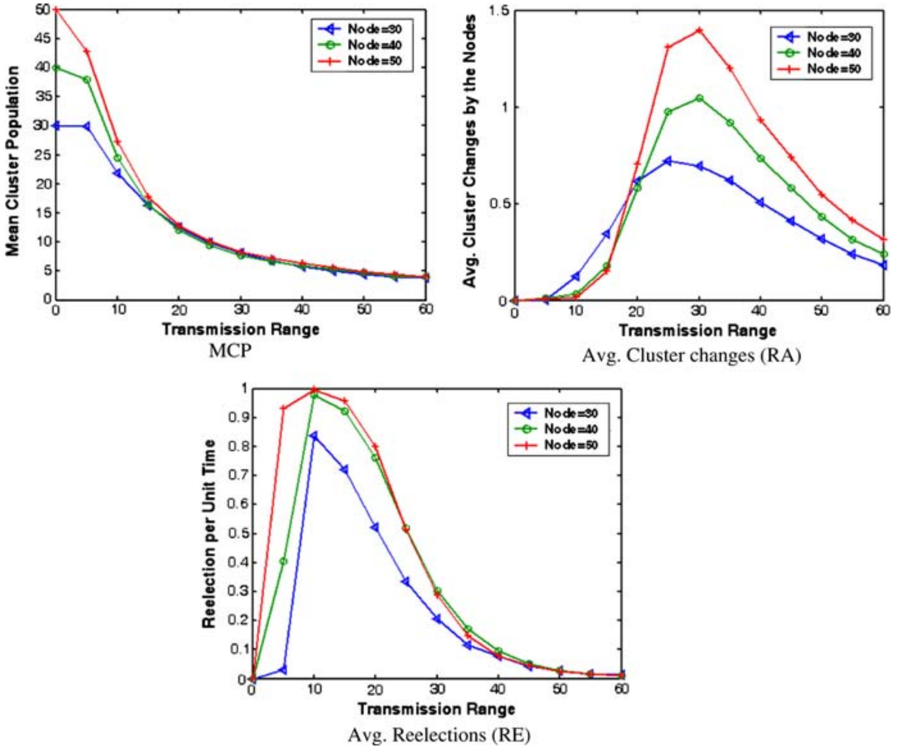
The limitation of the algorithm lies in yielding the global minima of weight values in the network. To have a distributed solution of the algorithm, a large number of information are stored and exchanged among the nodes to find the smallest weight. This becomes worse with the increase in network size. Further, the freezing time of mobility of nodes is high for the cluster setup. This is due to the need for computing so much of information for every node to calculate the combined weight. Whenever a reelection takes place the combined weight of every node needs to be calculated, resulting in further increase of the computation cost. The major drawback of this algorithm lies with not retaining the property of the lowest weight value to be the cluster head. This happens as WCA does not re-cluster when a member node changes its attaching cluster head [30]. That is, there may be a situation when a low weight node may enter into a cluster whose head is of higher value than this newly entered node.

Figure 6 depicts the simulation result for WCA algorithm. Here, the variation in node population does not have any effect on the cluster population *MCP* for higher transmission ranges. Moreover, being the on-demand clustering algorithm the reelection takes place whenever a member node goes out of the transmission range of all the current cluster heads or whenever a current head goes out of the range of its member nodes leaving them as orphan nodes. So the number of reelections *RE* is reduced to great extent. The cluster heads in this algorithm remain as neighbors to enhance the backbone connectivity which in turn reduces the number of reelections and re-affiliations. It can be observed that WCA outperforms any of the algorithms in terms of the re-affiliation *RA* and reelection rate *RE* of the mobile nodes in the dynamic network. WCA uses an upper threshold for the number of members in a cluster. This increases the re-affiliation rate at lower transmission range.

3.6 Generalized Distributed Mobility Adaptive Algorithm

A generalized version of DMAC algorithm is proposed by Ghosh et al. in [37] known as a Generalized Distributed Mobility Adaptive Clustering (GDMAC) algorithm. The algorithm improves the performance of DMAC by eliminating the restrictions imposed on the nodes and the following changes are incorporated into it:

- The maximum number of heads that are allowed to be neighbors at any time is *K* for the network.



MCP

Avg. Cluster changes (RA)

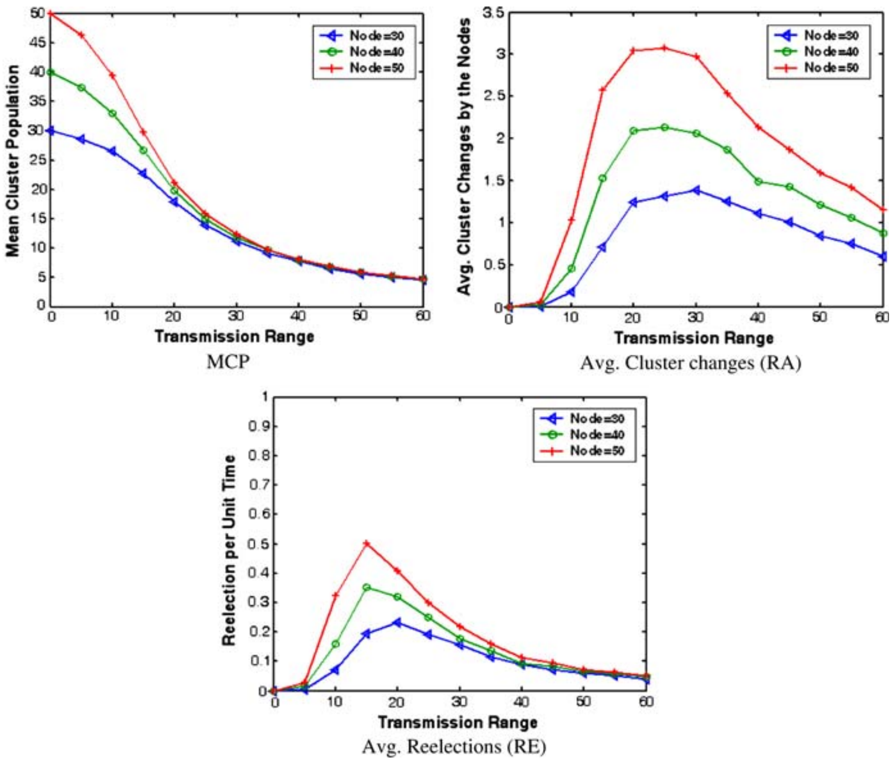
Avg. Reelections (RE)

Fig. 6 Performance Results of Weighted Clustering Algorithm (WCA)

- A member node within a cluster reaffiliates to a newly arrived cluster head only when the weight of the new head exceeds by a threshold amount of *H* over the weight of its current head.

It is clear that the first condition reduces number of reelections that was present in DMAC while the lower weight head was forced to resign from its role when a higher weight node comes to its neighbor. The value of *K* needs to be optimized as a lower *K* definitely keeps the number of cluster heads to a restricted size, but in turn it may not be able to outperform over DMAC for the number of reelections. Similarly, a higher *K* may reduce the rate of reelections to a significant value, but at the outset it increases the cardinality of the dominant set which may further decrease the routing efficiency. Thus, *K* can be defined as the *cluster density control parameter* for the whole network.

The second point introduces the threshold *H* which restricts the number of reaffiliations to take place when a member node comes in contact with multiple cluster heads. The higher the value of *H*, the lower is the chance that a node will switch to a new cluster. Thus, *H* can be defined as the *member density control parameter* for a single cluster. By nullifying both the values of *H* and *K*, GDMAC algorithm converges to the original DMAC algorithm.



MCP

Avg. Cluster changes (RA)

Avg. Reelections (RE)

Fig. 7 Performance Results of GDMAC for K = 3 and H = 35

Figure 7 depicts the result of the GDMAC algorithm. Though the cluster population *MCP* and average cluster changes *RA* by the nodes have no improvement over that of DMAC, but the reelection rate *RE* has a remarkable improvement over that of DMAC. As explained earlier, GDMAC operation basically depends on two factors such as K and H. It is worth noting that the higher value of K could increase the cluster population and decreases the number of re-affiliations *RA* and reelections RE. A comparison between DMAC and GDMAC for the cluster population is explained in [37]. But the value of H could change only the number of cluster changes by the nodes in the network. A higher H definitely causes lower re-affiliations as the nodes are allowed to stay connected with the existing head when the weight difference between the current head and the other neighbor head is less than that of the threshold value.

3.7 Weight Based Clustering Algorithm (WBCA)

Another weight based clustering algorithm WBCA has been proposed by Yang and Zhang [19]. This algorithm is a modification over the WCA algorithm that takes the mean connectivity degree and battery power into consideration for calculating the weight of nodes. The mean connectivity degree of a node is calculated as

$$C_v = \frac{\sum_{i=1}^{N_v} N_{vi} + N_v}{N_v + 1}$$

where N_{vi} is the degree of connectivity of ith neighbor of node v, and N_v is the degree of connectivity of node v. The consumed energy of a node is calculated as

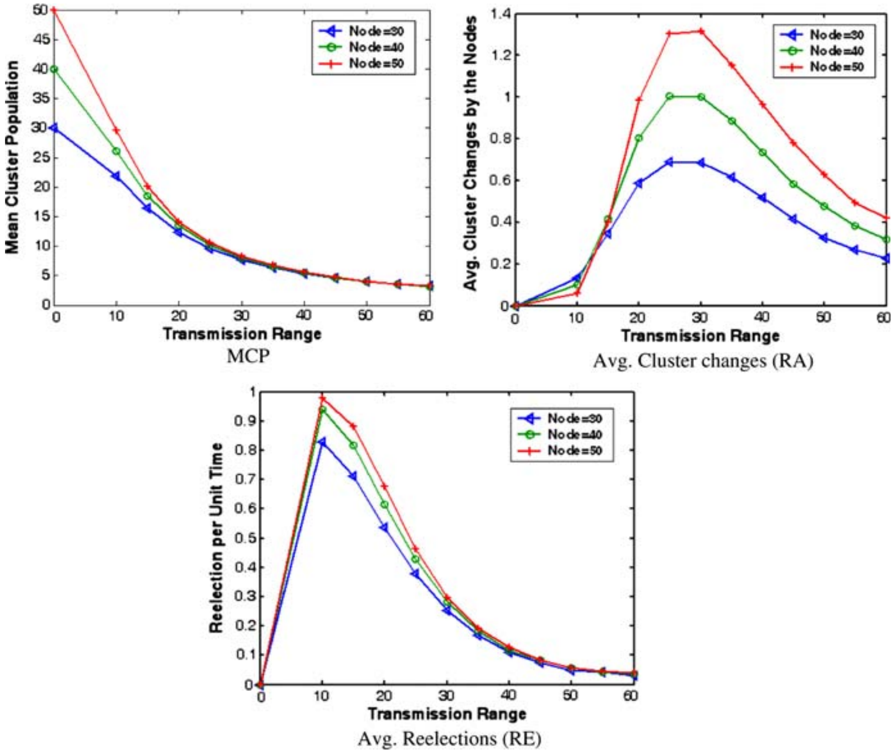
$$E_v = \sum_{i=1}^{q} N_{vi} \times e$$

where q is the time of period during which a node v acts as cluster head at ith times. Finally the combined weight is calculated as $W_v = w_1 D_v + w_2 E_v$, where D_v is the degree difference and is defined as $|N_v - C_v|$ for every node v. The values of w_1 and w_2 are the weighing factors that depend on the system requirements and $w_1 + w_1 = 1$. Unlike Lowest ID (LID) and Highest Connectivity (HC) algorithms, WBCA gives a uniform distribution of time for which the nodes act as cluster head. This also reduces the computation cost of cluster setup as it calculates only two values D_v and E_v for calculating the combined weight. However, calculating the mean connectivity degree of a single node needs to know the degree of connectivity of all its neighbor nodes. This is typically an unpredictable situation in a dynamic network as the mobility of nodes frequently changes its degree of connectivity. Thus like WCA, this algorithm also needs a considerable amount of freezing time for the nodes before the actual cluster setup.

The results of WBCA as depicted in Fig. 8 are almost similar to that of WCA. WBCA is the weighted variance of the WCA algorithm. That is, the weights of the nodes depend on different parameters in comparison to that of WCA. As a result, there occurs a slight change in the simulation result for *RE* and *RA* of WBCA algorithm.

4 Comparison of the Algorithms

Each of the algorithms has its own strengths and weaknesses. Depending on the network condition and requirement, any of the algorithms can be chosen for implementation prior to the actual routing job. For example LID results in an overall good performance among other algorithms in achieving the cluster population, re-affiliation overhead and number of reelections as shown in Fig. 9. But it biases the lower ID nodes to drain the resources fast or even node failure to occur fast. Similarly HC succeeds in minimizing the number of clusters (i.e. MCP) so that the routing delay is minimized due to reduced number of routing heads. But at the same time the re-affiliation overhead and the number of reelections are compromised which does not encourage a designer to choose this algorithm for implementation. MOBIC and GDMAC have a higher cluster population as in figure throughout because they allow more than one cluster heads to exist as neighbors for a Cluster Contention Interval (CCI) period or till their weight difference exceeds a threshold value. Higher cluster population of these two algorithms may increase the routing delay and affect the QoS of the network layer. But, when the stability of clusters is



MCP

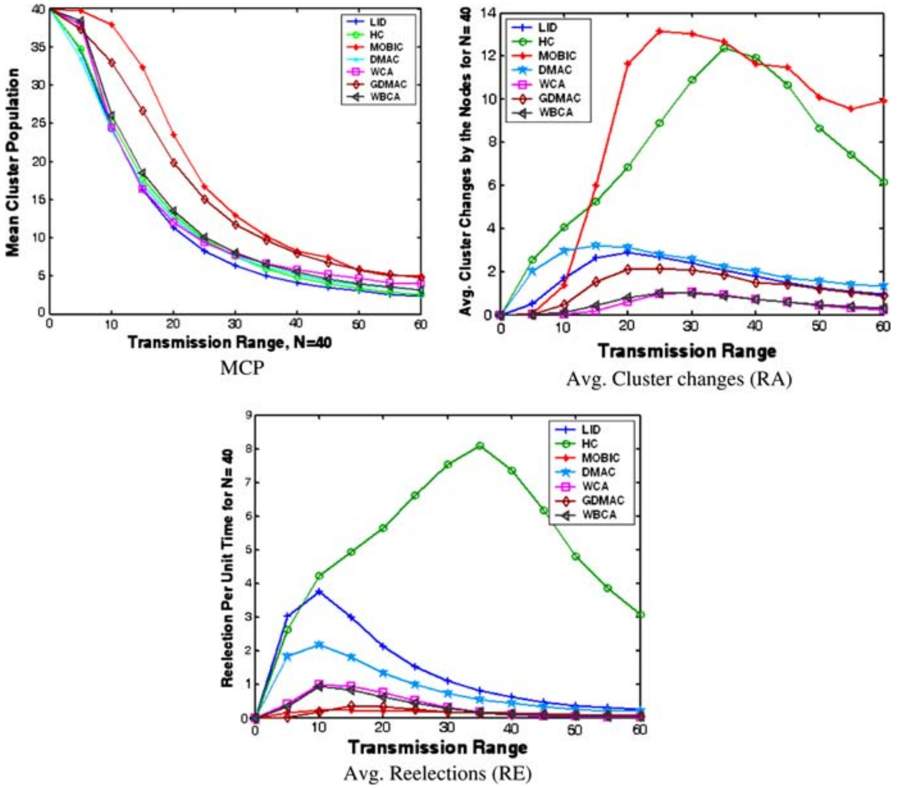
Avg. Cluster changes (RA)

Avg. Reelections (RE)

Fig. 8 Performance Results of Weight Based Clustering Algorithm (WBCA)

concerned, MOBIC and GDMAC provide the best result due to their lower reelection rate. In short they provide better cluster or route stability at the cost of increasing routing hops or the delay in packet forwarding. But as in figure MOBIC has a very high re-affiliation rate than GDMAC.

WCA and WBCA provide better overall performance when all the performance metrics are considered. But they need multiple parameters to be considered for its weight calculation. Thus initial cluster setup is delayed due to the weight calculation and become the designer’s choice to decide upon. Further, finding the lowest weight node among all the nodes in the network is not truly distributed in nature. However, the on demand reelection enhances the cluster stability by reducing the number of reelections RE as well as re-affiliations overhead. In a nutshell, WCA and WBCA can be better option if cluster setup delay is compromised. Finally, DMAC provides an average performance for all the parameters with a medium re-affiliation overhead and reelection frequency. Its major strength is its mobility adaptive ness during cluster setup. That is node mobility is not freezed during cluster setup. Also, clustering is done with the knowledge of the local topology of the nodes in a distributive fashion. So in a mobile ad hoc network where topology change can not be avoided even during the cluster setup, this algorithm proves to be most applicable than any other existing one-hop clustering algorithms. Table 1 summarizes the



MCP

Avg. Cluster changes (RA)

Avg. Reelections (RE)

Fig. 9 Comparisons of the simulation results for the algorithms

strengths and weaknesses of the algorithms in terms of their basics of cluster formation.

5 Conclusion

In this paper, we presented a detail simulation survey of the clustering algorithms considering single hop representatives of the energy constrained ad hoc network. The article starts with the clustering definition and its benefits to MANET. Linked Cluster Architecture is discussed in detail with greater emphasis on the cluster formation principles and the cluster maintenance of the algorithms. Simulation results are discussed to describe the effect of transmission range and the size of the network on the parameters like cluster density, frequency of reelection, frequency of cluster changes and granularity of cluster heads in the dynamic network. An overall comparison is presented in a table highlighting their characteristics, strengths and weaknesses. Partitioning the mobile nodes is a NP-hard problem. So researchers have aimed to obtain either a minimum dominant set (i.e. HC algorithm) to reduce the packet delivery delay or maximum cluster stability (i.e. WCA and WBCA) so that quality of service of network routing is maintained. While it is not clear that any

Table 1 Comparisons of the one-hop clustering algorithms

Names	Weight calculation/cluster head selection characteristics	Non-mobility of nodes during cluster setup	Strengths	Weaknesses
LID	Lowest Identity based node among the neighbors	Yes	Simple; Faster cluster setup	Biasness of low ID nodes to become heads
HC	Highest Connectivity based node among the neighbors	Yes	Simple; Minimum cluster heads at high transmission range	Highly unstable; A single change in connectivity may lead to reelection; Avg. cluster changes by the nodes are very high
MOBIC	Lowest aggregate mobility of a node w.r.t. its neighbors. Received signal strength is the key parameter to measure the relative mobility	Yes	Highly stable as two heads are allowed to be neighbors for Cluster Contention Interval (CCI) period	Requires large cluster setup time; Very high reaffiliation rate increases computation and communication overhead; May increase routing delay due to more cluster heads
DMAC	Lowest weighted node within its local topology. Weight may be the node mobility/transmission power	No	Distributive and mobility adaptive; Does not freeze node mobility during setup	Does not allow two heads to be neighbors increasing reaffiliation and reelection rate to some extend
WCA	Lowest weighted node in the entire network. Weight is the function of degree of connectivity, mobility, available battery power and utilized transmission power of individual nodes	Yes	Reelection is done on-demand, So better cluster stability; Reaffiliation is very low reducing the overhead	Many parameters are considered for setup, this increases setup delay; Selecting the global minima in a distributed fashion needs more message exchanges between the nodes and channel bandwidth consumption is increased

Table 1 continued

Names	Weight calculation/cluster head selection characteristics	Non-mobility of nodes during cluster setup	Strengths	Weaknesses
GDMAC	Enhancement over DMAC with the introduction of threshold parameter H and K	No	Highly stable as K heads are allowed to be neighbors; Unlike DMAC a node raffiliates to another head only when the weight difference of heads is more than H	Routing delay may be increased as the number of cluster heads or the routing points are more.
WBCA	Lowest weighted node in the entire network. Weight is the function of mean connectivity degree of the node and its consumed battery power	Yes	Less no. of parameters is considered than WCA; Reelection is done on-demand; So better cluster stability; Reaffiliation is very low reducing the overhead	Computation cost is more as calculation of mean connectivity degree of a node needs the degree of connectivity of neighbor nodes. Thus cluster setup is delayed; Getting the global minima in a distributed fashion needs more message exchanges between the nodes and channel bandwidth consumption is increased

particular algorithm is best suited for all scenarios, but each algorithm has definite advantages and disadvantages and is best suited for a certain situation. For example, when mobility of nodes can not be avoided during cluster setup, DMAC may be a better choice with a slight compromise in cluster maintenance.

Though one-hop clustering provides a shorter path for packet transmission than the multi-hop one, but its maintenance overhead is more in comparison with the other. So network designers should target to minimize the maintenance overhead while choosing any one-hop clustering algorithm for implementation. It is observed that battery power constraint which is a major challenge in ad hoc network has not been taken into consideration effectively in any of the above mentioned algorithms. The recent advent in wireless devices is much ahead of its energy issue counterpart. Hence, some energy based parameters may be considered as the key factors for cluster formation, without compromising with the cluster-setup delay or cluster-stability in the MANET.

References

1. Basagni, S., Conti, M., Giordano, S., Stojmenovic, I.: Mobile ad hoc networking. Wiley, New York (2004)
2. Perkins, C.E.: ad hoc Networking. Addison-Wesley, Verlag (2000)
3. Ulema, M., Nogueira, J.M., Kozbe, B.: Management of wireless ad hoc networks and wireless sensor networks. J. Netw. Syst. Manag. **14**(3), 327–333 (2006)
4. Royer, E.M., Toh, C.-K.: A review of current routing protocols for ad hoc mobile wireless networks. IEEE Pers. Commun. **6**(2), 46–55 (1999)
5. Chiang,C.C., Wu, H.K., Liu, W., Gerla, M.: Routing in clustered multi-hop, mobile wireless networks with fading channel. In: Proceedings of IEEE Singapore International Conf. on Networks SICON' 97, pp. 197–212 (1997)
6. Chlamtac, I., Conti, M., Liu, J.-N.: Mobile ad hoc networking: imperatives and challenges. Ad Hoc Netw. **1**, 13–64 (2003)
7. Chlamtac, I., Farago, A.: A new approach to the design and analysis of peer-to-peer mobile networks. Wirel. Netw. **5**, 149–156 (1999)
8. Yu, J.Y., Chong, P.H.J.: A survey of clustering schemes for mobile ad hoc networks. IEEE Commun. Surv. Tutor. **7**(1), 32–47 (2005)
9. Erciyes, K., Dagdeviren, O., Cokuslu, D., Ozsoyeller, D.: Graph theoretic clustering algorithms in mobile ad hoc networks and wireless sensor networks-survey. Appl. Comput. Math. **6**(2), 162–180 (2007)
10. Ohta, T., Inoue, S., Kakuda, Y.: An adaptive multi-hop clustering scheme for highly mobile ad hoc networks. In: Proceedings of Sixth International Symposium on Autonomous Decentralized Systems (ISADS'03), Italy, April (2003)
11. Lin, C.R., Gerla, M.: Adaptive clustering for mobile wireless networks. IEEE J. Sel. Area. Commun. **15**(7), 1265–1275 (1997)
12. Lin, C. R., Gerla, M. A distributed control scheme in multi-hop packet radio networks for voice/data traffic support. In: Proceedings of IEEE GLOBECOM, pp. 1238–1242 (1995)
13. Bandyopadhyay, S., Coyle, E.J.: An energy efficient hierarchical clustering algorithm for wireless sensor networks. In: Proceedings of IEEE INFOCOM '03, April (2003)
14. Basagni, S., Chlamtac, I., Farago, A.: A generalized clustering algorithm for peer-to-peer networks. In: Proceedings of Workshop on Algorithmic aspect of communication (satellite workshop of ICALP), July (1997)
15. Chang, J. H., Tassiulas, L.: Energy conserving routing in wireless ad hoc networks. In: Proceedings of INFOCOM 2000, Tel-Aviv, Israel. March, pp. 22–31 (2000)
16. Lochin, E., Fladenmuller, A. Moulin, J.-Y., Fdida, S.: Energy consumption models for ad-hoc mobile terminals. Mobile Ad Hoc Network, Paris Cedex 05, France (2005)

17. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. Freeman, San Francisco, CA (1979)
18. Das, S., Chatterjee, M., Turgut, D.: WCA: a weighted clustering algorithm for mobile ad hoc networks. J. Cluster Comput. **5**(2), 193–204 (2002)
19. Yang, W.D., Zhang, G. Z.: A weight- based clustering algorithm for mobile ad hoc network. In: Proceedings of 3rd International Conference on Wireless and Mobile Communications (ICWMC '07), Guadeloupe, March, p. 3 (2007)
20. Bollbas, B.: Random Graphs. Academic Press, Oxford (1985)
21. Baker, D.J., Ephremides, A.: The Architectural Organisation of a Mobile Radio Network Via a Distributed Algorithm. IEEE Trans. Commun. **29**(11), 1694–1701 (1981)
22. Tobagi, F.A., Kleinrock, L.: Packet switching in radio channels: part II—the hidden terminal problem in carrier sense multiple access and the busy tone solution. IEEE Transact. Commun. **23**, 1417–1433 (1975)
23. Baker, D.J., Ephremides, A., Flynn, J.A.: The design and simulation of a mobile radio network with distributed control. IEEE J. Sel. Area. Commun. **2**(1), 226–237 (1984)
24. Lin, G., Noubir, G., Rajaraman, R.: Mobility models for ad hoc network simulation. In: Proceedings of IEEE INFOCOM'04, Hong Kong, March (2004)
25. Zonoozi, M., Dassanayake, P.: User mobility modeling and characterization of mobility pattern. IEEE J. Sel. Area. Commun. **15**(7), 1239–1252 (1997)
26. Camp, T., Boleng, J., Davies, V.: A survey of mobility models for ad hoc network research. Wirel. Commun. Mobile Comput. **2**(5), 483–502 (2002)
27. Siddiqui, A.A., Prakash, R.: Effect of availability factor threshold and clustering gap on performance of clustering mechanisms for multi-cluster mobile ad hoc networks. In: Proceedings of IEEE Conference on Communications (ICC'02), vol. 5, pp. 3296–3300. April (2002)
28. Ephremides, A., Wieselthier, J.E., Baker, D.J.: A design concept for reliable mobile radio networks with frequency hoping signaling. Proc. IEEE **75**(1), 56–73 (1987)
29. Gerla, M., Tsai, J.T.C.: Multicluster, mobile,multimedia radio network. Wirel. Netw. **1**, 255–265 (1995)
30. Parekh, A.K.: Selecting routers in ad hoc wireless network. In: Proceedings of the SBT/IEEE International Telecommunication Symposium, ITS, August (1994)
31. Basu, P., Khan, N., Little, T.D.C.: A mobility based metric for clustering in mobile ad hoc networks. In:Proceedings of IEEE ICDCS 2001 Workshop on Wireless Networks and Mobile Computing. Phoenix, AZ (2001)
32. Basagni, S.: Distributed clustering for ad hoc networks. In: Proceedings of International Symposium on Parallel Architectures, Algorithms and Networks, pp. 310–315. June (1999)
33. Basagni, S.: Distributed and mobility-adaptive clustering for multimedia support in multi-hop wireless networks. In: Proceedings of Vehicular Technology Conference, VTC, vol. 2, pp. 889–893. Fall (1999)
34. Vilzmann, R.: Analysis of different clustering techniques for mobile ad hoc networks. EE 381K.11—Wireless Communications, March (2002)
35. Das, S., Chatterjee, M., Turgut, D.: An on demand weighted clustering algorithm(WCA) for ad hoc networks. In: Proceedings of IEEE GLOBECOM 2000, San Francisco, pp. 1697–1701. November (2000)
36. Chatterjee, M., Das, S.K., Turgut, D.: A weight based distributed clustering algorithm for mobile ad hoc networks. In: Proceedings of 7th International Conference on High Performance Computing (HiPC 2000), Bangalore, India, pp. 511–521. December (2000)
37. Ghosh, R., Basagni, S.: Limiting the impact of mobility on ad hoc clustering. In: Proceedings of the 2nd ACM International Workshop PE-WASUN'05, Montreal, CA, pp. 197–204 (2005)

Author Biographies

Suchismita Chinara has received her ME degree in Computer Science Engineering from Sambalpur University, India in 2001. She is currently working towards the Ph.D. degree in the Department of Computer Science Engineering, National Institute of Technology, Rourkela. Her research interest includes wireless networks, mobile computing and simulation modeling. She is a member of the IEEE.

Santanu Kumar Rath has received his Ph.D. degree in Systems Engineering and Management from Indian Institute of Technology, Kharagpur, India in 1994. He is currently working as Professor in Computer Science at National Institute of Technology, Rourkela, India. His research interest includes software engineering, data modeling, data mining, wireless ad hoc networks and simulation modeling. He is a member of IEEE.