Efficient Clustering for Mobile Ad Hoc Network

Ratish Agarwal[1], Dr. Roopam Gupta[2], Dr. Mahesh Motwani[3]

[1, 2] Department of IT, RGPV Bhopal
[3] Department of CSE, JEC Jabalpur
[1]ratish@rgtu.net
[2]roopamgupta@rgtu.net
[3]mahesh_9@hotmail.com

Abstract: This paper proposes a technique for election of nodes called clusterheads to coordinate the cluster activities. There are a number of algorithms for election of clusterhead; some of them consider only one specific property of node such as node's id or degree for clusterhead selection. Some of the algorithms take in to account several performance factors such as node degree, mobility, and energy requirement and battery power of nodes. However, the entire cluster based protocol and cluster formation algorithm that have been proposed assume that the wireless nodes are trustworthy. This assumption may naturally lead to the selection of a compromised or malicious node to be the cluster head. Having a malicious cluster-head severely compromises the security and usability of the network. This paper proposes a method which calculates the trustworthiness of a node and takes it as a performance factor while electing clusterhead.

Keywords: Ad Hoc Networks, clusterhead, Dominating set, mobility, trust.

(Received January 07, 2010 / Accepted May 07, 2010)

1 Introduction

Ad hoc networks are wireless, infrastructure less, multi-hop, dynamic networks established by a collection of mobile nodes. This type of network is highly demanding due to the lack of infrastructure, cost effectiveness and easiness in installation. Mobile ad hoc network (MANET) has many emerging applications, which include commercial and industrial, war front applications, search and rescue operations, sensor networks and vehicular communications. The major issues in cluster based MANETs are (i) mobility management (ii) topology assignment (iii) clustering overhead (iv) frequent leader reelection (v) overhead of clusterhead (vi) depletion of battery power (vii) security and (viii) Quality of Service (QoS).

There is no stationary infrastructure; for instance, there are no base stations. Each node in the network also acts as a router, forwarding data packets for other nodes. A research issue in the design of ad hoc networks is the development of dynamic routing protocols that can efficiently find routes between two communicating nodes. The routing protocol must be able to keep up with the high degree of node mobility that often changes the network topology. In a large network, flat routing schemes produce an excessive amount of information that can saturate the network.

In addition, given the nodes heterogeneity, nodes may have highly variable amount of resources, and this produces a hierarchy in their roles inside the network. Nodes with large computational and communication power, and powerful batteries are more suitable for supporting the ad hoc network functions (e.g., routing) than other nodes.

Cluster-based routing is a solution to address nodes heterogeneity, and to limit the amount of routing information that propagates inside the network. The idea behind clustering is to group the network nodes into a number of overlapping clusters. Clustering makes possible a hierarchical routing in which paths are recorded between clusters instead of between nodes. This increases the routes lifetime, thus decreasing the amount of routing control overhead. Inside the cluster one node that coordinates the cluster activities is clusterhead (CH). Inside the cluster, there are ordinary nodes also that have direct access only to this one clusterhead, and gateways. Gateways are nodes that can hear two or more clusterheads.

Ordinary nodes send the packets to their clusterhead that either distributes the packets inside the cluster, or (if the destination is outside the cluster) forwards them to a gateway node to be delivered to the other clusters. By replacing the nodes with clusters, existing routing protocols can be directly applied to the network. Only gateways and clusterheads participate in the propagation of routing control/update messages. In dense networks this significantly reduces the routing overhead, thus solving scalability problems for routing algorithms in large ad hoc networks.

Several algorithms have been proposed by researchers for formation of cluster and election of clusterhead. A brief overview of some of them has given in next section.

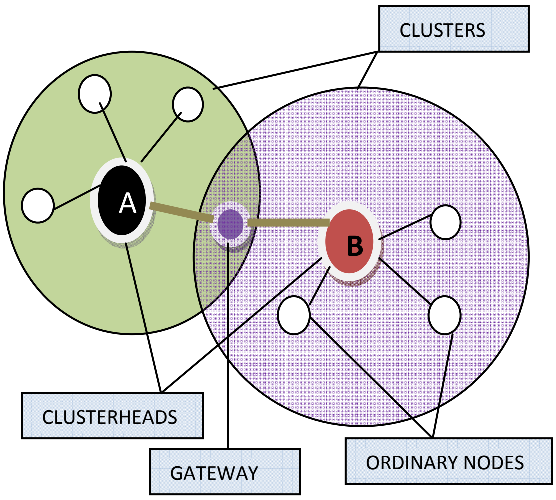

Figure 1: Cluster based routing in MANET

This paper proposes an approach in which clustering set up phase is accomplished by a Dominating Set finding algorithm for choosing some nodes that act as coordinators of the clustering process. Dominating nodes are potential nodes to become clusterheads and during the cluster formation phase, the ordinary nodes select their best as the clusterhead. This selection is based on quality, which is a function of parameters such as stability of the dominating node with respect to its neighbors, remaining battery power with the node, energy requirement, connectivity and trust value of the node. Selection of clusterhead based on these parameters help in maintaining the structure of the created cluster as stable as possible thus minimizing the topology changes and associated overheads during clusterhead changes.

2 Related works

A number of clustering algorithms for mobile ad hoc networks have been proposed in the literature. In Lowest ID cluster algorithm (LIC) [1] algorithm a node with the minimum id is chosen as a clusterhead. Thus, the ids of the neighbors of the clusterhead will be higher than that of the clusterhead. Each node is assigned a distinct id. Periodically, the node broadcasts the list of nodes that it can hear A node which only hears nodes with id higher than itself is a clusterhead. Otherwise, a node is an ordinary node. Drawback of lowest ID algorithm is that certain nodes are prone to power drainage [2] due to serving as clusterheads for longer periods of time.

In Highest connectivity clustering algorithm (HCC) [1] the degree of a node is computed based on its distance from others. Each node broadcasts its id to the nodes that are within its transmission range. The node with maximum number of neighbors (i.e., maximum degree) is chosen as a clusterhead. This system has a low rate of clusterhead change but the throughput is low. Typically, each cluster is assigned some resources which is shared among the members of that cluster. As the number of nodes in a cluster is increased, the throughput drops. The reaffiliation count of nodes is high due to node movements and as a result, the highest-degree node (the current clusterhead) may not be re-elected to be a clusterhead even if it looses one neighbor. All these drawbacks occur because this approach does not have any restriction on the upper bound on the number of nodes in a cluster.

K-CONID [3] combines two clustering algorithms: the Lowest-ID and the Highest-degree heuristics. In order to select clusterheads connectivity is considered as a first criterion and lower ID as a secondary criterion. In HCC clustering scheme, one cluster head can be exhausted when it serves too many mobile hosts. It is not desirable and the CH becomes a bottleneck. So a new approach [4] is given in which when a CH's Hello message shows its dominated nodes' number exceeds a threshold (the maximum number one CH can manage), no new node will participate in this cluster. Adaptive multihop clustering [5] sets upper and lower bounds (U and L) on the number of clustermembers within a cluster that a clusterhead can handle. When the number of clustermembers in a cluster is less than the lower bound, the cluster needs to merge with one of the neighboring clusters. On the contrary, if the number of clustermembers in a cluster is greater than the upper bound, the cluster is divided into two clusters.

Mobility-based d-hop clustering algorithm [6] partitions an ad hoc network into d-hop clusters based on mobility metric. The objective of forming d-hop clusters is to make the cluster diameter more flexible. Local stability is computed in order to select some nodes as clusterheads. A node may become a clusterhead if it is found to be the most stable node among its neighborhood. Thus, the clusterhead will be the node with the lowest value of local stability among its neighbors. In Mobility Based Metric for Clustering [7] a timer is used to reduce the clusterhead change rate by avoiding re-clustering for incidental contacts of two passing clusterheads. Mobility-based Frame Work for Adaptive Clustering [8] partition a number of mobile nodes into multi-hop clusters based on *(a, t)* criteria. The *(a, t)* criteria indicate that every mobile node in a cluster has a path to every other node that will be available over some time period *'t'* with a probability *'a'* regardless of the hop distance between them. This is achieved using prediction of the future state of the network links in order to provide a quantitative bound on the availability of paths to cluster destinations. A metric which captures the dynamics of node mobility, makes the scheme adaptive with respect to node mobility.
Most of protocols executes the clustering procedure periodically, and re-cluster the nodes from time to time in order to satisfy some specific characteristic of clusterheads. In HCC, the clustering scheme is performed periodically to check the "local highest node degree" aspect of a clusterhead. When a clusterhead finds a member node with a higher degree, it is forced to hand over its clusterhead role. This mechanism, involves frequent re-clustering. In LCC [9] the clustering algorithm is divided into two steps: cluster formation and cluster maintenance. The cluster formation simply follows LIC, i.e. initially mobile nodes with the lowest ID in their neighborhoods are chosen as clusterheads. Re-clustering is event-driven and invoked if two clusterheads move into the reach range of each other and When a mobile node cannot access any clusterhead. Adaptive clustering for mobile wireless network [10]. ensures small communication overhead for building clusters because each mobile node broadcasts only one message for the cluster construction.

3-hop between adjacent clusterheads (3-hBAC) [11] algorithm introduce a new node status, "clusterguest", which means this node is not within the transmission range of any clusterheads, but within the transmission range of some clustermembers. When a mobile node finds out that it cannot serve as a clusterhead or join a cluster as a clustermember, but some neighbor is a clustermember of some cluster, it joins the corresponding cluster as a clusterguest.
Most of the clustering algorithms require all the mobile nodes to announce cluster-dependent information repeatedly to build and maintain the cluster structure, and thus clustering is one of the main sources of control overhead. A clustering protocol that does not use dedicated control packets or signals for clustering specific decision is called Passive Clustering [12]. In this scheme, when a potential clusterhead with "initial" state has something to send, such as a flood search, it declares itself as a clusterhead by piggybacking its state in the packet. Neighbors can gain knowledge of the clusterhead claim by monitoring the "cluster state" in the packet, and then record the Cluster head ID and the packet receiving time. A mobile node that receives a claim from just one clusterhead becomes an ordinary node, and a mobile node that hears more claims becomes a gateway. Since passive clustering does not send any explicit clustering-related message to maintain the cluster structure, each node is responsible for updating its own cluster status by keeping a timer. When an ordinary node does not receive any packet from its clusterhead for a given period, its status reverts to "initial".

Load balancing clustering (LBC) [13] provide a nearby balance of load on the elected clusterheads. Once a node is elected a clusterhead it is desirable for it to stay as a clusterhead up to some maximum specified amount of time, or budget. Initially, mobile nodes with the highest *IDs* in their local area win the clusterhead role. LBC limits the maximum time units that a node can serve as a clusterhead continuously, so when a clusterhead exhausts its duration budget, it resets its *VID* to 0 and becomes a non-clusterhead node. However, the drawback is that the clusterhead serving time alone may not be a good indicator of energy consumption of a mobile node.

Power-aware connected dominant set [14] is an energy-efficient clustering scheme which decreases the size of a dominating set (DS) without impairing its function. The unnecessary mobile nodes are excluded from the dominating set saving their energy consumed for serving as clusterheads. Mobile nodes inside a DS consume more battery energy than those outside a DS because mobile nodes inside the DS bear extra tasks, including routing information update and data packet relay. Hence, it is necessary to minimize the energy consumption of a DS. Clustering for energy conservation [15] assumes two node types: master and slave. The purpose of of this scheme is to minimize the transmission energy consumption summed by all master-slave pairs and to serve as many slaves as possible in order to operate the network with longer lifetime and better performance.

Weighted clustering algorithm (WCA) [16] selects a clusterhead according to the number of nodes it can handle, mobility, transmission power and battery power. To avoid communications overhead, this algorithm is not periodic and the clusterhead election procedure is only invoked based on node mobility and when the current dominant set is incapable to cover all the nodes. The clusterhead election algorithm finishes once all the nodes become either a clusterhead or a member of a clusterhead. The distance between members of a clusterhead, must be less or equal to the transmission range between them. No two clusterheads can be immediate neighbors. In WCA high mobility of nodes leads to high frequency of reaffiliation which increase the network overhead. Higher reaffiliation frequency leads to more recalculations of the cluster assignment resulting in increase in communication overhead. Entropy based clustering [17] overcomes the drawback of WCA and forms a more stable network. It uses an entropy-based model for evaluating the route stability in ad hoc networks and electing clusterhead. Entropy presents uncertainty and is a measure of the disorder in a system. So it is a better indicator of the stability and mobility of the ad hoc network.

Vote-based clustering algorithm [4] is based on two factors, neighbors' number and remaining battery time of every mobile host (MH) Each MH has a unique identifier (ID) number, which is a positive integer. The clustering approach presented in WBACA [18] is based on the availability of position information via a Global Positioning System (GPS). The WBACA considers following parameters of a node for clusterhead selection: transmission power, transmission rate, mobility, battery power and degree. In Connectivity, energy & mobility driven Weighted clustering algorithm (CEMCA) [19] the election of the cluster head is based on the combination of several significant metrics such as: the lowest node mobility, the highest node degree, the highest battery energy and the best transmission range. This algorithm is completely distributed and all nodes have the same chance to act as a cluster head. CEMCA is composed of two main stages. The first stage consists in the election of the cluster head and the second stage consists in the grouping of members in a cluster.

3 Proposed Methodology

Clustering provides one of the best solutions for communication in ad hoc networks due to its inherent energy saving qualities and its suitability for highly scalable networks. Clustering naturally facilitates an energy efficient technique where nodes forwards to a cluster head for processing. Clustering can be extremely effective in multicast, unicast, or broadcast communication. However, the entire cluster based protocol and cluster formation algorithm that have been proposed [20] assume that the wireless nodes are trustworthy. This assumption may naturally lead to the selection of a compromised or malicious node to be the cluster head. Having a malicious cluster-head severely compromises the security and usability of the network.

This research work is to provide a framework for distributed trust in wireless ad hoc networks, with the help of a trust model of quantitative measure of trust and, a mechanism that elects trustworthy cluster heads.

3.1 The Trust Model

A trust model configured for use in pure ad-hoc networks is given in [21]. General trust is basically the trust that one entity assigns another entity based upon all previous transactions in all situations. Each node has a watchdog mechanism that allows it to monitor the network events of other nodes. The information obtained through monitoring enable the nodes to compute and store trust levels for its neighbors. A node can get information about the successful transmission of any packet that it sent, via passive acknowledgement. In passive acknowledgement the sender node places itself in promiscuous mode after the transmission of any packet so as to overhear the retransmission by the recipient node. In addition, while one node is transmitting, all other nodes in the neighborhood are listening so that they can also determine if the message was successfully delivered. In cases where the messages are to be forwarded, neighborhood nodes can tell if the message was modified before retransmission by comparing with the message in its buffer.

Generally, passive acknowledgement provides us with the following information about the behavior of a node
1. Data packets are dumped and not retransmitted
2. Data contents have been modified
3. Unique addresses have been changed

Trust in ad-hoc networks is always in a fluid state and is continuously changing due to the mobility of the nodes. As the period of interaction with any node may be brief, it is imperative that the trust be represented as a continual range to differentiate between nodes with comparable trust levels. In our trust model we represent trust from –1 to +1 signifying a continuous range from complete distrust to complete trust. Trust computation involves an assignment of weights or importance factor to the events that were monitored and quantified. The assignment is totally dependent on the type of application demanding the trust level and varies with state and time. All nodes dynamically assign these weights based upon their own criteria and circumstances. These weights have a continuous range from 0 to +1 representing the significance of a particular event from unimportant to most important. The trust values for all the events from a node can then be combined using individual weights to determine the aggregate trust level for another node.

3.2 Computation of Trust Level

A trust level, denoted by $T_N(X_v)$ about X_v, where $0 \leq v \leq n$, is created at each node N. This is the trust level node N has computed and assigned to node X_v based on observation of node X_v's past behavior. The $T_N(X_v)$ is computed as follows:

$$T_N(X_v) = \omega_a d_1 + \omega_b d_2 + \omega_c d_3 + \omega_d c_1 + \omega_e c_2 + \omega_f c_3$$

where ω_a to ω_f are weights of corresponding trust parameters and are chosen in such a way that the sum of these five factors must be equal to unity. And d_1, d_2, d_3 and c_1, c_2, c_3 are trust parameters related to the data packets and control packets respectively. They are computed as follows

$d_1 = DPFXv / DPRFXv$
$d_2 = 1 - DPMXv / DPFXv$
$d_3 = 1 - DPAMXv / DPFXv$

$c_1 = CPFXv / CPRFXv$
$c_2 = 1 - CPMXv / CPFXv$
$c_3 = 1 - CPAMXv / CPFXv$

Where
DPFXv = Data packets forwarded by node *Xv*
DPRFXv= Data packets received for forward by that node *Xv*
DPMXv = Data packets modified by node *Xv*
DPAMXv= Data packet address modified by node *Xv*
And *CPFXv, CPRFXv, CPMXv, CPAMXv* are parameters related to control packets.

3.3 Cluster formation

Cluster formation can be performed in two steps formation of dominating set and election of clusterheads among dominating nodes. Dominating nodes are potential nodes to become clusterheads and dominating set can be formed by following the steps given below [22]

1. Initially all nodes of the network are assumed in unmarked state.
2. Every node exchanges its open neighbor set with all its neighbors.
3. Every node assigns itself as dominating node if it has two unconnected neighbors.

It is clear from the figure dark nodes have two unconnected neighbors, there they are marked as dominating nodes.

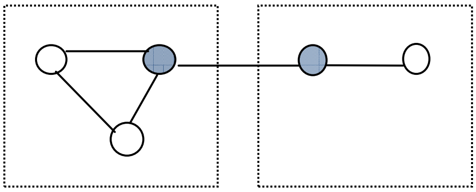

Figure 2: Marking of dominating nodes

After the formation of dominating set, dominating nodes computes its quality by assigning various weights to different parameters such as degree of the node, battery power, trust value and mobility. Dominating nodes send the message containing the quality to other nodes within r-hop distance. Each node in the network other than dominating nodes, selects the most qualified dominating node as its clusterhead. They send NODE JOIN REQ (NJ) message to the most qualified node. On receiving the NODE JOIN REQ message, the dominating node accepts the request by sending NJ ACK packet if the degree (number of accepted cluster members) of that dominating node does not exceed the threshold. If a dominating node does not receive any NODE JOIN REQ message for a specified time interval it can select the most qualified dominating node within r-hop distance as its clusterhead and can join with that cluster (Cluster merging). But the status of this node is quazi-dominating and can change to dominating node when required. This is possible only if that dominating node has k other dominating nodes within r-hop distance. This will reduce the total number of clusters created. If the ordinary node does not receive any NJ ACK messages within the stipulated time it can send NJ message to the next qualified dominating node within r-hop distance. If a node does not receive any NJ ACK messages even after k attempts and if it does not receive any new CLUSTER HEAD ADVERTISEMENT (CHA) message during the above period then that node can declare itself as a clusterhead. In case of clusterhead failure or if the battery power of the clusterhead goes below the minimum desired level, clusterhead sends this using CHA message. All the nodes attached to that clusterhead select the next qualified dominating node as its new clusterhead.

Following parameters are used for election of cluster-head according to proposed methodology.

A) Trust Value

Trust value measures how much any node in the network is trusted by its neighborhood. It's defined as the average of trust values received from each neighboring node. Nodes can collect the information from periodic exchange of HELLO message.

$$T_v = \sum_{N=1}^{n} T_N(X_v) \;\; / \, n$$

where $T_N(X_v)$ is the received trust value from node N.

B) Mobility

Mobility or stability is an important factor in deciding the clusterheads. In order to avoid frequent clusterhead changes, it is desirable to elect a clusterhead that does not move very quickly. When the clusterhead moves fast, the nodes may be detached from the clusterhead and as a result, a reaffiliation occurs. Reaffiliation can increase computation and processing, which is not a desirable feature.

Compute the running average of the speed for every node till current time T . This gives a measure of mobility and is denoted by M_v, as

$$M_v = 1/\,T \sum_{t=1}^{T} \sqrt{(X_t - X_{t-1})^2 + (Y_t - Y_{t-1})^2}$$

Where (X_t, Y_t) and (X_{t-1}, Y_{t-1}) are the coordinates of the node v at time t and $(t-1)$, respectively.

C) Degree

Each clusterhead can ideally support only δ (a pre-defined threshold) nodes to ensure efficient medium access control (MAC) functioning. If the clusterhead tries to serve more nodes than it is capable of, the system efficiency suffers in the sense that the nodes will incur more delay because they have to wait longer for their turn to get their share of the resource. A high system throughput can be achieved by limiting or optimizing the degree of each clusterhead.

The number of nodes supported by a clusterhead should be very close to the threshold value δ. Degree difference Δ_v can be a parameter of importance while electing a clusterhead. It is obtained by first calculating the number of neighbors of each node. The result of this calculation is defined as the degree of a node v, dv. To ensure load balancing the degree difference Δ_v is calculated as $|dv - \delta|$ for every node *v*. Lower the value of degree difference better the node is as clusterhead.

D) Energy Consumption

Clusterhead has to perform extra task for routing and forwarding the packets, so it is more prone to energy drainage. More power is needed for communicating long distant neighbors. We can define a parameter D_v as the sum of distances from a given node *v* to all its neighbors. This factor is related to energy consumption.

E) Power

A clusterhead consumes more battery than an ordinary node because it has extra responsibilities. We can estimate the remaining battery power by the amount of time spent by the node as a clusterhead. The parameter P_v is the cumulative time of a node being a clusterhead. P_v is used to measure how much battery power has been consumed by the node. Higher the value of P_v lower the remaining battery power.

All five parameters (T_v, M_v, Δ_v, D_v *and* P_v) explained above can be used as a performance matrix for selection of a node as a clusterhead. Weight of these parameters can change according to requirement. Weighing factors are chosen in such a way that sum of these factors must be equal to one.

Suppose we have chosen W_1, W_2, W_3, W_4 and W_5 as weighing factors for five performance parameters given above then

$$W_1 + W_2 + W_3 + W_4 + W_5 = 1$$

Combined weight of a node W_v can be calculated as follows

$$W_v = W_1.(1-T_v) + W_2.M_v + W_3.\Delta_v + W_4.D_v + W_5.P_v$$

The node with minimum weight is selected as a clusterhead.

4 Conclusions

The expected outcome of the proposed work is to develop a method for election of clusterhead taking in to account all important factors such as degree, mobility, transmission power and battery power together with a concern for security of the network. It has also the flexibility of assigning different weights and takes into account a combined effect of the ideal degree, transmission power, mobility, battery power and trust value of the nodes. The algorithm is executed only when there is a demand, i.e., when a node is unable to attach itself to any of the existing clusterheads.

References

[1] M. Gerla and J. T. Tsai, "Multiuser, Mobile, Multimedia Radio Network," *Wireless Networks*, v. 1, p. 255–65, 1995.

[2] A.D. Amis, R. Prakash, T.H.P Vuong, D.T. Huynh. "Max-Min D-Cluster Formation in Wireless Ad Hoc Networks". In proc *IEEE Conference on Computer Communications (INFOCOM)*, v. 1, p. 32-41, 2000.

[3] G. Chen, F. Nocetti, J. Gonzalez, and I. Stojmenovic, "Connectivity based k-hop clustering in wireless networks". *Proc. 35th Annual Hawaii International Conference on System Sciences*, v. 7, p. 188.3, 2002.

[4] F. Li, S. Zhang, X. Wang, X. Xue, H. Shen, "Vote-Based Clustering Algorithm in Mobile Ad Hoc Networks", proc. *International Conference on Networking Technologies, p.13-23,*2004.

[5] T. Ohta, S. Inoue, and Y. Kakuda, "An Adaptive Multihop Clustering Scheme for Highly Mobile Ad Hoc Networks," proc. *6th ISADS'03*, p.293, 2003.

[6] Er and W. Seah. "Mobility-based d-hop clustering algorithm for mobile ad hoc networks", *IEEE Wireless Communications and Networking Conference,* v.4, p. 2359-2364, 2004.

[7] P. Basu, N. Khan, and T. D. C. Little, "A Mobility Based Metric for Clustering in Mobile Ad Hoc Networks," *proc. IEEE ICDCSW' 01*, p. 413–18, 2001.

[8] B. MaDonald and T. F. Znati, "A Mobility-based Frame Work for Adaptive Clustering in Wireless Ad Hoc Networks," *IEEE JSAC*, v. 17, p. 1466–87, 1999.

[9] C.-C. Chiang *et al.*, "Routing in Clustered Multihop, Mobile Wireless Networks with Fading Channel," proc. *IEEE SICON'97*, p.197-211, 1997.

[10] C. R. Lin and M. Gerla, "Adaptive Clustering for Mobile Wireless Networks," *IEEE JSAC*, v. 15, pp. 1265–75, 1997.

[11] J. Y. Yu and P. H. J. Chong, "3hBAC (3-hop between Adjacent Clusterheads): a Novel Non-overlapping Clustering Algorithm for Mobile Ad

Hoc Networks," proc. *IEEE Pacrim'03*, v. 1, p. 318–21, 2003.

[12] T. J. Kwon *et al.*, "Efficient Flooding with Passive Clustering an Overhead-Free Selective Forward Mechanism for Ad Hoc/Sensor Networks," proc. *IEEE*, v. 91, no. 8, p. 1210–20, 2003.

[13] D. Amis and R. Prakash, "Load-Balancing Clusters in Wireless Ad Hoc Networks," proc. of *3rd IEEE ASSET'00*, p. 25–32, 2000.

[14] J. Wu *et al.*, "On Calculating Power-Aware Connected Dominating Sets for Efficient Routing in Ad Hoc Wireless Networks," *J. Commun. and Networks*, v. 4, no. 1, p. 59–70, 2002.

[15] J.-H. Ryu, S. Song, and D.-H. Cho, "New Clustering Schemes for Energy Conservation in Two-Tiered Mobile Ad-Hoc Networks," *proc. IEEE ICC'01*, v. 3, p. 862–66 , 2001.

[16] Mainak Chatterjee, Sajal K. Das and Damla Turgut, WCA: "A Weighred Clusrering Algorithm for Mobile Ad Hoc Networks", *Journal of Cluster Computing Kluwer (Springer) Academic Publishers,* v. 5, no. 2, p. 193-204, 2002.

[17] Yu-Xuan Wang, Forrest Sheng Bao, "An Entropy-Based Weighted Clustering Algorithm and Its Optimization for Ad Hoc Networks," *proc. Third IEEE International Conference on Wireless and Mobile Computing, Networking and Communications, WiMob,* p.56, 2007.

[18] S.K. Dhurandher and G.V. Singh" Weight-based adaptive clustering in wireless ad hoc networks" *IEEE,* p.95-100, 2005.

[19] F.D.Tolba, D. Magoni and P. Lorenz " Connectivity, energy & mobility driven Weighted clustering algorithm " *proc. IEEE GLOBECOM,*p.2786-2790, 2007.

[20] R. Agarwal, M. Motwani, "Survey of clustering algorithms for MANET" published in *International Journal on Computer Science and Engineering,* v.1 (2), p. 98-104, 2009.

[21] A. A. Pirzada, C. McDonald, "Establishing trust in pure ad-hoc networks," *Proc. 27th Australasian conference on Computer science, Dunedin, New Zealand*, v.56, p. 47 – 54, 2004.

[22] Wu, J. and Li, H. : A Dominating-Set-Based Routing Scheme in Ad Hoc Wire-less Networks, *Springer Science+Business Media B.V., Formerly Kluwer Academic Publishers,* B.V. ISSN: 1018-4864, 2001.