STABILIZING CRITERIA FOR MOBILE AD-HOC CLUSTERS

ADITI SINGH(09BCE003), NEELANJAN KAYAL(09BCE034), PRASHANT BHANSALI(09BCE039)

PROF PARVEEN SULTANA

SCHOOL OF COMPUTING SCIENCES AND ENGINEERING

VELLORE INSTITUTE OF TECHNOLOGY

Aim: to find an optimized clustering algorithm to divide the set of mobile nodes into a set of clusters based on a hierarchy of parameters and then selecting a cluster leader to communicate and route the data.

Abstract: A mobile ad-hoc network is a network wherein there is no fixed infrastructure and whose communication relies on their cooperation among the nodes to route the data using wireless multi-hop. Therefore, this kind of network is self-organizing and self-managing. In this paper, we propose an algorithm to stabilize the clusters based on a hierarchy of parameters such as transmitting power, connectivity, mobility and energy. For this the set of nodes are grouped together to form clusters based on their location as in in the distance of two hop nodes. The algorithm forms clusters and the cluster leaders are chosen. The communication between the sender and receiver is done using the cluster leaders acting as the routers.

Methodology: We are proposing an algorithm considering Stabilizing criteria for MANET.

1. Introduction

Wireless cellular systems have been in use since 1980s. Wireless systems operate with the aid of a centralized supporting structure such as an access point. These access points assist the users to keep connected with the wireless system, when they roam from one place to the other.

The presence of a fixed supporting structure limits the adaptability of wireless systems and the technology cannot work effectively in places where there is no fixed infrastructure. There will be a requirement of easy and quick deployment of wireless networks. Thus a need arises for the use of an ad-hoc network.

Recent advancements such as Bluetooth introduced a system known as mobile ad-hoc networks. Mobile ad-hoc networks or "short live" networks operate in the absence of fixed infrastructure. They offer quick and easy network deployment in situations where it is not possible otherwise. Ad-hoc is a Latin word, which means "for this or for this only." Mobile ad-hoc network is an autonomous system of mobile nodes connected by wireless links; each node operates as an end system and a router for all other nodes in the network.

Nodes in mobile ad-hoc network are free to move and organize themselves in an arbitrary fashion. Each user is free to roam about while communication with others. The path between each pair of the users may have multiple links and the radio between them can be heterogeneous. This allows an association of various links to be a part of the same network.

The popular IEEE 802.11 "WI-FI" protocol is capable of providing ad-hoc network facilities at low level, when no access point is available. However in this case, the nodes are limited to send and receive information but do not route anything across the network. Mobile ad-hoc networks can operate in a standalone fashion or could possibly be connected to a larger network such as the Internet.

An ad hoc network is a multihop communication network, which supports users without any infrastructure. The main problem to be addressed in ad hoc networks is scalability problem (Jiangchuan *et al*., 2004). Clustering helps in solving this problem. Partitional clustering is non-hierarchical. Since a set of clusters is output, the clusters are generated based on the number of nodes. In this paper, based upon a hierarchy of parameters such as transmitting power, connectivity, mobility and interference, cluster leaders and hence clusters are formed by grouping nodes in a network. Each node is well aware of its properties and it broadcasts them to surrounding nodes to bring them under its clustering until it finds a node which is more superior to it and joins it. The source of a data knows the position of the destination. Then, the cluster leaders are identified by the source and destination. Through the optimal path, they send/receive the data through their cluster leaders.

2. LITERATURE REVIEW

Clustering: grouping done to a set of nodes on the basis of certain parameters, for example, relative distance of one node to the other, density of the node set distribution or the number of expected clusters in the final output. For these cases there are many different types of algorithms. Some of them are listed below.

Hierarchical algorithms: In these algorithms, there exists a hierarchy of clusters, that is, clusters within clusters. The main parent cluster consists of one or more children clusters, thus forming a relation wherein the child cluster nodes belong to the parent subset. These algorithms are either agglomerative or divisive. Agglomerative algorithms begin with each element as a separate cluster and join them to form consecutively larger clusters. Divisive algorithms begin with the entire set and proceed to split it into consecutively small clusters.

Partition algorithms: Here, all clusters are determined at once. There are various types of Partitional clustering algorithms such as minimum spanning tree, k-means clustering [1] (Tapas *et al*., 2002), nearest neighbour algorithm. In Minimum Spanning Tree (MST) algorithm, inconsistent edges are identified and removed. Then, a set of ordered pairs are formed as clusters. In k-means the set of clusters are selected at random initially. Then the items are moved between clusters till desired set is formed. High degree of similarity among items in a cluster is obtained. In nearest neighbour algorithm, items are iteratively merged into the existing clusters that are close to each other.

Clustering in Ad-hoc networks: An ad-hoc network is a temporary and highly dynamic association of nodes not depending upon any fixed support infrastructure. It is a wireless network used for communication. In mobile ad hoc networks, the location of the network nodes may change the topology swiftly. In topology maintenance, a quick change in topology leads to a rise in the overhead message; the clustering schemes for mobile ad hoc networks [2] (Dali and Chan, 2007). In a multihop packet switched network, nodes are required to route packets between the source and destination even if they are not directly connected. Ad hoc networking technology is one, which can be formed everywhere, regardless of the location. In an ad hoc network, users and computing devices will be able to connect conveniently and even transparently. In mobile ad hoc networks, nodes communicate without stationary infrastructure. Intermediate nodes act as routers, when messages are exchanged between nodes. As proposed by [3](Frank *et al*., 2010), node clustering is an important operation performed in graph analysis. As stated by Frank *et al*. (2010), the spectral techniques works better for larger graphs and that the sparse network model works well with smaller number of clusters. The algorithm [4] (Alneu and Nils, 2011) improves the k-nn classification accuracy.

Independent dominating sets as clusters: Insisting that the dominating set is also an independent set can produce a relatively small number of clusters of a given graph.

3. RELATED WORK

Many clustering methods for MANETs focusing on node mobility and power consumption have been proposed. Chatterjee et al. proposed a clustering algorithm for MANET by considering the network as a vertex weighted graph [6]. The algorithm chooses a node as a cluster head if the node is expected to from a cluster of proper size based on battery power and the movement distance of the node. Bazzal et al. proposed a method to reduce the number of clusters by selecting a node with more neighbors as a clusterhead [7]. Chinara et al. proposed a clustering algorithm to extend the life time of the network [8]. The algorithm keeps a node with large amount of battery as a clusterhead. Theoleyre et al. proposed a new criterion for cluster head selection that considers load balance and stability [9]. The method elects a node with the optimal degree (decided by the application), relatively small mobility, and sufficient energy as a cluster head. The above mentioned methods choose a cluster head based on movement distance of the node, local topology, and battery power, however, they did not consider a realistic mobility pattern such as group mobility. One way to obtain a stable cluster structure is to reflect the characteristics of the target mobility when a cluster head is selected. Cheng et al. proposed a method to optimize multiple criteria by using the multiobjective evolutionary algorithm (MOEA) [10]. The method also considers the stability of the cluster structure in group mobility. The algorithm first extracts stable topology according to the distances between neighbouring nodes, then runs the multi-metric clustering procedure based on MOEA on the extracted topology. The method offers the Pareto-optimal solutions so that it considers not only stability of the cluster but also other metrics. However, this method is a centralized algorithm. Some clustering algorithms calculate the position of nodes from the signal strength without using GPS. Basu et al. proposed MOBIC, that is based on the aggregate local mobility [11]. The aggregate local mobility of a node is calculated based on the ratio of received signal strength of two successive transmissions from each neighbouring node. When a node keeps relative position (i.e., neighbours), its aggregate local mobility is small. MOBIC chooses a node with smaller aggregate local mobility as a cluster head. Er et al. proposed MobDHop, using the standard deviation of relative mobility calculated from received signal strength. MobDHop considers a situation that nodes move in groups. Nodes that do a similar movement form one cluster. However, signal strength varies because of many environmental conditions. By using GPS, we can obtain a more stable cluster structure, and GPS chip has been miniaturized and implemented in many mobile terminals at a low price. Johnen et al. proposed a robust self-stabilizing clustering algorithm (RSCA) for a vertex weighted graph representing a topology of a MANET [3]. Robustness guarantees that every Ordinary node belongs to a cluster even in the presence of topology changes and node's weight changes. A clusterhead is selected based on nodes' weight: a node with larger weight is a better candidate of a clusterhead. Yamauchi et al. pointed out that in RSCA a node with large weight moving around causes frequent change of clusterheads, and proposed a new method to solve this problem [4]. They proposed a method assigns a larger weight to a node with fewer changes in its relative position. However, this method did not consider any realistic node mobility. In this paper, we propose a weight assignment method for stable cluster structure. The proposed method is applicable to any clustering algorithm that considers a MANET as a weighted graph such as [3]. In MANET, mobility models that abstract real users' mobility are often used [12]. Two types of mobility models exist:(1) the models where each node moves independently, and (2) the models where nodes form groups and move together. The latter model is generally called a *group mobility model*. The *Random Way Point (RWP)* model is well used as a mobility model where each node moves independently [13]. In RWP, each node moves independently to a randomly selected destination with a randomly selected velocity. The *Reference Point Group Mobility (RPGM)* model is well used as a group mobility model. In RPGM, each group has a virtual central position, and the virtual central position have a movement vector that decides the movement of the group. Each node in the mobility group follows the virtual central position with keeping a randomly selected distance.

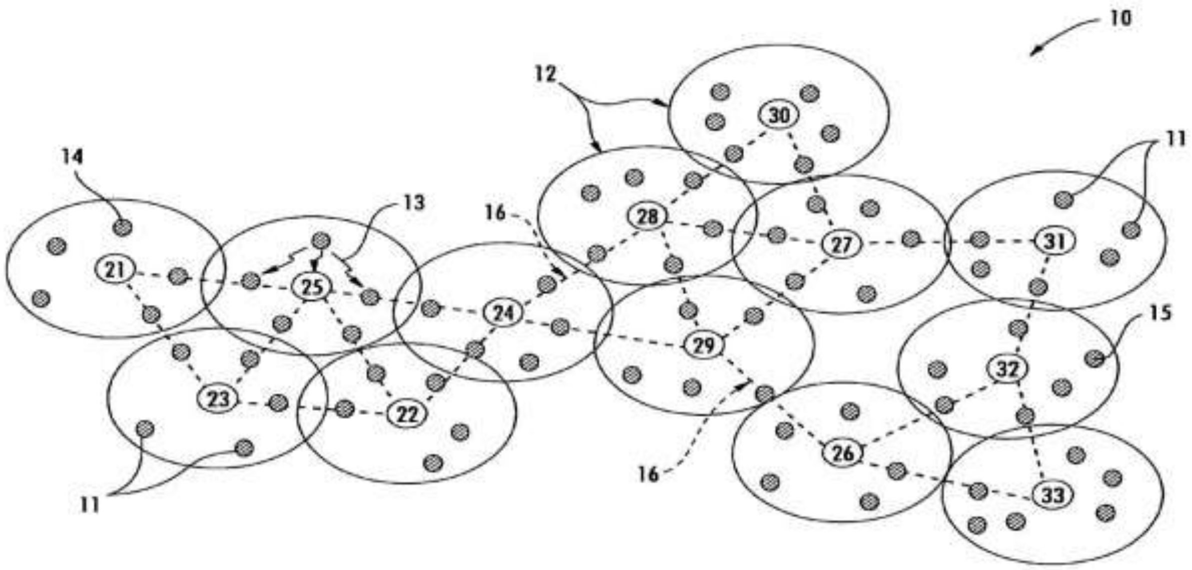


Fig1. Cluster formation with cluster leader managing the routing of data from source to destination.

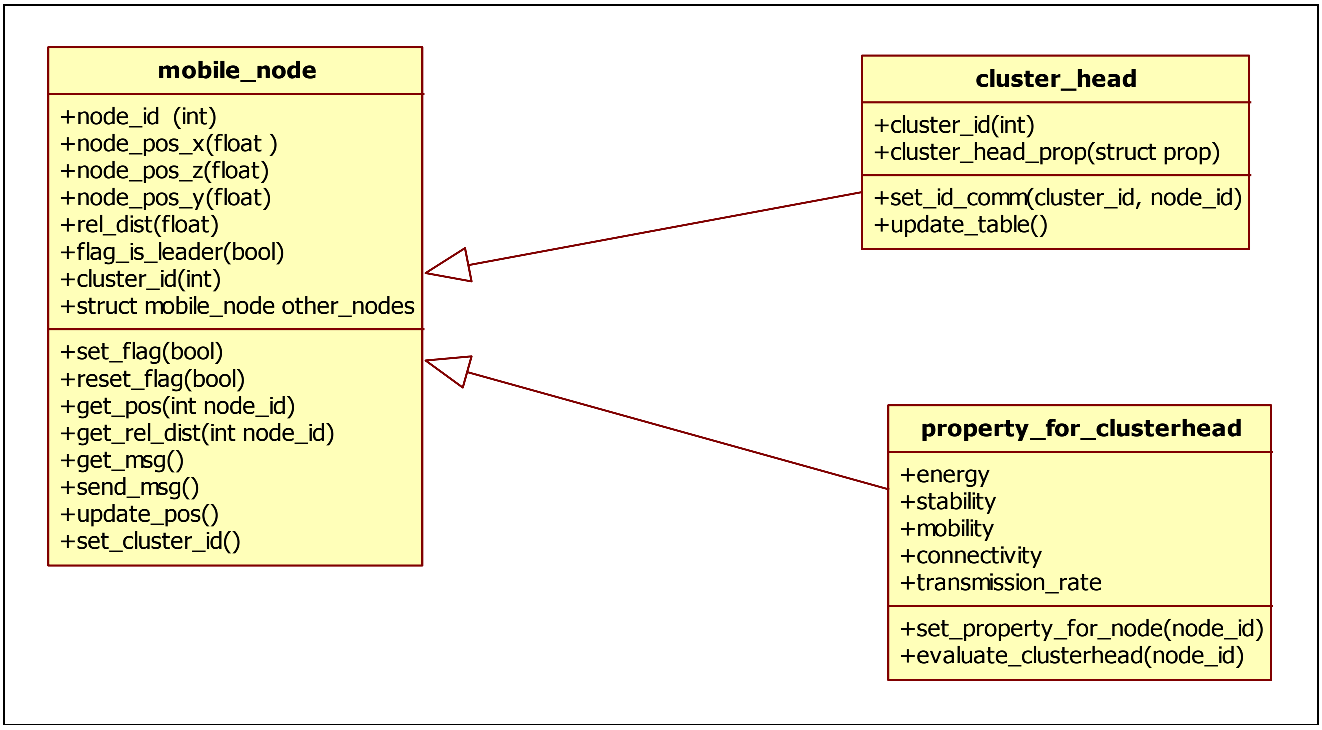


Fig 2: class diagram for a highly dynamic mobile ad- hoc network

4. DESIGN AND ANALYSIS:

A method is provided for sending data in a wireless ad-hoc network including a plurality of nodes grouped into clusters of nodes and a plurality of wireless links connecting the plurality of nodes, where each cluster has a designated cluster leader node. The method may include sending a cluster-level route request from a source node of a source cluster to a cluster leader node of the source cluster, and determining a cluster-level route between the source cluster and a destination cluster including a destination node responsive to the cluster-level route request and using a plurality of the cluster leader nodes. Further, a node-level route may be determined from the source node to the destination node along the cluster-level route, and each node along the node-level route may store an address of a next node along the node-level route. Additionally, the data may be transferred via the node-level route based upon the stored addresses.

4.1 DESCRIPTION OF CLASS DIAGRAM:

4.1.1 class mobile_node

4.1.1.1 Attributes:

node_id (int): saves the node id for the mobile node.

node_pos_x(float): has the x coordinate

node_pos_y(float): has the y coordinate

node_pos_z(float) : has the z coordinate

rel_dist(float): has the relative distance between the source and the node

flag_is_leader(bool): if clusterhead then 1 else 0

cluster_id(int): has the cluster id of the cluster it belongs to

struct mobile_node other_nodes: saves the node_id and the information in a table for scalability and robustness.

4.1.1.2 Methods:

set_flag(bool): set flag to 1 to denote if clusterhead

reset_flag(bool): set flag to 0 to denote not a clusterhead

get_pos(int node_id): calculate position of the node_id

get_rel_dist(int node_id): calculate the relative distance

get_msg(): get the msg as a router from the sender via a hop node

send_msg():send the msg as a router from the sender to destination via a hop node

update_pos(): update the position in table if nodes move out / cluster changes.
set_cluster_id(): calculate the cluster in which the node lies.

4.1.2 class clusterhead

4.1.2.1 Attributes:

cluster_id(int): stores the cluster id
cluster_head_prop(struct prop): stores the link to the properties which define it as the cluster leader

4.1.2.2 Methods:

cluster_head_prop(struct prop): calculate the properties of the clusterhead
update_table(): update table to update changes

4.1.3 class property_for_clusterhead

4.1.3.1 Attributes:

Energy: store the power and energy of the node
Stability: see if the node is stable enough
Mobility: see if the node is not dynamic enough
Connectivity: see that the connectivity of the node is high
Transmission_rate: see whether the transmission rate is good enough

4.1.3.2 Methods:

set_property_for_node(node_id): set all values for the properties defined above
evaluate_clusterhead(node_id): on the basis of the above properties check whether a node is qualified to become the cluster head or not.

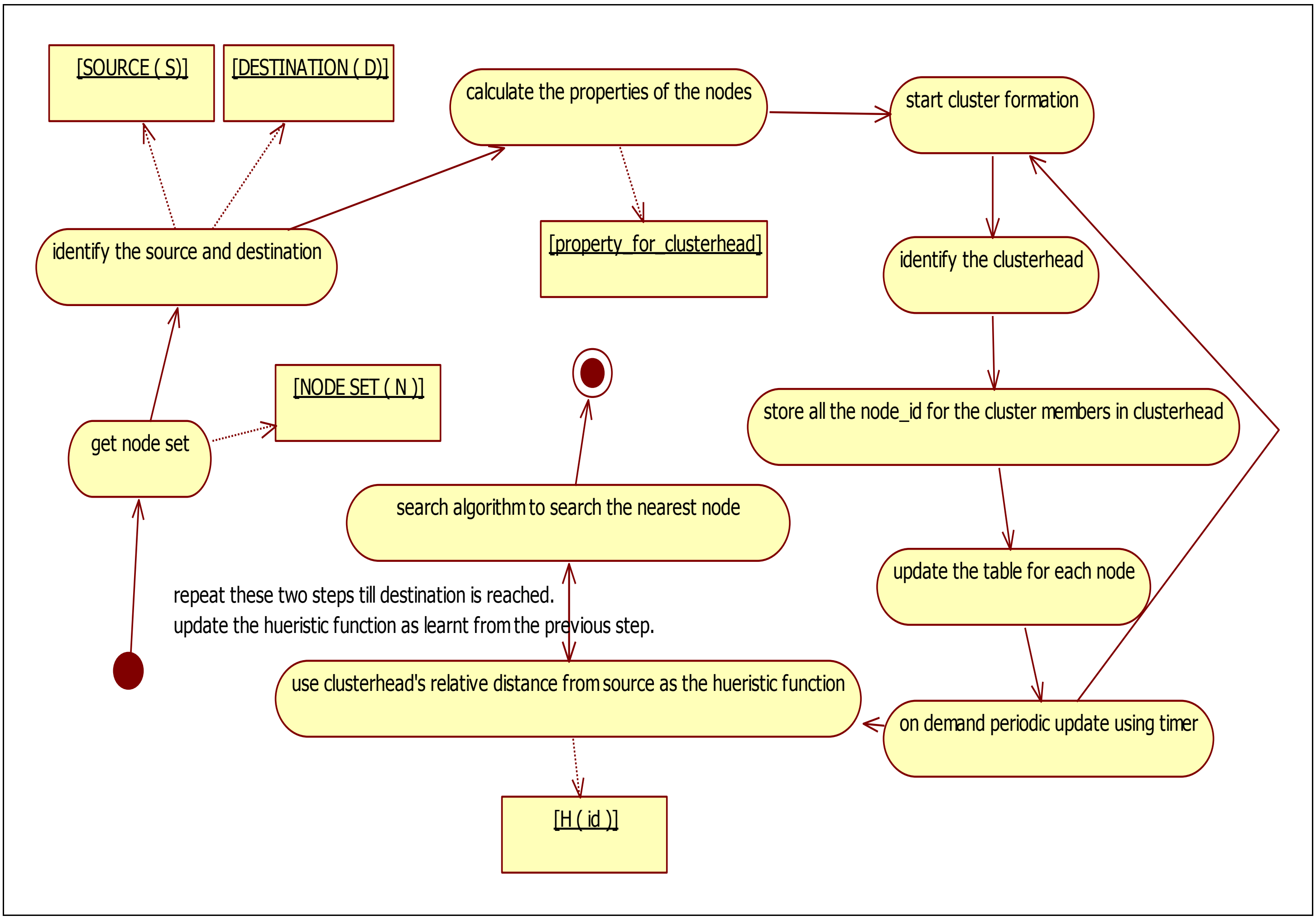


Fig3: the activity diagram.

5. IMPLEMENTATION:

5.1 Proposed algorithm

The basic ideas of proposed algorithm are focused on the quick adaptation with the changes of topology and reduce the re-election of cluster head. On the other hand, to make the best choice of the necessary factors at the time of election of cluster head. In this algorithm, we prefer to collect the metrics which contribute in the stability of cluster. For this, in the election of cluster head we privilege the node that has:

• The low mobility, because if the contrary, the node risk to lose its neighbours and joint another cluster.

• The best value of transmission range, because it affects several aspects namely: energy, connectivity, quality of signal. For this, we use our algorithm SEMC (Saving Energy and Maximizing Connectivity) that is developed in [16] to choose the best value of transmission range which saves the energy and keeps the connectivity.

• The great reserve of energy in order to avoid that the node becomes not operational after a short time of the selection.

• The higher connectivity in order to assure cluster head responsibilities.

5.2 Algorithm details

The proposed algorithm is composed of two phases. The first phase consists in the election of cluster head and the second consists in the formation of members of the cluster. The different steps of each phase are described as follows:

Phase 1: Election of the cluster head

In this stage we need the position of nodes, thus it is necessary that each node broadcasts its SUPERIOR PACKET to all its neighbours in the same transmission range. Each neighbour that received the broadcasted message compares it with its packet so as to decide the cluster head.

SUPERIOR PACKET

Representattive ID
Own ID
Mobility Processing power Transmission Power Connectivity Interference
TTL
Password SSID

Step 1: **Calculate the mobility, connectivity, transmission power and battery of the node.**

Step 2: **Hot-Spot is created by the first node**

Initially, one node generates a hot-spot.

It considers itself the leader of its cluster. It starts sending SUPERIOR PACKETS with TTL=2, immediately. It keeps broadcasting these packets every 1 seconds. When another node requests to join this ad-hoc network, it will receive SUPERIOR PACKETS from the leader. Now, on receiving such a packet, a HOLD_DOWN timer is started in this node. The HOLD_DOWN timer is of 120 seconds. During this time, the new node considers the current cluster-leader as its own leader too. But, once the HOLD_DOWN timer expires, an election for the position of the leader takes place

Step 3: **Comparison and re-election for a new leader**

For this election, the new node starts sending superior packets itself. So, when the other node receives these packets, it compares the values of the various clustering factors in this packet to those of its own.

The comparison is based on four fields in the packet: Mobility, Processing Power, Transmitting power and Connectivity (in that order).

When a node is comparing the values, if its own values are better than that of the sending node, it remains the leader. However, if its parameters are inferior, it elects the other node as its leader and consequently, stops sending its own SUPERIOR PACKETS. Instead, the non-leader(s) will just forward the received SUPERIOR Packet with the TTL value decremented by 1. (This node will, from now on, broadcast only these packets, until it becomes a leader itself). Thus a cluster is formed with a leader elected.

Step 4: **More nodes join the cluster.**

So, when more nodes request to join this MANET, they broadcast their own SUPERIOR PACKETS after their HOLD_DOWN timers expire.

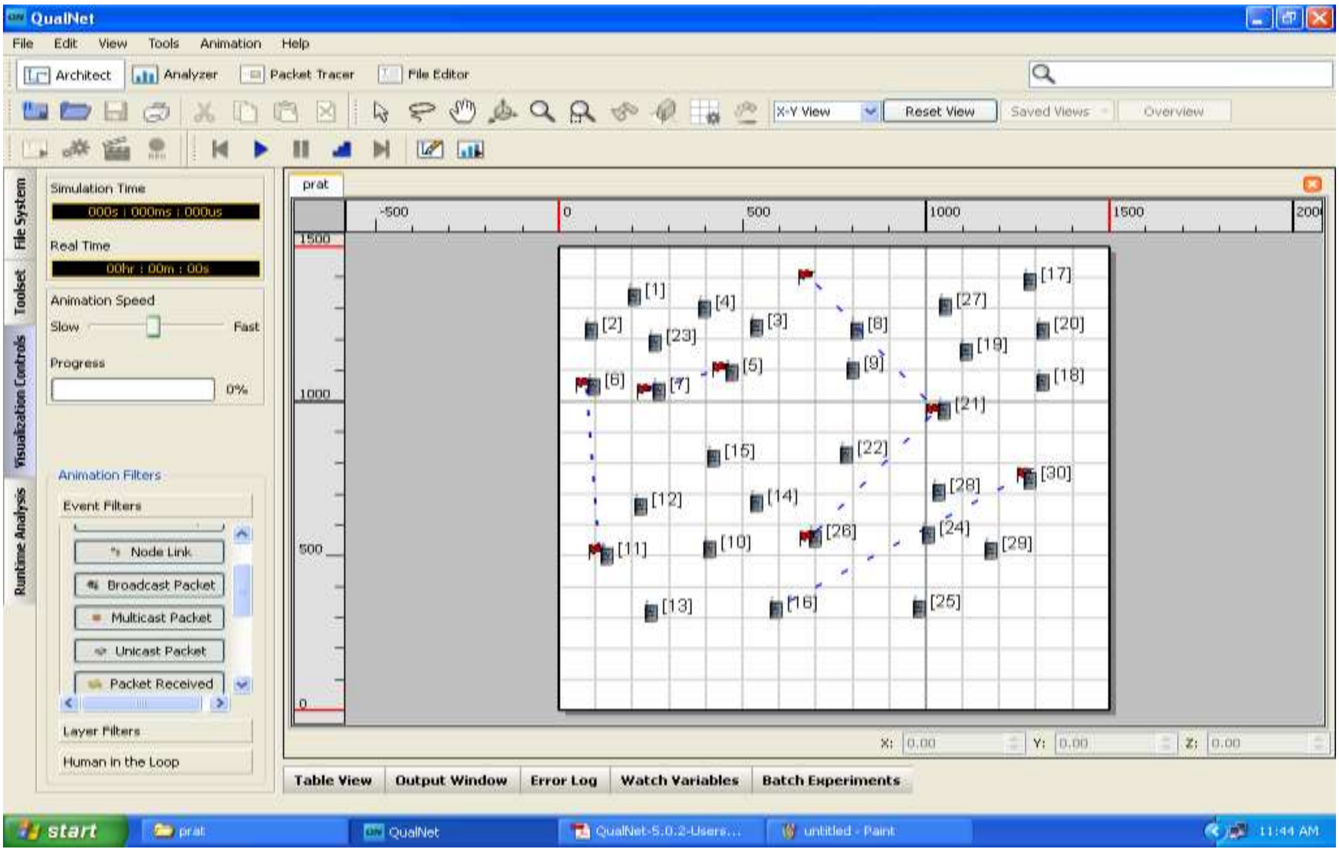
Phase 2: When a node goes down

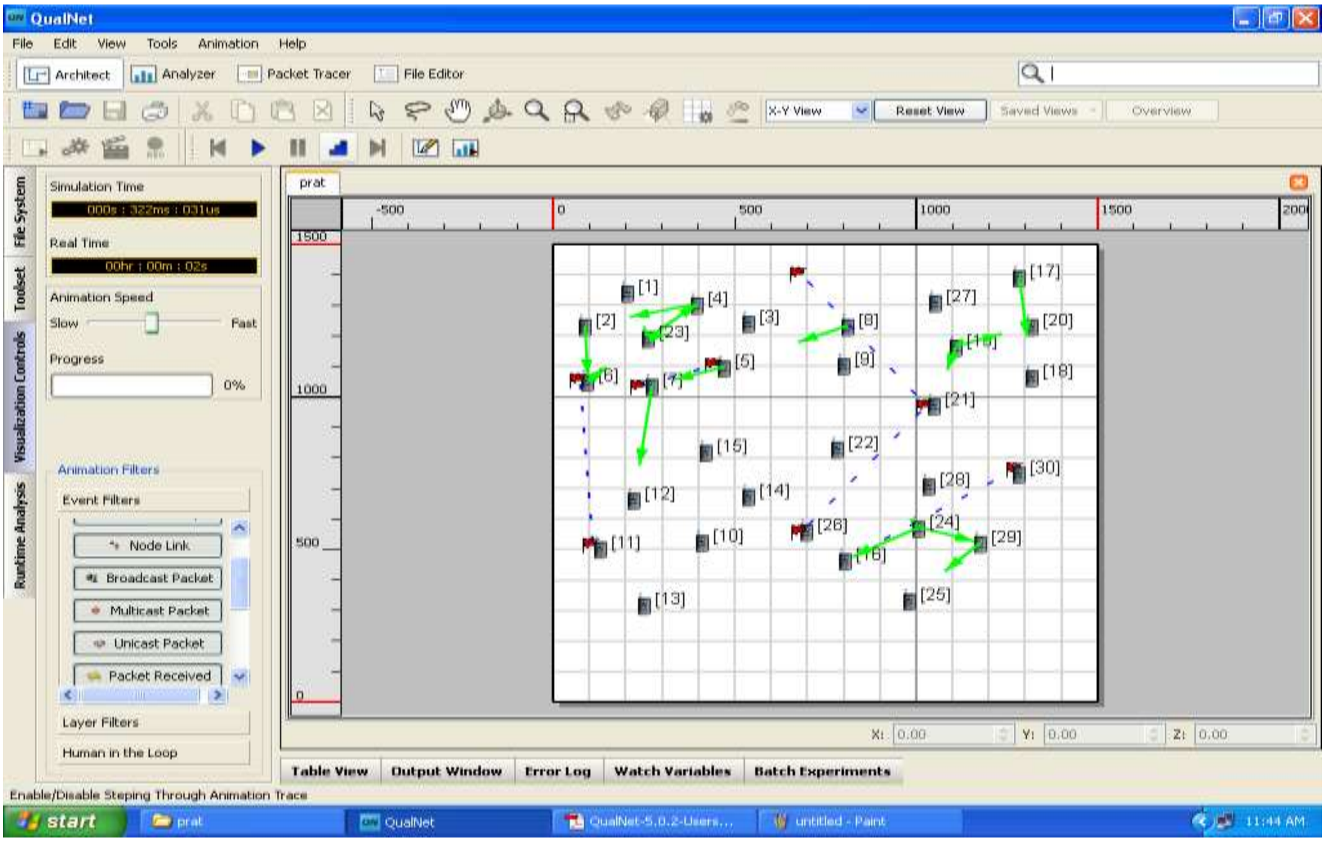
If it is a leader node, it will stop broadcasting its SUPERIOR Packets (obviously). So, when the other nodes fail to receive their leader's packets 3 consecutive times, they will try to receive SUPERIOR Packets from any other leader, failing which, a re-election is initiated, wherein every node again starts sending their SUPERIOR PACKETS. However, if some of them receive SUPERIOR Packets from other leader(s), they will accept them and (re)start their HOLD_DOWN timer and consider themselves as part of these new clusters (until their

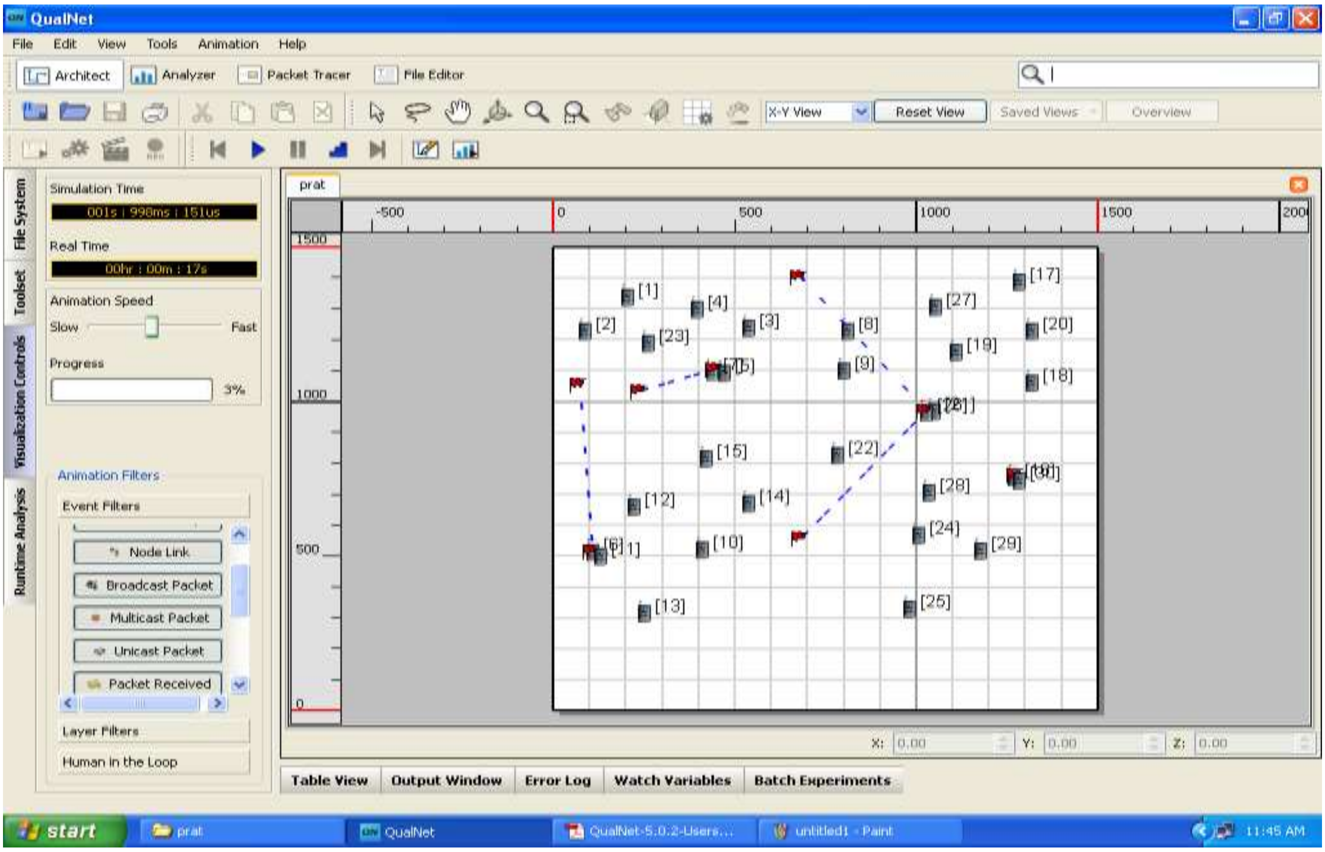
HOLD_DOWN timers expire). The other non-receiving nodes will start their own election.
If it is a non-leader node, no changes to the topology are necessary.

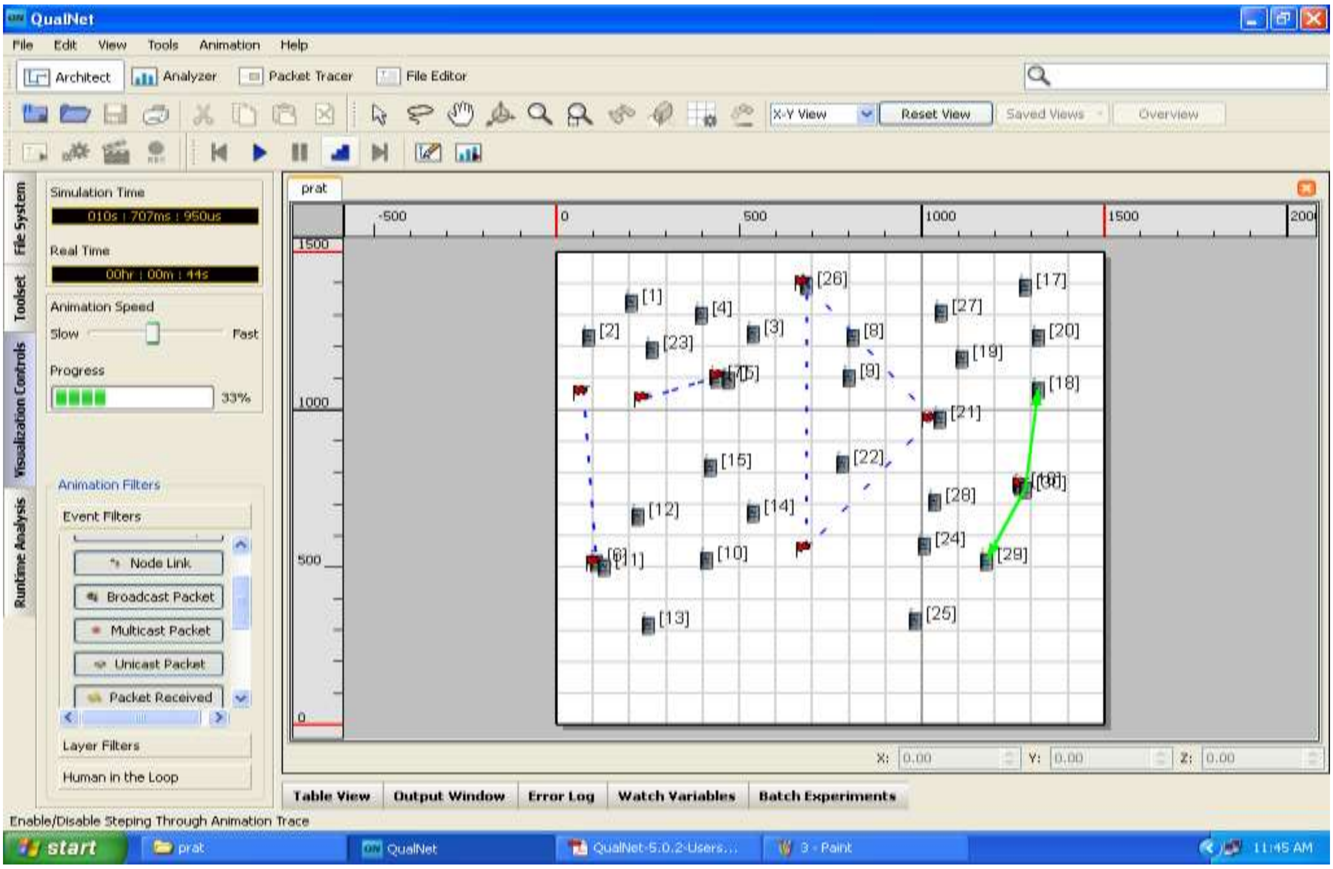
And, when a node leaves the vicinity of one cluster
If it is a leader node, it will continue to broadcast its SUPERIOR Packets in the new cluster zone. But, the other nodes in that cluster will not accept it as their leader (or, initiate a re-election, for that matter) by starting their HOLD_DOWN timers. Thus, the leader will act like a regular node in the new cluster. If it is a regular node, it will wait to receive SUPERIOR Packets from some leader, failing which, it will initiate an election by commencing its own SUPERIOR Packet broadcasts.

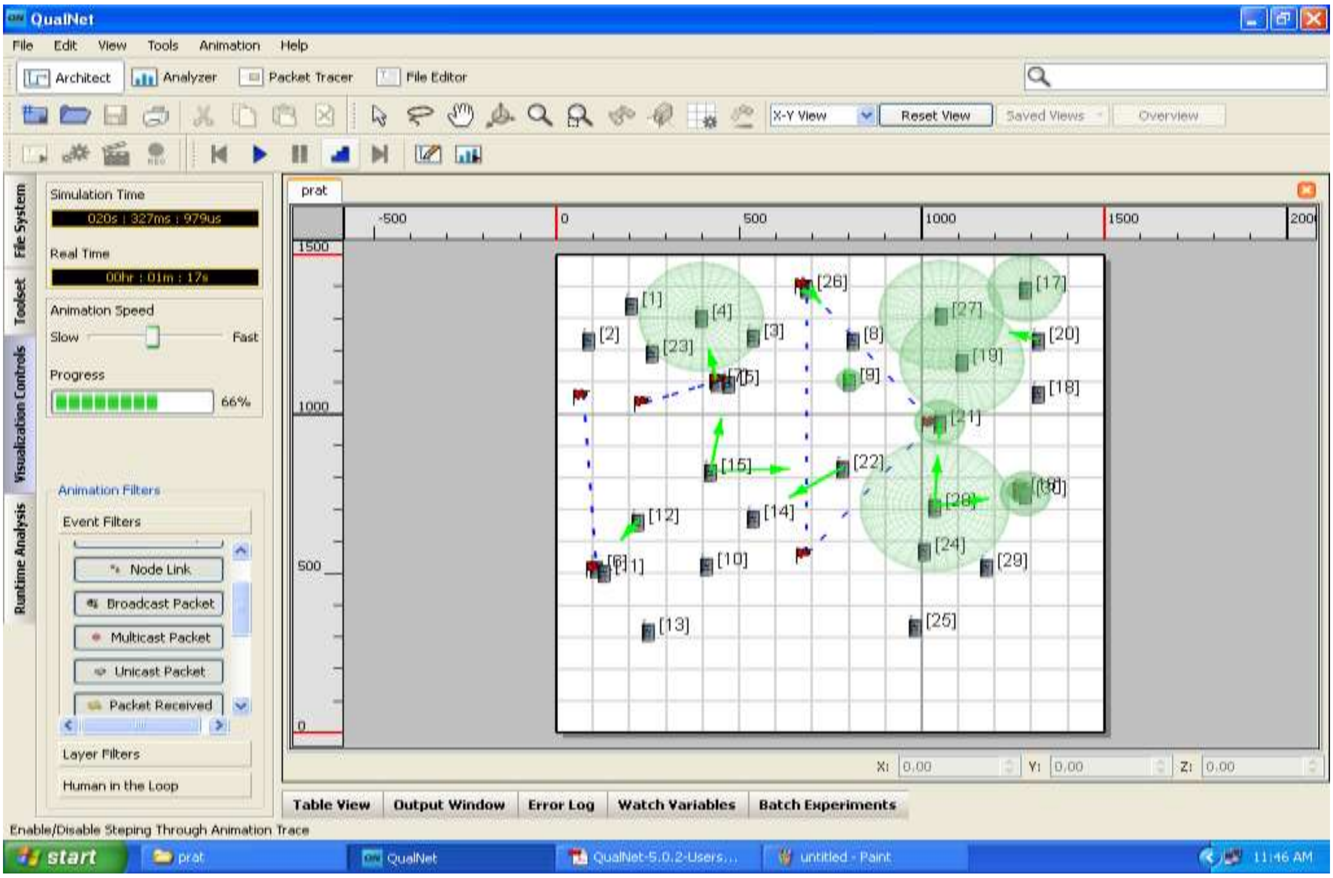
5. QUALNET SNAPSHOT




QualNet
File Edit View Tools Animation Help
Architect
Analyzer
Packet Tracer
File Editor
X-Y View
Reset View
Saved Views
Overview
Simulation Time
000s : 322ms : 031us
Real Time
00hr : 00m : 02s
Animation Speed
Slow
Fast
Progress
0%
Animation Filters
Event Filters
Node Link
Broadcast Packet
Multicast Packet
Unicast Packet
Packet Received
Layer Filters
Human in the Loop
File System
Toolset
Visualization Controls
Runtime Analysis
prat
Table View
Output Window
Error Log
Watch Variables
Batch Experiments
Enable/Disable Steping Through Animation Trace
start
11:44 AM


QualNet
File Edit View Tools Animation Help
Architect
Analyzer
Packet Tracer
File Editor
X-Y View
Reset View
Saved Views
Overview
Simulation Time
001s : 998ms : 151us
Real Time
00hr : 00m : 17s
Animation Speed
Slow
Fast
Progress
3%
Animation Filters
Event Filters
Node Link
Broadcast Packet
Multicast Packet
Unicast Packet
Packet Received
Layer Filters
Human in the Loop
File System
Toolset
Visualization Controls
Runtime Analysis
prat
Table View
Output Window
Error Log
Watch Variables
Batch Experiments
start
11:45 AM


QualNet
File Edit View Tools Animation Help
Architect Analyzer Packet Tracer File Editor
X-Y View Reset View Saved Views Overview
File System Toolset Visualization Controls Runtime Analysis
Simulation Time
010s : 707ms : 950us
Real Time
00hr : 00m : 44s
Animation Speed
Slow Fast
Progress
33%
Animation Filters
Event Filters
Node Link
Broadcast Packet
Multicast Packet
Unicast Packet
Packet Received
Layer Filters
Human in the Loop
prat
-500 0 500 1000 1500
1500 1000 500 0
[1] [2] [3] [4] [5] [8] [9] [10] [12] [13] [14] [15] [17] [18] [19] [20] [21] [22] [23] [24] [25] [26] [27] [28] [29]
X: 0.00 Y: 0.00 Z: 0.00
Table View Output Window Error Log Watch Variables Batch Experiments
Enable/Disable Steping Through Animation Trace
start prat QualNet QualNet-5.0.2-Users... 3 - Paint 11:45 AM


QualNet
File Edit View Tools Animation Help
Architect Analyzer Packet Tracer File Editor
X-Y View Reset View Saved Views Overview
File System Toolset Visualization Controls Runtime Analysis
Simulation Time
020s : 327ms : 979us
Real Time
00hr : 01m : 17s
Animation Speed
Slow Fast
Progress
66%
Animation Filters
Event Filters
Node Link
Broadcast Packet
Multicast Packet
Unicast Packet
Packet Received
Layer Filters
Human in the Loop
prat
-500 0 500 1000 1500
1500 1000 500 0
[1] [2] [3] [4] [5] [8] [9] [10] [12] [13] [14] [15] [17] [18] [19] [20] [21] [22] [23] [24] [25] [26] [27] [28] [29]
X: 0.00 Y: 0.00 Z: 0.00
Table View Output Window Error Log Watch Variables Batch Experiments
Enable/Disable Steping Through Animation Trace
start prat QualNet QualNet-5.0.2-Users... untitled - Paint 11:46 AM

6. REFERENCES

[1] Tapas, K., M.M. David, S.N. Nathan, D.P. Christine, S. Ruth and Y.W. Angela, 2002. An efficient k- Means clustering algorithm: Analysis and implementation. IEEE Trans. Pattern Analy. Mach. Intell., 24(7): 881-892.

[2] Dali, W. and H.A. Chan, 2007. Clustering Ad Hoc Networks: Schemes and Classifications, Annual IEEE Communications Society on Sensor and Ad Hoc Communications and Networks, pp: 920-926.

[3] Frank, L., B. Ramnath and W.C. Willia, 2010. Node Clustering in Graphs: An Empirical Study, NIPS, Vancouver, BC, Canada.

[4] Alneu de Andrade Lopes and Nils Murrugarra-Llerena, 2011. An Adaptive Graph-Based K-Nearest Neighbor, ICMC, USP.

[5] Baker, D. J. and A., Ephremides, 1981, The architectural organization of a mobile radio net work via a distributed algorithm. IEEE Trans. Commun. COM, 29: 1694-1701.

[6] M. Chatterjee, S. Das, and D. Turgut. Wca: A weighted clustering algorithm for mobile ad hoc networks. *Cluster Computing Journal*, 5(2):193–204, 2002.

[7] G. Chen and I. Stojmenovic. Clustering and routing in mobile wireless networks. Technical Report TR-99-05, Technical Report, 1999.

[8] A. Ephremides, J. Wieselthier, and D. J. Baker. A design concept for reliable mobile radio networks with frequency hopping signaling. In *Proc. of the IEEE*, pages 56–73, 1987.

[9] B. et S. Papavassiliou. A mobility-based clustering approach
to support mobility management and multicast routing in mobile ad-hoc wireless networks. *International Journal of Network Management*, 11:387–395, 2001.

[10] W. J. Hsu, K. Merchant, H. W. Shu, C. Hsu, and A. Helmy. Weighted waypoint mobility model and its impact on ad hoc networks. In *Proc. of the Mobile Computing and Communications Review*, 2005.

[11] K. I. Kim and S. H. Kim. A novel factor for robust clustering in mobile ad hoc networks. *IEICE Transactions on Communication*, E89-B(4), 2005.

[12] G. Lin, G. Noubir, and R. Rajaraman. Mobility model for ad hoc network simulation. In *Proc. of IEEE Infocom*, 2004.

[13] W. Navidi and T. Camp. Stationary distributions for the random waypoint mobility model. *IEEE Transactions on Mobile Computing*, 3(1), 2004.

[14] A. Parekh. Selecting routers in ad hoc wireless networks. In *Proc. of the IEEE International Telecommunications Symposium*, 1994.

[15] Y. Shang and S. Cheng. A stable clustering formation in mobile ad hoc network. In *Proc. of Wireless Communications, Networking and Mobile Computing*, 2005.

[16] F. D. Tolba, D. Magoni, and P. Lorenz. Energy saving and connectivity tradeoff by adaptive transmission range in 802.11g manets. In *Proc. of the IEEE International Conference on Wireless and Mobile Communication (ICWMC)*, Bucharest, Romania, 2006.

[17] Junpei Kuroiwa, Yukiko Yamauchi, Weihua Sun and Minoru Ito. A Self-stabilizing Algorithm for Stable clustering in Mobile Ad-hoc Networks, 2011