Research Article

Topology Control by Transmission Range Adjustment Protocol for Clustered Mobile Ad Hoc Networks

Suchismita Chinara and Santanu Kumar Rath

Department of Computer Science and Engineering, National Institute of Technology, Rourkela 769008, India

Correspondence should be addressed to Suchismita Chinara, suchi.nitrkl@gmail.com

The growth of interest and research on multihop wireless network is exponential in recent years. In mobile ad hoc networks (MANET), the nodes play the role of routers to forward the packets of neighbor nodes as there is no fixed infrastructure available to do so. Clustering is a proven solution that maps the architecture of cellular networks into ad hoc networks. Here, selected nodes form the virtual backbone of the network and take part in packet routing. This achieves faster packet delivery as limited nodes are responsible for the same even though the network is not strongly connected. In this paper, a distributed topology adaptive clustering algorithm is designed that requires local information by the nodes for the formation of clusters. The role of cluster head is fairly distributed among the nodes to obtain a longer network lifetime. The change of cluster heads and the mobility of nodes disturb the node connectivity resulting in communication instability. To overcome such situations, a topology control protocol is developed that adjusts the transmission range of concerned mobile nodes to achieve local connectivity among nodes within the clusters even after the hand-off by the heads takes place.

1. Introduction

The nodes of the mobile ad hoc networks (MANET) are equipped with wireless transceivers and move freely while remaining reachable to each other. With limited transmission range the nodes are capable of communicating with each other using intermediate relays or multi-hop wireless links. So the basic assumption for the mobile ad hoc network is that a node can play the role of a router for forwarding the packets of its neighbors [1]. The node movement in the dynamic environment causes frequent topology changes disturbing the path of packet transmission. Further, the scarcity of radio resources and bandwidth, limited battery power, and computing power of nodes pose challenge in MANET scalability and efficiency [2]. Such a scenario demands for the partitioning of the network into logical groups so that the above challenges could be met efficiently.

In the cellular architecture, the presence of a fixed infrastructure such as a base station allows the host to communicate directly with itself reducing the wireless part of communication to a single-hop problem. A number of

good solutions have been proposed till date for handling the mobility of hosts by this base station. So the concept of mapping the cellular architecture into a peer- to- peer network leads to the formation of clusters [3]. Every cluster has a cluster head that plays the role of base station while allocating resources to the one-hop cluster members and lies responsible for the intercluster communication [4]. The set of cluster heads form the virtual back bone of the ad hoc network for data communication. Hence, a smaller population of members in the virtual backbone guarantees for faster communication.

The process of clustering is never completed without a proper maintenance scheme. The objective of cluster maintenance is to preserve as much of the existing clustering structure as possible. The node movement in the network results in frequent link failure or link establishment between the nodes. It demands for updating the cluster members from time to time. Further, the changing topology and limited node lifetime eliminate the possibility of the existence of permanent cluster heads in the network. This requires new cluster heads to be elected from time to time to adapt

the changing scenario. Hence, a well-designed clustering algorithm needs to be followed by least cluster maintenance. That means that the rate of updating cluster members and the number of reelections should be as low as possible.

Cluster heads being the communication hotspots tend to drain the battery power rapidly in comparison to the cluster members [5] of the network. So to eliminate the possibility of early death of any head node and to increase the network life time, every node in the network must get fair chance of acting as the cluster head. As a result the change of cluster heads leads to several hand-offs taking place. So the network connectivity is disturbed and the goal of communication is not fulfilled. Further, the mobility of nodes also changes the network connectivity. Therefore, topology control protocol is required to be designed to keep the nodes connected for packet routing.

The rest of the paper is organized as follows. Section 2 describes the work done in the related area. Section 3 describes the design of the clustering algorithm along with the neighbor detection protocol for the selection of volunteer and nonvolunteer cluster heads. Section 4 discusses the topology control for a clustered network by the design of a transmission range adjustment protocol followed by Section 5 that discusses the simulation results of the clustering algorithm in terms of the cluster maintenance parameters and the network life time. Finally, Section 6 concludes the paper.

2. Related Work

Partition of the random dynamic network into logical clusters was initially proposed by Baker and Ephremides [6] that was followed by other clustering algorithms which are mostly greedy. Some algorithms emphasize on minimizing the number of cluster heads [2, 7] in the virtual back bone so that the end- to- end delay and congestion is reduced while some clustering algorithms aim to maximize the cluster stability by unaltering the head nodes [4, 6]. The simulation survey of existing one-hop clustering algorithms is provided by the authors of [8].

The lowest ID-algorithm proposed by Ephremides et al. [4] is an ID-based algorithm where a node having lowest ID among its neighbors is selected as the cluster head. It retains its utility as a benchmark for its simplicity and producing reasonably stable cluster control architecture [7]. However, the lower ID nodes being biased to become the heads all the time result in faster energy drainage and the network lifetime is reduced [9]. Similarly, the degree of connectivity-based algorithm proposed by Parekh [2] selects a node with the highest connectivity among its neighbors as the cluster head. This algorithm results in a smaller dominant set ensuring faster communication, but the node mobility causes frequent change in the connectivity. Hence, the cluster head selection algorithm is called repeatedly.

A mobility metric-based algorithm MOBIC proposed in [10] selects a node with-least relative mobility among the neighbors as the cluster head. It improves the cluster stability by allowing two cluster heads to remain within the transmission range of each other for a cluster_contention_interval (CCI) period [8]. However, the process of obtaining relative mobility metric for every node degrades its performance making it unsuitable for most situations. Further, a distributed mobility adaptive algorithm DMAC proposed by the authors of [11, 12] eliminates the assumption of nonmobility of the hosts during clustering setup and maintenance. When two cluster heads come within the transmission range of each other; the head with the lower weight is forced to resign from its current role and become the member node of the other head having higher weight. This nonneighborhood restriction of cluster heads and the constraint of affiliating a node to a higher weighted head (if it is found within its transmission range) reduces the clustering efficiency. This problem was overcome by the authors of [13] in the generalized distributed mobility adaptive clustering algorithm (GDMAC) where K numbers of cluster heads are allowed to remain as neighbors. Further, reaffiliation by a node occurs only when the difference in weight values of two neighbor cluster heads exceed a threshold value H.

The combined metric weighted clustering algorithms (WCA) proposed by the authors of [14–16] use multiple node parameters to select the cluster head. But obtaining so much of information to compute the combined weight for every node in the network itself needs a longer frozen period of motion before the cluster is actually formed. A large number of message exchanges are also required to yield the node with the lowest weight in the entire network.

All the above clustering algorithms assume fixed transmission ranges of the mobile nodes for the sake of simplicity. The connectivity of the nodes changes due to several reasons within the network and so as the network topology. To resume the connectivity of the nodes their transmission ranges could be adjusted efficiently with optimum energy consumption. There exist some topology control algorithms out of which some are centralized whereas others are distributed in nature. Topology control protocol proposed by the authors of [17] finds an enclosure graph for every node that contains its immediate neighbors strongly connected. Then after the Bell-man Ford shortest path algorithm is applied on the enclosure graph where the power consumption by the links is considered as the cost metric, the algorithm successfully finds the minimum energy path from the static nodes to a certain fixed node. However applying the same to a mobile ad hoc network is not possible due to the changing environment.

Local information no topology (LINT) algorithm proposed by the authors of [18] is a distributed algorithm that allows the nodes to adjust their transmission range according to the degree of connectivity. The algorithm deals with three parameters. They are the degree of connectivity of the node and the lower and upper threshold for its number of neighbors. If the degree of connectivity is more than the upper threshold d_h, then the range is reduced to de-register some of the connected nodes and vice versa so that the number of neighbors remains within a predefined range. This is quite unrealistic in a mobile ad hoc network where the node position and mobility is highly unpredictable. The modification to LINT was the local information link state topology (LILT) algorithm that uses the global topology

information to adjust the range of a node. Similarly, the authors of [19] proposes a self-adjusting transmission range control protocol that keeps predefined number of nodes as neighbors to protect the neighborhood relationship among the nodes. All these algorithms aim for a network wide topology control to keep the nodes connected for communication. We propose a protocol that aims to control the topology within a cluster so that the nodes within a cluster remain connected and the frequency of reclustering is reduced.

3. TACA: Topology Adaptive Clustering Algorithm

The mobile ad hoc network can be modeled as a graph $G = (V, L)$ where V is a finite set of mobile nodes and L is a finite set of links that exist between the nodes. In the dynamic network the cardinality of the nodes $|V|$ remains constant, but the cardinality of links $|L|$ changes due to the mobility of the nodes. Each node $v \in V$ is uniquely identified by an integer identifier along with a wireless transmission range v_{trange}. When a node v_1 is within the transmission range of v_2, (i.e., dist $(v_1, v_2) < v_{\text{trange}}$) they are assumed to be connected by a unidirectional link $l_{12} \in L$, such that whenever v_1 broadcasts a message, it is received by v_2 via l_{12}. Similarly, when v_2 is within the transmission range of v_1, they are assumed to be connected by an unidirectional link $l_{21} \in L$, such that whenever v_2 broadcasts a message, it is received by v_1 via l_{21}.

3.1. Basis of the Algorithm. A preliminary version of the work is provided in [20]. The current topology adaptive clustering algorithm TACA and the range adjustment protocol TRAP for topology control has the following features.

   (i) The nodes have the capability to increase/decrease their transmission range as per the requirement. However, the maximum possible transmission range that a node can posses is $\text{Range}_{\text{MAX}}$.

  (ii) Mobility of the nodes in the dynamic network is a prime factor so as its battery power. Both of them decide the cluster stability and the network life time. Hence, these two factors are chosen as the weight deciding factors for the nodes.

 (iii) A volunteer cluster head election procedure takes place in the network when the network is first activated.

 (iv) A nonvolunteer cluster head election takes place within a cluster when the present cluster head drains its battery power below a threshold value.

  (v) In order to use the node battery power efficiently, all the nodes get almost fair chances of serving as cluster heads, so that load on individual nodes could be avoided.

 (vi) The volunteer head election process is repeated when no more non-volunteer cluster head election is possible to take place. That is, none of the node in a cluster have required battery power to serve as cluster

head. Or a node moves out of range of all other nodes becoming an isolated node.

3.2. Neighbors Detection Protocol (NDP). Before the actual clustering takes place, it is required for every node to set up a bidirectional link between its immediate neighbors. The proposed neighbor detection protocol (NDP) provides mechanism to detect the one-hop neighbors of the nodes in the ad hoc network as and when required. In this protocol the one-hop neighbors are recognized by the exchange of a *neighbor detection packet* (NDPAK) between the nodes.

The format of the NDPAK is as:

SID	RID	STR/RTR	NRQ/NAC	WT	STATUS

SID refers to source identification and RID refers to receiver identification with *X denoting all and NUM denoting node with identifier NUM*. STR stands for source transmission range, RTR receiver transmission range, *NRQ the neighbor request packet, NAC neighbor acknowledge packet*, WT weight of the packet sender, and STATUS node status of the packet sender, *0 is uncovered, 1 is cluster member, and 2 is cluster head*.

The NRQ/NAC field distinguishes a neighbor request packet from a neighbor acknowledge packet. The weight WT of the node specifies its ability to be selected as a cluster head. The STATUS field of the node indicates its current role in the network. Initially it is set to 0 for all the nodes, as the roles of the nodes remain undecided before the cluster is actually formed. As the node becomes cluster member or cluster head, this field is filled with appropriate values.

The steps associated with the NDP are the following

Step 1. Node u broadcasts a neighbor detection packet (NDPAK) specifying it as a request message (NRQ).

Step 2. The packet may be received by any node v which is within the transmission range of u. As a result node v sends back the NDPAK to the former specifying it as a *neighbor acknowledgement packet* (NAC). This packet carries the detail information of node v piggybacked into it.

Step 3. After receiving NAC packet from v, node u updates its neighbor table (NTAB) by registering v as its neighbor along with its node information. The details of the NDTAB are discussed later in this section.

Step 4. Finally, u sends back a *neighbor confirmation* (NC) message so that v updates its own neighbor table and a bidirectional link is established between the two nodes.

Step 5. Step 2 through 4 repeated for all are nodes, that is within the range of u.

Figure 1 provides below a sample network topology with five nodes. Here, the nodes are identified by unique integer numbers from 1 to 5. It is clear from Figure 1 that node 3 lies in the transmission range of both nodes 1 and 2 whereas node 4 and 5 are exclusively in the range of 1 and
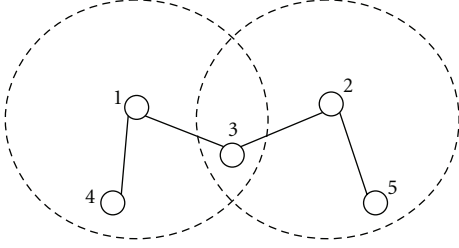
FIGURE 1: A sample network topology.

2 respectively. The NDPAK sent by node 1 has the contents as:

1	X	1_{trange}	NRQ	WT (1)	0

The receiver field of the packet has the value X to indicate that the packet is to be received by all the nodes that are in the transmission range. WT (1) value indicates the current weight of the node 1. This packet will be received by both node 3 and 4, and they will send back the NAC packet to acknowledge it. The data structure of the acknowledgement packet (NAC) sent back by node 3 to node 1 may be written as:

3	1	3_{trange}	NAC	WT (3)	0

For the topology of Figure 1, where node 3 receives the NRQ packet from both nodes 1 and 2, it sends back the NAC packets accordingly. The arrival of the NAC packets informs the sender about the presence of its one-hop neighbor and it updates its own neighbor table NTAB accordingly.

The detail structure of the *neighbor table* NTAB is as:

NID	DIST	NS	NTR	NWT

NID stands for *neighbor ID*, DIST stands for *neighbor distance*, NS stands for *neighbor status*, NTR stands for *neighbor transmission range*, and NWT stands for *neighbor weight*.

The neighbor distance DIST may be calculated from the signal strength of the received signal [21]. However, for the current work the Euclidian distance between two nodes is considered as the actual distance between the two nodes. When a NAC packet arrives, the source node updates its NTAB and sends a confirmation message to the later so that a bidirectional link is established between both of them.

3.3. Calculation of the Node Weight. The mobility of nodes may change the network topology very frequently. So choosing the less mobile nodes for the formation of the virtual back bone is always preferred. This ensures better backbone stability. Further the limited battery power devices consume their energy and become dead while routing the packets through them. This delinks the path for packet routing and demands for re-establishment of routing back bone. In order to ensure the availability of routers in the routing backbone, nodes with more available battery power are chosen as the back bone-forming nodes.

Keeping these factors in mind, the node weights are calculated by considering the node mobility and its available battery power as the key values. Here δ is assumed to be the maximum permissible speed of a node in the network. Thus the mobility factor of every node is calculated by computing the difference of δ and its average mobility during a certain time interval. A larger mobility factor indicates a less mobile node and vice versa. The available battery power is the energy associated with the node at the instant of weight calculation. These two parameters are added with different weighing factors to find the individual node weights. The steps for calculating the weights are discussed below.

Step 1. Let the total distance covered by a node v during last n discrete time units be $D_v = \sum_{i=t-n}^{i=t} \text{dist}_v$, where t is the current time.

Compute average mobility of a node $S_v = D_v/n$.

Step 2. Compute the Mobility factor $\Delta M = \delta - S_v$, That is, The difference of the average speed of the node from δ.

Step 3. Compute available battery power as

$$P_{\text{a}v} = P_{\text{a}v} - P_{\text{cons}}, \tag{1}$$

where $P_{\text{a}v}$: Available battery power of the node and $P_{\text{cons}} =$ battery power consumed by the node (discussed in Section 3.5).

Step 4. Compute the weight of the node as

$$\text{WT}(v) = x_1 \Delta M + x_2 P_{av}, \tag{2}$$

where x_1 and x_2 are the weight factors that are chosen carefully to provide due weight age to either factors depending on the network scenario. The values of the two weight factors are normalized so that

$$x_1 + x_2 = 1. \tag{3}$$

3.4. Selection of Volunteer Cluster Head. After the weight calculation of the nodes, the initial clustering algorithm is called upon to select the set of *volunteer cluster heads*. A pseudocode segment of the algorithm is presented in Algorithm 1.

The code indicates that a node having maximum weight among its 1-hop neighbors declares itself as the volunteer cluster head. And its 1-hop uncovered neighbors (i.e., whose role is not yet decided) become the members of the volunteer head. The set of covered nodes are exempted from taking part in subsequent selection procedure and this process is repeated till all the nodes are assigned with their role either as a cluster head or a cluster member. During the clustering phase, every node broadcasts its ID along with its weight WT_i to the neighbors and stores the weights WT_j that it hears from other nodes. If it does not hear another node ID with weight higher than itself then it declares itself as a volunteer cluster head and its one-hop uncovered neighbor nodes become its members. In case of a tie in the node weights the lower ID node is preferred for the role of cluster

```
Begin
    ......

For (every v ∈ V)
If STATUS (v) = 0 then
{
If WT_v ⟩ All WT_i where i ∈ Γ(v)
                            (//Γ(v) is neighbor set of v)
Then Set head = v
Set STATUS (v) = 2
End if

For (every x ∈ Γ(v))

If STATUS (x) = 0 Then

Set HEAD (x) = head
Set STATUS (x) = 1
End if

End for
}
End if
End for

    ......
        End
```

ALGORITHM 1: Pseudocode for volunteer cluster head selection.
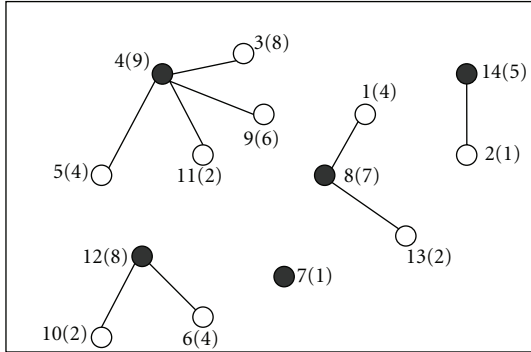

FIGURE 2: Volunteer cluster heads are elected to form clusters.

head. The example of a clustered topology is demonstrated with the help of Figure 2.

In Figure 2 the nodes are identified with unique integer numbers followed by another integer in parenthesis. In this case the number in the parenthesis denotes the assumed node weight. As shown here, the dark circled nodes with higher weight their neighbors are selected as the cluster heads. A head can have zero or more number of members within its cluster. Unlike [10] once a member node is affiliated with a cluster head, it does not reaffiliate to a new head unless it goes out of the range of its current head or the head drains out of battery power. This reduces the number of reaffiliations lowering the cluster maintenance overhead.

After the clusters are actually formed in the network, every node in the network maintains its own cluster table.

The format of the cluster table (CTAB) may be written as:

MID	MWT	MDIST	MT$_{trange}$

MID indicates member ID, MWT member weight, MDIST member distance, and MT$_{trange}$ member transmission range.

The cluster table is updated with the occurrence of node re-affiliation and reelection. That is, when a member of the current head goes out of its transmission range, it is deleted from the CTAB. Similarly a new node is added into the table when it becomes the member of this cluster.

3.5. Battery Power Consumption Model. Battery power consumption of the mobile devices depends on the operating mode of its wireless network interfaces. Considering a broadcast communication between the nodes of the dynamic network, wireless interfaces can be assumed to be in any of the following operating modes:

   (i) transmit: for transmitting packets,

   (ii) receive: for receiving packets,

   (iii) idle: a default mode when the node is ready to transmit or receive,

   (iv) sleep: the low power consumption state when a node cannot transmit or receive until woken up.

Usually in mobile ad hoc networks the sleep state of the nodes is rarely taken into consideration because of the lack of any central infrastructure. Assuming the cluster members to be senders or receivers at any instant of time, their power consumption could be modeled as

$$\text{Energy}_{\text{member}} = m_{\text{send/receive}} \times \text{size}_{\text{packet}} + c_{\text{broadcast}}, \quad (4)$$

where an incremental cost m and a fixed cost c for the broadcast communication are taken into consideration [22].

But the scenario is bit different for the cluster heads. In addition to its own packet transmission and reception, it has to route the packets for its cluster members along with allocating resources to them. As a whole, the energy consumption of a head node is considered to depend on the following parameters:

   (i) the traffic forwarded by the head,

   (ii) no. of members served by the head,

   (iii) total transmission power utilized by the head in serving the members.

Keeping this in mind, a battery power consumption model has been developed below for the cluster heads

$$\text{Energy}_{\text{head}} = \alpha^* |n_i| + \beta^* \text{Traffic}_{\text{bcast}} + \gamma^* \sum_{v' \in n_i} \text{dist}(v, v'), \quad (5)$$

where $|n_i|$ is the cardinality of a cluster, that is the number of members served by the cluster head. The more the number of members, the higher is the battery consumption. Traffic$_{\text{bcast}}$ is the cost of energy consumption in traffic forwarding. This

```
Begin
    ......
Let i = current_head//volunteer or non-volunteer
Compute max_wt = max {WT(j) where j ∈ member(i)}

Set next_head = v where WT(v) = max_wt

Head (i) = next_head
For (every j ∈ member(i) other than next_head)
{
If dist (next_head, j) ⟨= next_head_trange Then
{
Head (j) = next_head    //hand off

Else
Reaffiliate j to other head within range    //Reaffiliation

Else
Select j as volunteer head    //Reelection
}
End if
}
End for
    ......
        End
```

ALGORITHM 2: Pseudo code for non-volunteer cluster head selection.

FIGURE 3: Non-volunteer cluster head election.

indicates the volume of data being forwarded by the cluster head. Finally, $\sum_{v' \in n_i} \text{dist}(v, v')$ is the total transmission power utilized in communicating the member nodes of the cluster head. The radio range coverage by the head node has a considerable effect on its energy consumption. Depending on the RF environment the energy consumption can vary from p_v^2 to p_v^4 as suggested by the authors of [23], where p_v is the transmission power utilized by the head node in communicating a 1-hop neighbor v' within its cluster. However, as we consider the distance between the nodes in a cluster is very small, we setup a linear relation between the transmission power and the battery power consumption of the head node. α, β and γ are the weighing factors for the corresponding network parameters. These values are kept flexible so that they can be changed as per the network scenario. For example, when the network traffic is very high β can be given more weight than the other two. Similarly, in a dense network where the cardinality of clusters is more, the weight of α dominates the other factors. All three parameters are chosen in proportionate so that $\alpha + \beta + \gamma = 1$.

3.6. Selection of Nonvolunteer Cluster Head. The need for selection of non-volunteer head occurs when the current head (either the volunteer head or existing non-volunteer head) remains with battery power below a threshold value. During this phase, the current head invites one of its own cluster members having the highest weight to be the cluster head. But it is the choice of the selected node to accept the cluster head role or not depending on its available resources. After accepting the invitation of the current head it becomes the new cluster head for that cluster. The former
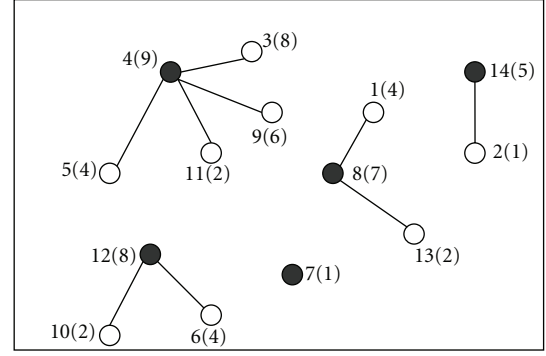
head now hands off its existing members to the new head before it affiliates itself as a member to the newly selected non-volunteer cluster head. Here, the selection process takes place locally within a cluster reducing the computation and communication overhead that would have been obtained in the global cluster head selection procedure. The pseudocode segment of the algorithm for finding the non-volunteer head may be written as Algorithm 2.

Sample of a non-volunteer cluster head selection is shown in Figure 3. It can be seen that node 4 selects node 3 as the new head. Nodes 11 and 9 lie in the transmission range of 3. So node 4 hands off these two nodes to the new head 3. But node 5 is out of the transmission range of 3. It also does not lie within the transmission range of any other cluster head. So it becomes an isolated node. This is clear from the code that selection of a non-volunteer cluster head results in hand-off reaffiliation or reelection (isolated head).

Formation of such new isolated cluster heads increases the members of the dominant set and hence the length of the routing back bone. This, in return, enhances the packet transmission delay. Further, due to the dynamic nature of the nodes, the connectivity of members to the head node may be disturbed and the situation is total chaos. To improve the situation a topology control algorithm could be implemented to retain the node connectivity.

4. Topology Control Protocol for the Clusters

Design of topology control protocol deals with the algorithms where the mobile radio nodes are allowed to adjust their transmission ranges so that the node connectivity is maintained with optimum energy consumption. As discussed in [24], a nonhomogeneous topology control allows the nodes to choose different ranges within a maximum limit depending on the information they obtain to compute the topology.

A distributed topology control approach is proposed in this paper to leave maximum possible nodes connected in a cluster zone with minimum possible energy consumption and transmission interference. The transmission range adjustment protocol TRAP is designed that allows a node to adjust its transmission range so that it could affiliate itself to any nearby cluster head in spite of forming an isolated node.

TRAP can be called upon to avoid formation of isolated cluster heads during

(i) the hand-off of members so that the unreachable members can be affiliated to new head,

(ii) the node movement when a member/head node goes beyond the range of head/member nodes, respectively.

The principle of working of TRAP could be explained as follow. Let node u be an isolated node. It increases its transmission range by ΔR. Repeat Steps 1 1 and 2 till transmission range of u is $\leq$ Range$_{MAX}$ and node u gets affiliated with an existing head.

Step 1. Node u broadcasts an NDPAK packet with NRQ request to the network indicating its new transmission range.

Step 2. The NDPAK packet may be received by an other node(s) that is (are) within the new transmission range of u. Below the different scenarios are explained:

Case 1. Let a single node v receives the NDPAK packet it is a cluster member node.

Action. Node v checks the increased transmission range of the sender packet. But it does not send back the NAC packet. Because v is already affiliated to a head node and there is no point in enhancing its own range at the cost of increased energy consumption.

Case 2. Let a single node v receive the NDPAK packet it is a cluster head.

Action.

(i) Node v checks for the possibility to increase its range. If it is ready with the resources to affiliate u as a member, then it increases its transmission range to $R+\Delta R$ and sends an NAC message to u.

(ii) Node u includes v as a neighbor in its NTAB and sends a neighbor confirmation NC packet to v.

(iii) Node v sets its transmission range to $R+\Delta R$.

(iv) Node u joins as a cluster member to v, and v updates its cluster table (CTAB) accordingly.

(v) Node v continues with the updated transmission range till it retires from the status of cluster head.

Case 3. Two nodes v_1 and v_2 receive the NDPAK packet and both are cluster heads.

Action.

(i) Both v_1 and v_2 increase their ranges to $R + \Delta R$ and send back the NAC packet to u.

(ii) Node u chooses the node with higher weight as neighbor and sends back the NC packet accordingly to the concerned head.

(iii) The head that receives the NC packet updates its transmission range to $R + \Delta R$ and the cluster table CTAB accordingly.

(iv) It continues with the updated transmission range till it retires from the status of cluster head.

(v) The head that does not receive the NC packet within the stipulated time period adjusts back its range to the original value R.

Case 4. None of the nodes receive the NDPAK packet with NRQ request.

Action. Node u waits for time out period to get NAC and further increases the range by ΔR. Thus the new range is now $R + 2\Delta R$. Go to Step 1 1 till the given conditions are satisfied.

Step 3. Declare node u as an isolated head.

5. Simulation Results and Discussion

The simulation of the topology adaptive clustering algorithm TACA is carried out with N numbers of nodes in a 100×100 grid area where the nodes are deployed randomly in the network. The Random Walk mobility pattern [25, 26] is implemented on the nodes which represents the most erratic and unpredictable movement where a node moves from its current location to a new location by choosing random speed between 0 and δ and direction between 0 and 2Π. When the node reaches at the simulation boundary, it bounces back into the simulation area. For the current simulation the maximum speed (δ) for the nodes is 5 m/sec.

The energy consumption of cluster members for different operating modes is considered [27, 28] as

$$\text{Broadcast send} = 1.9 \, \frac{\mu J}{\text{byte}} * \text{size}_{\text{packet}} + 250 \, \mu J,$$

$$\text{Broadcast receive} = 0.50 \, \frac{\mu J}{\text{byte}} * \text{size}_{\text{packet}} + 56 \, \mu J, \quad (6)$$

$$\text{Idle} = 808 \, J/\text{sec}.$$

The packet size is considered as 1024 bytes (or 1 KB) for the simulation. All the nodes in the network are equipped with the same initial battery power of 1 Joule. Before the topology control protocol is implemented on TACA, all the nodes are assumed to have equal transmission range. The maximum range Range$_{MAX}$ that the nodes can posse is 50. The following metrics denote the performance result of the proposed algorithm along with its result when the TRAP is applied onto it:

(i) average node re-affiliations

(ii) average updating rate of cluster heads

(iii) network life-time

(iv) average cluster density.

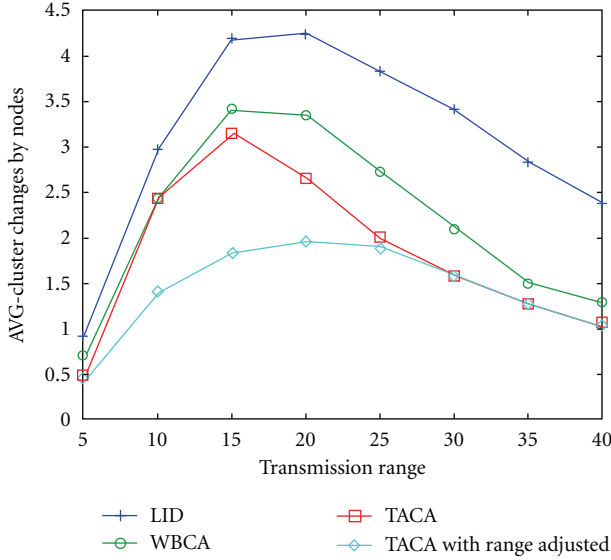*5.1. Average Node Re-Affiliations.* Re-affiliation by a member node occurs when it leaves its current head's transmission

FIGURE 4: Comparison of algorithms for reaffiliation, $N = 50$.

● Elected non-volunteer cluster head
● Isolated head formed

FIGURE 5: Comparison of algorithms for reelection, $N = 50$.

zone and enters into another cluster zone. In addition when a non-volunteer head is elected, hand-off takes place by the current head so that nodes of the existing head re-affiliate to the new head (if exists within the transmission range or finds another head within its range).

Figure 4 compares the result of the proposed TACA and its variable transmission range version with that of LID and WBCA algorithm. LID, being one of the simple and base algorithms for most other algorithms, is preferred for reference. Similarly, WBCA being a most recent clustering algorithm provides the similar result as the benchmark WCA algorithm as studied in [8]. It is clearly seen that TACA results in lower re-affiliations of the member nodes than the other algorithms. The lower re-affiliation reduces the communication and the computation complexity. This also reduces the frequent update of the cluster table CTAB.

When TRAP is implemented on TACA, the re-affiliation rate is further reduced as seen in the figure. For the low transmission range of the nodes, TRAP enables a node to enhance its range so that it remains affiliated to a cluster head. Thus the re-affiliation is minimized.

But, for high transmission range of the nodes, TRAP has very less impact on TACA which can be seen in the figure. The reason is that the value of $Range_{MAX}$ and standard transmission range of nodes for simulation are very close to each other. So the result of TRAP implemented on TACA coincides with that of TACA alone.

5.2. Average Updating Rate of Cluster Heads.
The election of cluster heads in TACA takes place for two occasions. First, when a single node becomes orphan or isolated by moving away from the transmission range of other nodes declaring itself as an isolated cluster head. Second, an existing cluster head consumes the threshold amount of its battery power and selects a member node with maximum weight as a head (i.e., non-volunteer cluster head).
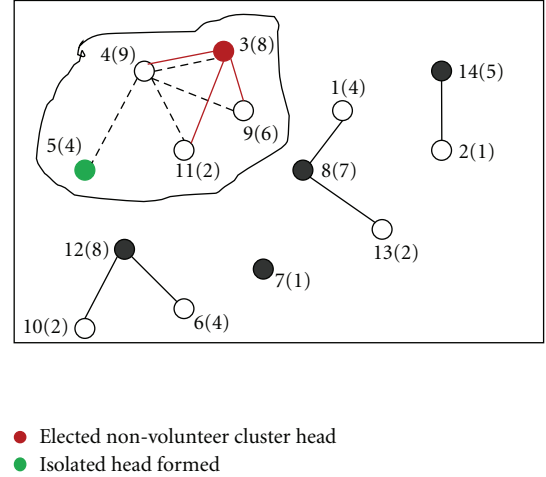

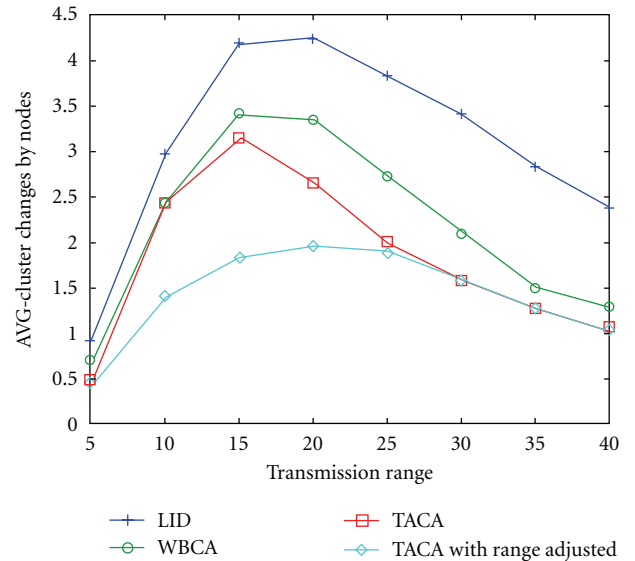FIGURE 6: Network life time (in ms) for various $Range_{MAX}$.

The average reelection rate of TACA is similar to that of WBCA as seen in Figure 5. However, by the implementation of TRAP, the improvement on TACA could be easily seen below. It is understood that for low transmission range, the isolated heads increase the range to $Range_{MAX}$ and get the opportunity to affiliate to other cluster heads. This results in lowering the election rate of the cluster heads. However, for higher transmission range the simulation result of TACA implemented with TRAP merges with that of former algorithms. The reason is that the ranges of the nodes are closer to the value of $Range_{MAX}$ chosen for simulation.

Figure 6 gives a result for the effect of transmission range over the battery power consumption. This helps in choosing the $Range_{MAX}$. It is seen that for lower values of $Range_{MAX}$, a greater network life time is obtained whereas the same goes on decreasing as $Range_{MAX}$ is increased. This is because the energy consumption of the node is proportional to its radio
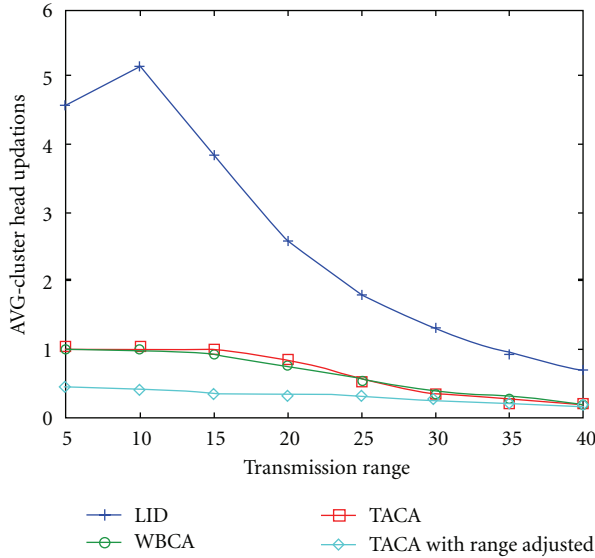
Figure 7: Comparison of network life time (in ms).

Figure 8: Comparison of algorithms for number of cluster heads.

range that results in faster battery drainage of the mobile nodes reducing its life span.

5.3. Network Lifetime. The network life time may be defined as the span of time period from the first activation of the network to the time when the first node becomes dead by consuming its battery power. The available battery power of a node decides its life time. In the simulation it is assumed that when a node consumes 95% of its maximum battery power it becomes dead. And the network fails when a single node becomes dead.

The result of network life time for LID algorithm and WBCA algorithm is compared with that of the topology adaptive clustering algorithm TACA. In LID algorithm the lowest ID node is always biased to become the cluster head. Thus it consumes its battery power very fast and becomes dead. Similarly, WBCA fails to achieve better result in terms of the network life time. TACA makes a random selection of head nodes (volunteer or non-volunteer) removing the biasness of any node to act as a head so that the consumption of energy is fairly distributed among them. This improves the node life time and as a whole the network life time as indicated in Figure 7.

When TRAP is implemented with TACA, the life time is compromised as in the above figure. This is due to the increased transmission range of nodes by the implementation of TRAP. It is understood from the figure that the network life time is still more than that of LID and WBCA while considering the maximum value of the Range$_{MAX}$ from Figure 6.

5.4. Average Cluster Density. The ultimate purpose of clustering is to design a virtual back bone with minimum cluster density. This reduces the communication latency as the heads take part in packet routing through the virtual backbone. As seen in the Figure 8, TACA results in more numbers of cluster heads in comparison to LID and WBCA. This is because
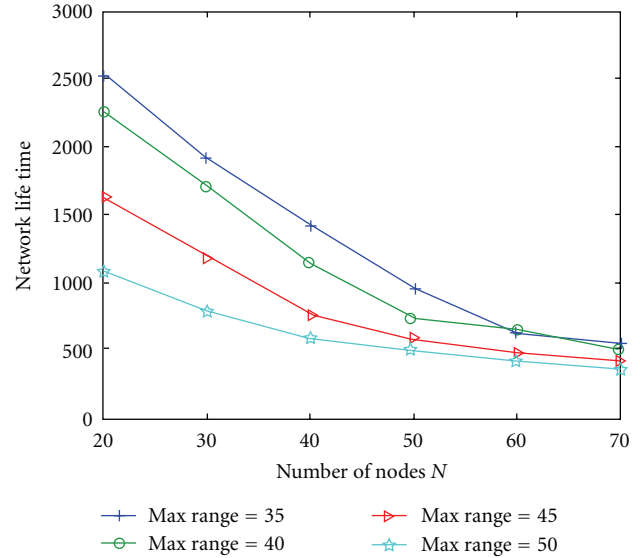
the increased number of isolated cluster heads selected in TACA, especially during the member hand-off process of non-volunteer head selection.

As the transmission range adjustment protocol TRAP is implemented on TACA, the density of heads is reduced to a great extent. Usually for the low transmission range of the nodes the number of heads is more because of their isolation from each other. However, TRAP solves this problem by allowing the nodes to adjust their transmission range so that the nodes get opportunity to become the members of the existing heads in spite of becoming a cluster head. This in return reduces the density of cluster heads.

6. Conclusion

TACA is energy efficient and topology adaptive distributed clustering algorithm that ensures better cluster stability and enhances the network life time. Being a topology adaptive algorithm, it eliminates the freezing time of motion of mobile nodes during the cluster setup. A node with lower mobility and higher available battery power is chosen for cluster head so that cluster stability is improved. Introducing the selection of non-volunteer nodes reduces the number of global reelections for the cluster heads. At the same time the load on nodes acting as heads is also minimized enhancing the network life time. But TACA results in slightly higher cluster population that increases the number of members in the virtual back bone of communication. To further improve the clustering results, a transmission range adjustment protocol has been designed that enables a node to adjust its transmission range as and when required. The new design helps in reducing the number of isolated cluster heads and eventually the number of hops in the back bone. It is seen that when the transmission range is adjusted for certain nodes, the clustering algorithm reduces the density of clusters as well as the number of reelections in the network without compromising the network life time.

References

[1] S. Basagni, M. Conti, S. Giordano, and I. Stojmenovic, *Mobile Ad Hoc Networking*, IEEE Press, Wiley Interscience, Hoboken, NJ, USA, 2004.

[2] A. K. Parekh, "Selecting routers in ad hoc wireless network," in *Proceedings of the SBT/IEEE International Telecommunication Symposium (ITS '94)*, August 1994.

[3] S. Basagni, I. Chlamtac, and A. Farago, "A generalized clustering algorithm for peer-to-peer networks," in *Proceedings of the Workshop on Algorithmic Aspect of Communication*, July 1997.

[4] A. Ephremides, J. E. Wieselthier, and D. J. Baker, "A design concept for reliable mobile radio networks with frequency hopping signaling," *Proceedings of the IEEE*, vol. 75, no. 1, pp. 56–73, 1987.

[5] J. Chang and L. Tassiulas, "Energy conserving routing in wireless ad-hoc networks," in *Proceedings of the 19th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '00)*, pp. 22–31, Tel-Aviv, Israel, March 2000.

[6] D. J. Baker and A. Ephremides, "The architectural organization of a mobile radio network via a distributed algorithm," *IEEE Transactions on Communications*, vol. 29, no. 11, pp. 1694–1701, 1981.

[7] M. Gerla and J. T.-C. Tsai, "Multicluster, mobile, multimedia radio network," *Wireless Networks*, vol. 1, no. 3, pp. 255–265, 1995.

[8] S. Chinara and S. K. Rath, "A survey on one-hop clustering algorithms in mobile ad hoc networks," *Journal of Network and Systems Management*, vol. 17, no. 1-2, pp. 183–207, 2009.

[9] S. Chinara and S. K. Rath, "Energy efficient mobility adaptive distributed clustering algorithm for mobile ad hoc network," in *Proceedings of the 16th International Conference on Advanced Computing and Communications (ADCOM '08)*, pp. 265–272, Chennai, India, December 2008.

[10] P. Basu, N. Khan, and T. D. C. Little, "A mobility based metric for clustering in mobile ad hoc networks," in *Proceedings of IEEE Workshop on Wireless Networks and Mobile Computing (ICDCS '01)*, Phoenix, Ariz, USA, 2001.

[11] S. Basagni, "Distributed and mobility-adaptive clustering for multimedia support in multi-hop wireless networks," in *Proceedings of Vehicular Technology Conference (VTC '99)*, vol. 2, pp. 889–893, 1999.

[12] S. Basagni, "Distributed clustering for ad hoc networks," *Proceedings of the International Symposium on Parallel Architectures, Algorithms and Networks (I-SPAN '99)*, pp. 310–315, 1999.

[13] R. Ghosh and S. Basagni, "Limiting the impact of mobility on ad hoc clustering," in *Proceedings of the 2nd ACM International Workshop on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks (PE-WASUN '05)*, pp. 197–204, Montreal, Canada, October 2005.

[14] S. Das, M. Chatterjee, and D. Turgut, "Wca: a weighted clustering algorithm for mobile ad hoc networks," *Journal of Cluster Computing*, vol. 5, no. 2, pp. 193–204, 2002.

[15] M. Chatterjee, S. K. Das, and D. Turgut, "An on-demand weighted clustering algorithm (WCA) for ad hoc networks," in *Proceedings of IEEE Global Telecommunication Conference (GLOBECOM '00)*, pp. 1697–1701, San Francisco, Calif, USA, 2000.

[16] W.-D. Yang and G.-Z. Zhang, "A weight-based clustering algorithm for mobile ad hoc Network," in *Proceedings of the 3rd International Conference on Wireless and Mobile Communications (ICWMC '07)*, Guadeloupe, March 2007.

[17] V. Rodoplu and T. H. Meng, "Minimum energy mobile wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 8, pp. 1333–1344, 1999.

[18] R. Ramanathan and R. Rosales-Hain, "Topology control of multihop wireless networks using transmit power adjustment," in *Proceedings of the 19th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '00)*, pp. 404–413, March 2000.

[19] K. Paul and S. Bandyopadhyay, "Self-adjusting transmission range control of mobile hosts in ad hoc wireless networks for stable communication," in *Proceedings of the 6th International Conference on High Performance Computing (HiPC '99)*, Kolkata, India, 1999.

[20] S. Chinara and S. K. Rath, "TACA: Topology Adaptive Clustering Algorithm for mobile ad hoc networks," in *Proceedings of the International Conference on Wireless Networks (ICWN '09)*, vol. 2, pp. 391–397, Las Vegas, Nev, USA, July 2009.

[21] R. Dube, C. D. Rais, K.-Y. Wang, and S. K. Tripathi, "Signal stability-based adaptive routing (SSA) for ad hoc mobile networks," *IEEE Personal Communications*, vol. 4, no. 1, pp. 36–45, 1997.

[22] L. M. Feeney, *An Energy Consumption Model for Performance Analysis of Routing Protocols for Mobile Ad Hoc Networks , Mobile Networks and Applications*, vol. 6, Kluwer Academic Publishers, Dordrecht, The Netherlands, 2001.

[23] Q. Gao, K. J. Blow, D. J. Holding, I. W. Marshall, and X. H. Peng, "Radio range adjustment for energy efficient wireless sensor networks," *Ad Hoc Networks*, vol. 4, no. 1, pp. 75–82, 2006.

[24] P. Santi, *Topology Control in Wireless Ad Hoc and Sensor Networks*, John Wiley & Sons, New York, NY, USA, 2005.

[25] T. Camp, J. Boleng, and V. Davies, "A survey of mobility models for ad hoc network research," *Wireless Communications and Mobile Computing*, vol. 2, no. 5, pp. 483–502, 2002.

[26] G. Lin, G. Noubir, and R. Rajaraman, "Mobility models for ad hoc network simulation," in *Proceedings of the 23rd Annual Joint Conference on Computer Communications (IEEE INFOCOM '04)*, pp. 454–463, Hong Kong, March 2004.

[27] L. M. Feeney and M. Nilsson, "Investigating the energy consumption of a wireless network interface in an ad hoc networking environment," in *Procedings of the 20th Annual Joint Conference of the IEEE Computer and Communications Societies*, pp. 1548–1557, Anchorage, Alaska, USA, April 2001.

[28] E. Lochin, A. Fladenmuller, J.-Y. Moulin, and S. Fdidda, "Energy consumption models for ad-hoc mobile terminals," Paris Cedex 05, France, 2005.