TACA: A Topology Adaptive Clustering Algorithm For Mobile Ad Hoc Network

Suchismita Chinara and Santanu Kumar Rath [2]
Department of Computer Science and Engineering,
National Institute of Technology, Rourkela, Orissa, India.

Abstract – *The research on various issues in Mobile ad hoc networks are getting popularity because of its challenging nature and all time connectivity to communicate. We propose a topology adaptive clustering algorithm for mobile ad hoc network. Two major node parameters like its mobility and available battery power are considered for the node suitability as cluster head. This results into a faster cluster set up time. Non-volunteer cluster heads are selected locally as and when required. This improves the network life time and reduces the maintenance over head.*

Keywords: Mobile ad hoc network, clustering, volunteer and non-volunteer cluster heads, node operating mode, energy consumption.

1. Introduction

The interest and research on wireless networks as well as the communication on fly has taken an exponential growth in the recent era. Mobile ad hoc networks (MANET) consist of mobile hosts that move freely while remaining reachable to each other. With limited transmission range the devices are capable to communicate using intermediate relays or multi-hop wireless links. So a mobile device is required to play the role of a router for forwarding the packets of its neighbors in the dynamic environment where the node movement causes frequent topology changes. The link creation and deletion due to such topology change, scarcity of radio resources and bandwidth, limited battery power and computing power of nodes pose challenge in MANET scalability and efficiency [2]. In such a scenario, a hierarchical approach based on partitioning the network into logical groups is a proven solution to meet the scalability issue as well as to obtain better fault management.

In conventional cellular network the mobile nodes communicate directly with the fixed base station. 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 [6]. In such a virtual cellular architecture (VCA) cluster heads are selected to play the role of base stations of cellular architecture. The cluster head along with its one-hop members form the virtual cells retaining the merits of cellular structure. The cluster head within each cluster that acts as the local coordinator for its member nodes also guarantees for faster communication [3]. The cluster head in every cluster does the resource allocation to the cluster members and lies responsible for inter-cluster communication.

The process of clustering is never completed without a proper maintenance scheme. The objective of cluster maintenance is to preserve as much as of the existing clustering structure as possible. The node movement in the network results in frequent link failure or link establishment between the nodes. This demands cluster member updation to take place from time to time. Moreover, the changing topology and node lifetime / capability (with respect to its available battery power) eliminate the possibility of permanent cluster heads. Thus new cluster heads are required to be elected with the changing scenario. Hence, a well designed clustering algorithm needs to follow a least maintenance overhead phase.

Cluster heads being the communication hotspots tend to drain its battery power rapidly while serving its member nodes [8]. Further, energy consumption is a key factor that hinders the deploy ability of a real ad hoc and sensor network. It is due to the limited life time of the battery powered devices that motivates intense research into energy efficient design of operating systems, protocols and hardware devices. The rest of the paper is organized as follows. Section II describes the work done in the related area. Section III describes the proposed algorithm for the selection of volunteer and non-volunteer cluster heads. Section IV explains the simulation results of the algorithm in terms of the cluster maintenance parameters and network life time. Finally section V concludes the paper.

2. Related Work

The concept of partitioning of the random dynamic network into logical clusters (also called as the *Linked Cluster Algorithm LCA)* was initially proposed by Baker and Ephremides [7]. The existing one-hop clustering algorithms emphasize either on minimizing number of cluster heads ([1], [2]) in the virtual back bone to reduce the routing delay or maximizing the cluster stability by un-altering the head nodes ([3],[7]).

A small variation to LCA was proposed by Ephremides, Wieselthier and Baker in [3] as a lowest ID algorithm. In this algorithm a node having lowest ID among its neighbors is selected as the head node. It retains its utility as a benchmark for producing reasonably stable cluster control architecture as discussed by Gerla and Tsai in [1]. 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 their faster energy drainage which in turn perturbs the cluster stability [24].

A modified version of LCA was proposed by Parekh [2] that aims to reduce the number of clusters in the network. If N_i represents the set of neighbors 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. A higher degree of connectivity ensures lower delay in communication through cluster heads. However, 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 which leads to more number of cluster head reelections as well as link updations resulting poor cluster stability.

A mobility metric based version of lowest ID algorithm MOBIC was proposed by Basu, Khan and Little [22]. The algorithm uses mobility based metric for calculation of weights of the nodes by using the ratio of two consecutive signal strengths received by a node to know its relative motion with respect to its neighbors. Once the relative mobility metric for every node is decided, MOBIC is called upon the nodes which works almost same as the Lowest ID algorithm, where the node IDs are replaced by the relative mobility metrics of each node. When two cluster heads accidentally come within their transmission range, re-clustering is deferred for Cluster_Contention_Interval (*CCI*) period as per the LCC [4] algorithm. Though MOBIC provides a better cluster stability, but the need to collect relative speed information of a node from its neighbors degrades its performance.

The author in ([4], [9]) proposed a weight based distributed mobility adaptive algorithm DMAC that removes the non-mobility assumption of the hosts during clustering setup and maintenance. However, when two head nodes come within the transmission range of each other; the node with the lower weight has to resign its role as head and is forced to become the member node of the node with higher weight. This non-neighborhood restriction of two cluster heads results frequent reaffiliations of the member nodes and the rate of cluster heads updation. Moreover, the condition of affiliation of member nodes to a head with higher weight than its present head (if it finds any at any time within its transmission range) further increases the reaffiliation rate reducing the clustering efficiency.

The combined metric clustering algorithms ([5], [10], [12]) use some node parameters like running average, degree of connectivity or mean connectivity, transmission power, available battery power or consumed battery power to find its suitability as a head. But obtaining so much of information (specially, mean connectivity in a dynamic network) 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 take place globally to yield the node with lowest weight. The authors in [18] have proposed an energy efficient cluster design which is possible either in case of a static network or where the cluster head is known a priori. For a real ad hoc network these conditions are never satisfied.

The algorithm we propose here is topology adaptive clustering algorithm **TACA**. It considers both the mobility and available battery power of the node as the weight deciding factor. As the network is activated clusters are formed with the election of volunteer cluster heads. When any of these volunteer head drains a threshold amount of its battery power then it selects a non-volunteer head within its own cluster so that it can hand over the responsibilities to the newly selected head locally. This avoids as much as possible to a global cluster head election procedure that demands considerable computation and communication overhead.

3. TACA: topology Adaptive clustering

3.1 Basics of the Algorithm

The mobile ad hoc network can be modeled as a unidirectional graph G = (V, L) where V is the set of mobile nodes and L is the set of links that exist between the nodes. We assume that there exists a bidirectional link L_{ij} between the nodes i and j when the distance between the nodes $d_{ij} < t_{range}$ (transmission range) of 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.

The preliminary version of this work is presented in [24]. This algorithm proposes for the selection of a non-volunteer cluster head local to a cluster. That is when the volunteer cluster head of a cluster drains half of its available battery power while working as a cluster head; it selects a local member within its own cluster zone with maximum battery power. This selected node then after works as the non-volunteer cluster head for the members of the previous head those lie within its transmission range. The members of the former head those who don't lie within its transmission range reaffiliate with any other cluster head in the proximity or become isolated heads. This process continues till there remains any node with enough battery power within the cluster. The weakness of [24] is that, it considers only the available battery power of the local node for selecting it as head. But the mobility of the node is absolutely ignored here. Thus a high mobile node can be selected at any instant as the non-volunteer head. Such a situation can completely disturb the stability of the network contradicting its basic goal to obtain better cluster stability. The proposed topology adaptive clustering algorithm (TACA) is extension of the earlier work so that the mobility and battery power are considered as the deciding parameters in the network.

3.2 Selection of Volunteer Head in TACA

TACA is a distributed algorithm that takes into account the mobility of a node and its available battery power as the parameters to decide its suitability as a cluster head. Let δ be the maximum permissible speed of any network. The average of last n displacements gives the average speed of any node. Thus the difference of δ and average speed finds the mobility factor of a node. A large mobility factor indicates a slower node and small mobility factor indicates a faster node. Available battery power is the energy contained in the node at the instant of weight calculation. These two parameters are added with different weight factors to find the weights of the individual nodes. The steps for calculating the weights of the nodes are described below:

Step 1: The total distance covered by a node during last n seconds is

$$D_t = \sum_{i=t-n}^{i=t} Dist_i \qquad \text{where i= t is the current time.}$$

Compute average speed of a node is $Sv = D_t / n$.

Step 2: Compute Mobility factor $\Delta M = \delta - Sv$. That is how far is the average speed of the node from the maximum permissible speed δ of the network.

Step 3: Compute available battery power as

$$P_{av} = P_{av} - P_{cons} \quad \text{where}$$

P_{av} = Available battery power of the node (Initially it is the maximum battery power).

P_{cons} = Battery power consumed by the node.

Step 4: Compute the weight of node as

$$w_v = x_1 \Delta M + x_2 P_{av}$$

Where x_1 and x_2 are the weight factors

Once the weights of the nodes are calculated, the following algorithm is executed to select the set of *volunteer cluster heads.*

TABLE I
PSEUDO CODE FOR VOLUNTEER CLUSTER HEAD SELECTION

For (every $v \in V$ **)**
If $w_v \rangle w_i$ **where** $i \in \Gamma(v)$ $//\Gamma(v)$ **is the neighbor set of** v
Then Set head= v
For (every $x \in V_{uncovered}$**)**
If dist (head, x) $\langle =$ **head**$_{trange}$ **Then**
Set HEAD $_x$ **= head**
End for
End for

The algorithm indicates that a node having maximum weight among its 1-hop neighbors declares itself as the cluster head. So such a head is called as the volunteer 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 head or a member.

During the cluster head selection phase every node broadcasts it's ID along with its weight W_i to all neighbors and stores the weights W_j that it hears from other nodes. If it does not hear another node ID with weight higher than itself then it becomes a cluster head and its one-hop uncovered neighbor nodes become its members. In case of a tie in the node weights the low ID node is preferred for the role of cluster head. Unlike [4] once a member node is affiliated with a cluster head, it does not re-affiliate 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 re-affiliations lowering the cluster maintenance cost.

The example of cluster setup phase of the proposed algorithm is demonstrated with the help of Figure 1. Here every node is identified with a unique ID and its associated weight in parenthesis. We assume that the weights are already being computed for every node. The link between every pair of nodes denotes that they are within the transmission range of each other and establish a bidirectional link among each other. Volunteer cluster heads are identified after the exchange of their weights within the local topology. A node having the higher weight among its 1-hop neighbors become the head and its immediate uncovered neighbors become its members.

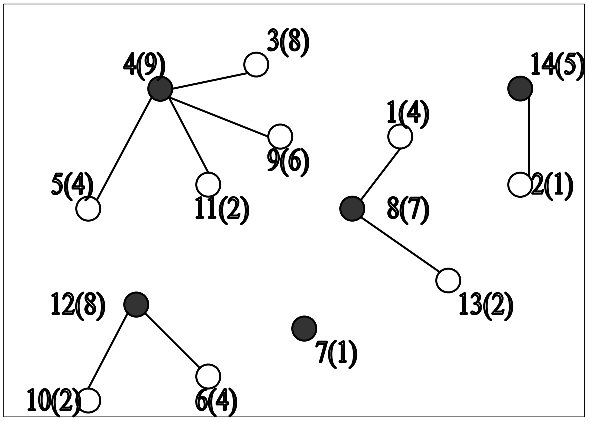

Figure 1: Volunteer clusters are formed

Here, the dark circled nodes indicate the cluster heads and the ordinary circles denote the member nodes. As we know this is only the virtual partition of the nodes. And this structure changes with every topology change in the dynamic network.

When the clusters are formed in the network, energy consumption of individual nodes is computed from time to time. As discussed in [24] energy 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 in any of the following operating modes:

(i) *transmit:* for transmitting data
(ii) *receive:* for receiving data
(iii) *idle :* a default mode when the node is ready to transmit or receive
(iv) *sleep :* the low power consumption state when a node can not transmit or receive until woken up.

The authors of [19] provides an incremental cost *m* and a fixed cost *c* for a broadcast communication as

$$Energy_{member} = m_{send/receive} \;\; X \;\; size_{packet} \; + \; c_{broadcast}$$

In a broadcast traffic, the sender listens briefly to the channel. If the channel is found to be free then the packet is sent and is received by all nodes in wireless range. If the channel is found busy, the sender has no choice but to back off and retry later. In order to avoid the complexity of retransmission, we have ignored the situation of retransmissions, acknowledgements as well as discarding cost of overhear packets by the hidden and exposed terminals.

As proposed in [24] we consider the energy model for cluster heads as

$$Energy_{head} = \alpha * |n_i| + \beta * Traffic_{bcast} + \gamma * \sum_{v' \in n_i} dist(v, v')$$

Where $|n_i|$ represents the cardinality of the cluster, $Traffic_{bcast}$ is the cost of energy consumption in traffic forwarding and $\sum_{v' \in n_i} dist(v, v')$ is the total transmission power utilized in communicating the member nodes of the cluster head. 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 age than the other two. Similarly, in a dense network where the cardinality of clusters are more, the weight age of α dominates the other factors. All three parameters are chosen so that $\alpha + \beta + \gamma = 1$.

As a whole, we consider the energy consumption of a head node basically depends 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.**

3.3 Selection of Non-volunteer cluster head

The need for selection of a new non-volunteer head arises when the current head (either the volunteer or non-volunteer head) drains its battery power to half of its available battery power while acting as a cluster head. During this phase, the current head selects one of its member nodes inside the cluster with maximum weight and invites it to take the role of the cluster head. Maximum weight of a node ensures the low mobility and high available battery power. But, it is the choice of the selected node to accept the cluster head role or not depending on its available resources. The selection process takes place locally within a cluster reducing the computation and communication overhead that would have yield in the global one. The algorithm for finding the non-volunteer head is as:

TABLE II
PSEUDO CODE FOR NON-VOLUNTEER CLUSTER HEAD SELECTION

Set i **= current_head //volunteer or non-volunteer**

Set max_wt= maximum (w_v **) where** $v \in cluster_i$

Set next_head= v_{max_wt}

Head (i **) = next_head**

For (every $v_{member} \in cluster_i$ other than next_head)

If dist (next_head, v_{member}) $\langle =$ next_head $_{Trange}$ Then

Head (v_{member}) =next_head

Else

Reaffiliate v_{member} to other head within range

Else

Select v_{member} as volunteer head

End if
End for

4. Simulation Results and Discussion

The simulation of TACA is carried by using the Random Walk mobility model. This mobility model ([14], [15]) represents the most erratic and unpredictable movement of an entity. N nodes are moved in a 100 x 100 grid area. The maximum speed δ of nodes is 5 m/sec within the network. In our simulation the running average of every node is calculated for 5 unit of time elapsed.

The energy consumption of member nodes for different operating modes is considered ([17], [19]) as:

Broadcast send= 1.9 μ W.s/byte * size $_{packet}$ +250 μ W.s

Broadcast recv= 0.50 μ W.s/byte * size $_{packet}$ +56 μ W.s

Idle = 808 mW

where the packet size is taken as 1024 bits for the simulation. TACA makes the selection of head nodes as per their suitability to do so by eliminating the biasness of any node so that the consumption of energy is fairly distributed among them.

The available battery power of a node decides its life time. In the simulation we have assumed that when a node consumes 90% of its maximum battery power it becomes dead. And the network fails when a single node becomes dead. Figure 2 gives the result of network life time for LID algorithm and our proposed TACA. In LID the lowest ID node is always biased to become the cluster head. Thus it consumes its battery power very fast and becomes dead. But TACA gives a fair chance to every other node to serve as a head. Thus the consumption of battery power is nearly uniform for every node. This improves the node life time and as a whole the network life time.

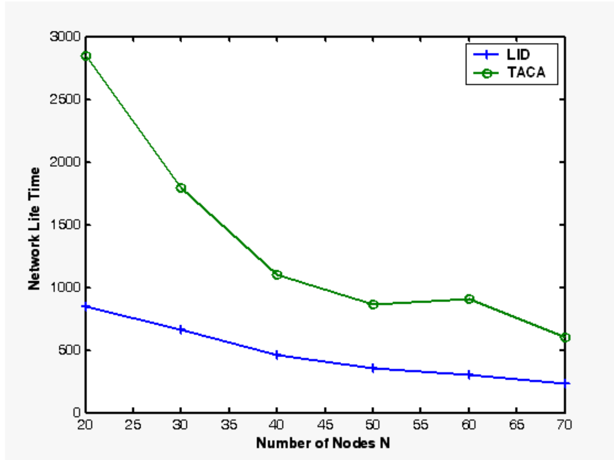

Figure 2: comparison of network life time for LID and TACA.

Custer maintenance of TACA involves three major operations like reaffiliation of member nodes, local reelection for selecting non-volunteer cluster heads and a global reelection for selecting volunteer cluster heads.

A reaffiliation occurs when:

- A member node leaves its current head's transmission zone and enters into another head's cluster zone.

- When a non-volunteer head is elected so that member nodes of the existing head reaffiliates to the new head (if exists within the transmission range) or to other heads within their transmission range.

Figure 3 gives the comparison of algorithms for their node reaffiliation rate with different cluster heads for various transmission ranges. It is well understood from the figure that TACA has a lower reaffiliation rate than other algorithms. The lower reaffiliation reduces the communication and computation complexity.

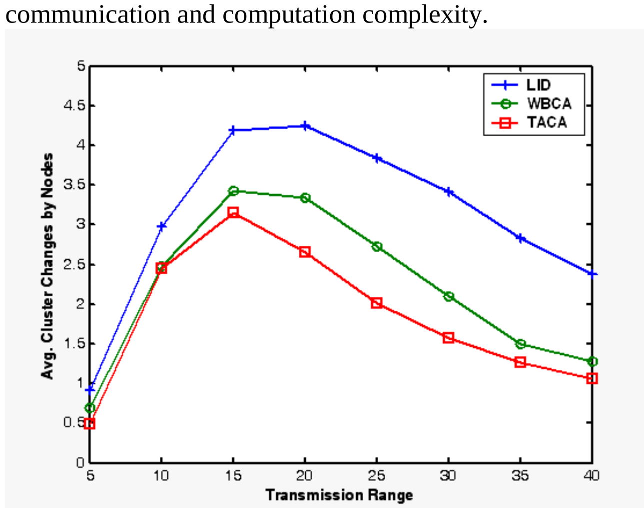

Figure 3: Comparison of algorithms for reaffiliation, N=50 and δ =5.

A local reelection of non-volunteer cluster head takes place when:

- An existing cluster head consumes the threshold amount of battery power; so that it resigns from its current role and hands off its role to a member node with maximum weight.

The global reelection of volunteer cluster heads take place when:

- A single node becomes orphan or isolated by moving away from all other nodes declaring itself as a volunteer cluster head.

- All the nodes of a cluster consumes threshold amount of battery power and become unable to serve as cluster head.

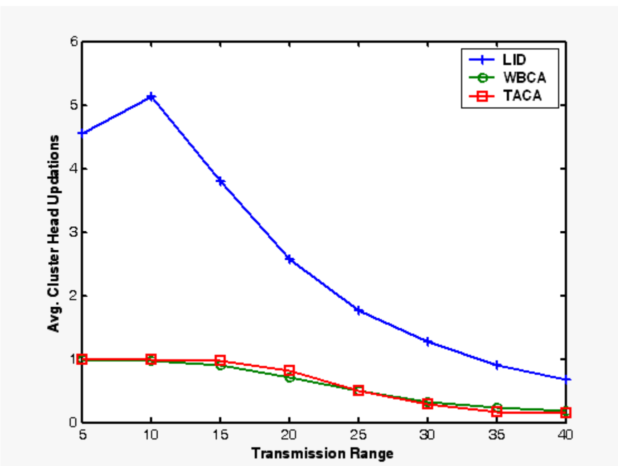

Figure 4: Comparison of algorithms for reelection, N=50 and δ =5.

Comparison of head updation rate for different algorithms is indicated in Figure 4. The updation cost of TACA is almost same as that of WBCA. As seen in the figure it has a much better result than LID algorithm. This ensures better cluster as well as routing stability of the network.

5. 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. We keep a record of previous n set of movements of every node to predict their average mobility. A node with lower mobility and higher battery power is chosen for cluster head so that cluster stability can be improved. Introducing the selection of non-volunteer nodes reduces the number of global reelection complexity and load on individual nodes.

References

[1] M.Gerla and J.T.C.Tsai, "multicluster, mobile,multimedia radio network", *wireless networks*, 1,3,pp.255-265,1995.

[2] A.K.Parekh, "Selecting routers in ad hoc wireless network", *Proceedings of the SBT/IEEE international telecommunication symposium*, August 1994.

[3] D.J.Baker, J.E.Wieselthier and A. Ephremides, "A design concept for reliable mobile radio networks with frequency hoping signaling", *proceedings of IEEE,* vol. 75, no.1, Jan. 1987.

[4] S.Basagni, "Distributed and mobility-adaptive clustering for multimedia support in multi-hop wireless networks", *proceedings of vehicular technology conference*, *VTC*, vol.2, pp. 889-893, 1999.

[5] S.Das, M.Chatterjee, D.Turgut, "Wca: A weighted clustering algorithm for mobile ad hoc networks", *Journal of Cluster computing (special issue on mobile ad hoc networks)*, vol. 5, no. 2, 4, pp. 193-204, 2002.

[6] S. Basagni, I. Chlamtac, A. Farago, "A generalized clustering algorithm for peer-to-peer networks", *Proceedings of Workshop on Algorithmic aspect of communication* (satellite workshop of ICALP), July 1997.

[7] D.J.Baker and A. Ephremides, " The architectural organization of a mobile radio network via a distributed algorithm", *IEEE Transactions on Communications COM-29*, pp. 1694-1701, 11(Nov.1981).

[8] J.H.Chang and L.Tassiulas, "Energy conserving routing in wireless ad hoc networks", *Proceedings of INFOCOM 2000*, Tel-Aviv, Israel, March 2000.

[9] S. Basagni,"Distributed clustering for ad hoc networks", *Proceedings of International Symposium on Parallel Architectures, Algorithms and Networks*, pp.310–315, June 1999.

[10] S.Das, M.Chatterjee, D.Turgut, "An on demand weighted clustering algorithm(WCA) for ad hoc networks", *in proceedings of IEEE GLOBECOM 2000*, pp.1697-1701, San Francisco, Nov. 2000.

[11] T.Ohta, S.Inoue, Y.Kakuda, "An adaptive clustering scheme for highly mobile ad hoc networks", *in proceedings of sixth international symposium on autonomous decentralized systems(ISADS'03)*, April 2003.

[12] Wei-dong Yang and G-z Zhang, "A weight-Based Clustering Algorithm for mobile Ad Hoc Network", *proceedings of Third International Conference on Wireless and Mobile Communication* (ICWMC'07), 2007.

[13] S.Basagni, M. Conti, S.Giordano and I.Stojmenovic, *Mobile ad hoc networking,* A John Wiley & Sons Inc. publication, 2004.

[14] T.Camp, J.Boleng, V.Davies. "A Survey of Mobility Models for Ad Hoc Network Research" Wireless Communications and Mobile Computing (WCMC): Special issue on *Mobile Ad Hoc Networking: Research, Trends and Applications,* vol. 2(5), pp. 483-502, 2002.

[15] G.Lin, G.Noubir and R. Rajaraman, "Mobility models for ad hoc network simulation", *IEEE INFOCOM 2004*.

[16] E.Lochin, A. Fladenmuller, J-Y Moulin and S. Fdidda, “Energy consumption models for ad-hoc mobile terminals”, *Paris Cedex 05*, France, 2005.
[17] L.M. Feeney and M Nilsson, “ Investigating the Energy Consumption of a Wireless Network Interface in an Ad Hoc Networking Environment”, *Proceedings of IEEE INFOCOM 2001*, Anchorage, Alaska, vol.3, pp. 1548-1557, 2001.
[18]] C-F Chiasserini, I. Chlamtac, P. Monti, A. Nucci, “ Energy Efficient Design of Wireless Ad Hoc Networks”, *proceedings of European wireless*, February 2002.
[19] L.M. Feeney, “ An Energy Consumption Model for Performance Analysis of Routing Protocols for Mobile Ad Hoc Networks” , *Mobile Networks and Applications , Kluwer Academic Publishers*, vol. 6, pp. 239-249, 2001.
[20] 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* , vol4, pp.75-82, 2006.
[21] S. chinara, S.K.Rath, P. Bhattad and A. Dey, “Mobility Based Distributed Clustering Algorithm for Mobile Ad Hoc Network”, *Proceedings of NCNTE-2008*, Vashi, Mumbai, pp. 169-172, 2008.
[22] P.Basu, N.Khan and T.D.C. Little, “A mobility based metric for clustering in mobile ad hoc networks”, *Proceedings of IEEE ICDCS 2001 workshop on wireless networks and mobile computing*, Phoenix, AZ, 2001.
[23] R. Ghosh and S. Basagni, “ Limiting the impact of mobility on ad hoc clustering”, *Proceedings of the 2nd ACM international workshop PE-WASUN '05*, Montreal, CA, pp. 197--204, 2005.
[24] S.Chinara and S.K. Rath, “Energy Efficient mobility adaptive distributed clustering algorithm for mobile ad hoc network”, *in Proceedings of sixteenth IEEE international conference on Advanced Computing and Communication* (*ADCOM 2008)*, pp. 265-272, Chennai, Dec. 2008.