# LOAD BALANCING IN WIRELESS MOBILE AD HOC NETWORKS

*A thesis submitted in partial fulfillment of the requirement for the degree of*

Bachelor of Technology
In
Computer Science and Engineering

**Arnab Maji**
10606016

Under the Guidance of
Prof. Suchismita Chinara

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

**National Institute of Technology, Rourkela**

# Certificate

This is to certify that the thesis titled, "**Load Balancing in Wireless Mobile Ad Hoc Network**", submitted by Arnab Maji in the partial fulfillment of the requirement for the award of the Bachelor of Technology in Computer Science and Engineering at National Institute of Technology, Rourkela is an authentic work carried out by him under my supervision and guidance.

To the best of my knowledge, the matter embodied in this thesis has not been submitted to any other university/institute for the award of any degree or diploma.

Date: May 2010

Prof. Suchismaita Chinara
Dept. of Computer Science and Engineering
National Institute of Technology, Rourkela

# Acknowledgement

I am sincerely thankful to Prof. Suchismita Chinara, Dept. of Computer Science and Engineering, NIT Rourkela for giving us the opportunity to work under her and lending every support possible at every stage of this project. The level of flexibility offered by her in implementing this project is highly appreciable.

I would also like to convey our sincerest gratitude and indebtedness to our entire faculty members and staff of the Dept. of Computer Science and Engineering, NIT Rourkela, who bestowed their efforts and guidance at appropriate times without which it would have been very difficult on our part to finish the project work.

# Abstract

Ad hoc networks consist of a set of homogeneous nodes (computers or embedded devices) that move in an independent fashion and communicate with the other node in the topology over a wireless channel. Such networks are logically realized as a set of clusters by grouping together nodes which are in close proximity with one another or through another wireless node. Clusters are formed by clubbing together nodes along the wireless links. *Cluster Heads* are the nodes which communicate with the other nodes that it can cover under its communication range. *Cluster Heads* form a virtual backbone and may be used to route packets for nodes in their cluster. Nodes, being in an Ad Hoc network, are presumed to have a non-deterministic mobility pattern. Different heuristics employ different policies to elect *Cluster Heads*. Many of these policies are biased in favor of some nodes. As a result, these nodes shoulder greater responsibility which may deplete their energy faster due higher number of communication made, causing them to drop out of the network. Therefore, there is a need for load-balancing among *Cluster Heads* to allow all nodes the opportunity to serve as a *Cluster Head*. I propose a few enhancements to existing algorithms to remove the unbalanced distribution of nodes under the *Cluster Heads* and increase the active life of a node in a network.

# ABBREVIATIONS

| | |
|---|---|
| ARPANET | Advanced Research Project Agency Network |
| DARPA | Defense Advanced Research Projects Agency |
| HC | Highest Connectivity |
| AP | Access Points |
| IP | Internet protocol |
| LID | Lowest ID |
| WPAN | Wireless Personal Area Network |
| WLAN | Wireless Local Area Network |
| WMAN | Wireless Metropolitan Area Network |
| WWAN | Wireless Wide Area Network |
| WEP | Wired Equivalent Privacy |
| WPA | Wi-Fi Protected Access |
| PRNET | Packet Radio Networks |
| MANET | Mobile Ad Hoc Networks |

# LIST OF FIGURES

# TABLE OF CONTENTS

**1. Introduction........................................................................................................8**

1.1 Ad Hoc Networks.........................................................................................................8

**2. Overview................................................................................................................10**

2.1 Motivation.................................................................................................................10

2.2 Organization of Thesis...............................................................................................10

**3. Literature Review....................................................................................................12**

3.1 Wireless Network.......................................................................................................12

3.2 Ad Hoc Networks.........................................................................................................14

3.3 MANET.......................................................................................................................16

3.4 Clusters......................................................................................................................16

3.5 Multi – Clusters Architecture.......................................................................................20

**4. Clustering Algorithms..............................................................................................21**

4.1 Related Work.............................................................................................................21

4.2 Parameter Identification.............................................................................................21

4.3 HC Algorithm.............................................................................................................22

4.4 LID Algorithm.............................................................................................................23

**5. Proposed Enhancement...........................................................................................25**

5.1 Proposed Enhancement..............................................................................................25

**6. Simulation and Results............................................................................................31**

6.1 Simulation Bed...........................................................................................................31

6.2 Results.......................................................................................................................32

**7. Conclusion..............................................................................................................33**

# CHAPTER 1
# INTRODUCTION

## 1.1 Ad Hoc Network

The Ad Hoc network is defined by the mobile nature of the nodes and the removal of the requirement for an infrastructure based network i.e. the use of routers and gateways. Ad Hoc networks generally work in clusters i.e. the grouping of wireless mobile devices (computers or embedded devices which is based on efficient communication between all the nodes). The infrastructure is auto generated by converting the all ready existing nodes into routers, repeaters or gateways.

A general Ad Hoc network will have the following features

**Mobility**: - The nodes can reposition themselves in matter of seconds, making the mobility pattern of the nodes non deterministic. This mobility pattern had a major effect on the formation of clusters within the network. This mobility and wireless nature is one of the major features of the Ad Hoc networks and helps it to be deployed in any kind of terrain.

**Multi hop Network**: Since the nodes work as group, a multi hop network is created so that even if a node is not in direct contact with the cluster head it can still get the information via the intermediate nodes by forwarding the same data. This multi hop networks is generated by the conversion of a normal node to a router or gateway.

**Multiple roles for a node**: The Ad Hoc networks should be able to organize itself by generating routers, gateways etc. to maintain communication with all the other nodes. This is done by converting a normal node to a router or a gateway.

**Energy Constraints**: In an Ad Hoc network the nodes are mobile and communicate over a wireless channel. Being mobile the power is used from a battery and the size needs to be kept at a minimum. Hence there is a need to manage the battery power consumption, so that they do not drop out of the network prematurely due to low power.

Out of these the greatest challenge for an Ad Hoc network is resource management and that too particularly the battery life which, later in the thesis is referred to as network lifetime.

The lifetime of an Ad Hoc network can be defined as the moment when the network starts working till one of the node drops out of the network because the battery was depleted. When clusters are formed in the network, one of the nodes becomes the cluster head (there could also be multiple cluster head depending on the selection parameters), which shoulders the responsibility of maintaining communication with all the other nodes. This responsibility of maintaining the communication links depletes the battery very fast as this node has to make more number of communications as compared to a normal node and the node drops out of the network. This reduces the network lifetime. In this thesis we discuss different clustering algorithms and propose an enhancement for efficient distribution of nodes under one cluster head.

# CHAPTER 2
# OVERVIEW

## 2.1 Motivation

Balanced clustering is the key to increasing the network lifetime. Also the Cluster Head consumes the maximum of its battery as compared to the rest of the nodes. Hence, if the number of nodes under one Cluster Head is more as compared to the rest of the Cluster Heads, then this node will prematurely drop out of the network. This dropping out of the cluster head drastically reduces the network life time. Hence the energy consumed in communicating with the different nodes in the networks, formation of the cluster, checking for living nodes etc. must be kept at a minimum. One way of doing this is to optimize the number of communications made by the Cluster Head. This can be done by balancing the number of nodes under all the Cluster Head i.e. making each Cluster Head have almost the same number of nodes under every Cluster Head.

## 2.2 Organization of Thesis

This thesis is organized into seven chapters where we start by giving an introduction to the thesis.

In the second chapter we discuss the motivation for the research work i.e. why we are doing this research.

In the third chapter we have done the literature review of the Ad Hoc networks and its components. Here we define all the terms related to Ad Hoc networks and wireless networks like clustering, MANET, Ad Hoc networks.

In the fourth chapter we have done a review of the clustering algorithms. These clustering algorithms are our base algorithm. Here we first identify the parameters that are important for implementing these algorithms along with the implementation of our proposed enhancements and then discuss about the HC algorithm, LID algorithm and a location unaware distributed clustering algorithm by using fuzzy logic.

In the fifth chapter we discuss the enhancement that we have suggested. This enchament helps the existing clustering algorithm to work in bit efficient way.

In the sixth chapter we implement the algorithm and discuss the results obtained. The results generated are then compared with the existing algorithms.

Finally in the seventh chapter we conclude the whole thesis.

The enhancements made with the base algorithms implements the balanced distribution of the nodes under all the ClusterHeads allowing each ClusterHead to have almost the same load over it. There by increasing the network lifetime.

# CHAPTER 3
# LITERATURE REVIEW

## 3.1 Wireless Network

**Wireless network** [5] [10] [11] [12] [13] is any type of computer network that is not connected by wires, and is generally associated with a telecommunications network where the connection between the nodes are made without the using any wires.[1] Wireless telecommunications networks generally use some type of electromagnetic waves (such as radio waves or microwaves) for the transmission of data or communication.[2]

**Types of Wireless Networks:**

- **Wireless PAN -** Wireless Personal Area Networks (WPANs) interconnect devices within a small area (ranging in meters). For example, by using Bluetooth we create WPAN for interconnecting a headset to a cell phone.

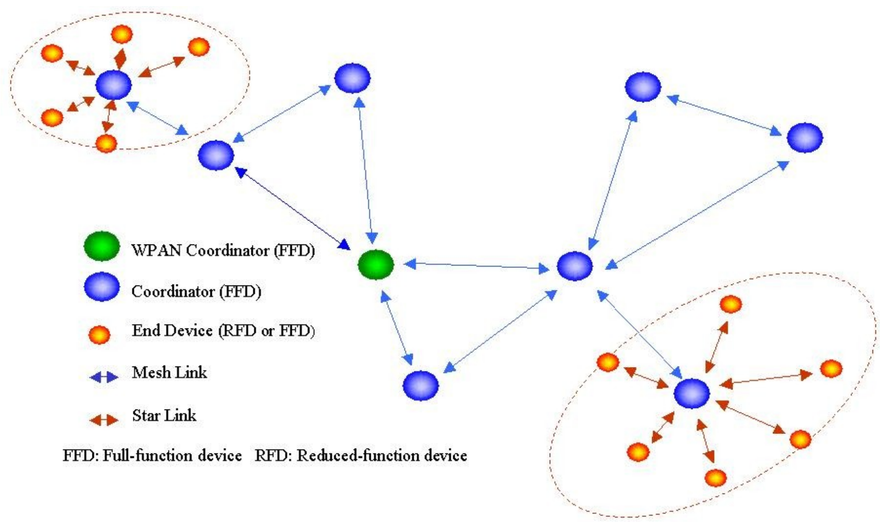


Fig 1: Typical WPAN

- **Wireless LAN** – Wireless LAN [5] is represented as a Wi-Fi network or a Fixed Wireless Data Communication. Fixed Wireless Data implements point to point links between computers(can be two independent networks also) at two different locations, by using dedicated microwave signals or laser beams over the line of sight paths. It is often used to connect two networks existing in two or more adjacent locations.

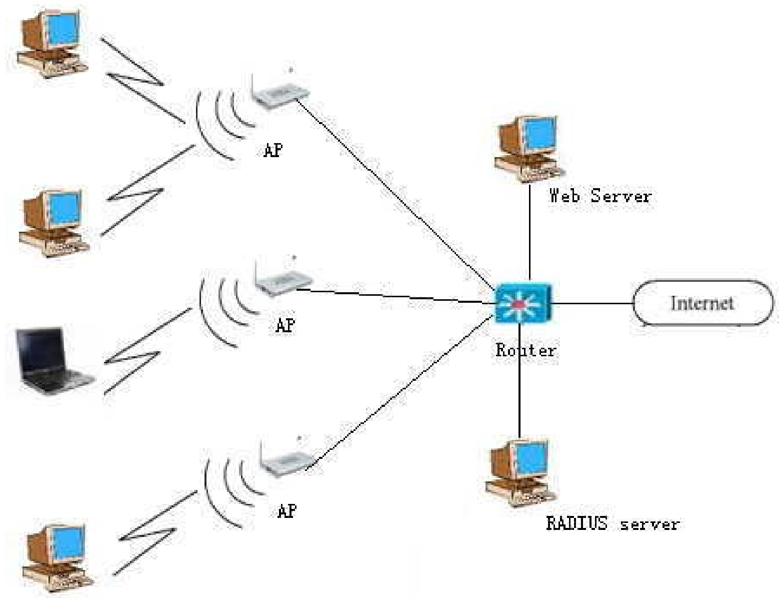


Fig 2: WLAN

- **Wireless MAN -** Wireless Metropolitan Area Networks [5] connects multiple Wireless LANs. WLAN is also known as WiMAX and is covered in IEEE 802.16d/802.16e.

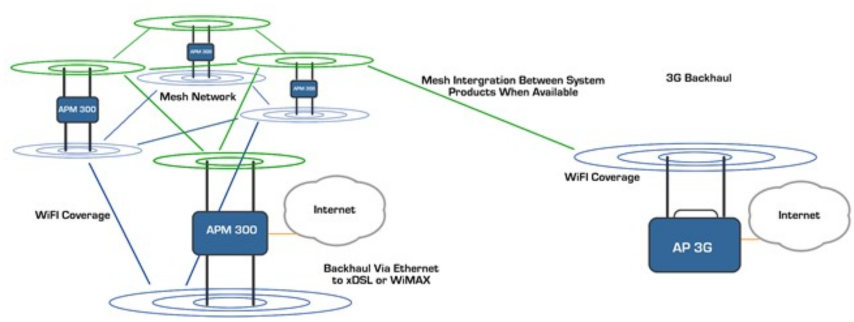


Fig 3: WMAN

- **Wireless WAN -** Wireless Wide Area Networks [5] are wireless networks that cover large outdoor areas. They are deployed at the frequency of 2.4 GHz.

## 3.2 Ad Hoc Networks

A **wireless ad hoc network**[13][14][15][16][17] is a decentralized wireless network where the network does not depend on a preexisting infrastructure, such as routers in wired networks or access points (AP) in managed (infrastructure) wireless networks. Instead, each node participates in routing by forwarding data to the other nodes, and so the determination of which nodes forward data is made dynamically i.e. the normal nodes are converted to a routers and gateways.

The earliest wireless ad hoc networks were the "packet radio" networks (PRNETs) from the 1970s, after the ALOHA net project sponsored by DARPA.

This decentralized nature of the Ad Hoc networks make them suitable at places where a single central node acting as a base station does not work efficiently or the terrain is not suitable to deploy stationary nodes which are connected via wired links. This makes Ad Hoc networks very useful in emergency condition

### Types of Ad Hoc Network:

- **MANET -** A mobile ad hoc network (MANET) [18], sometimes also known as a mobile mesh network, is a self-configuring network of mobile devices connected by wireless links. Each device in a MANET is free to move independently in any direction, and will therefore change its links to other devices frequently. Each must forward traffic unrelated to its own use, and therefore be a router.

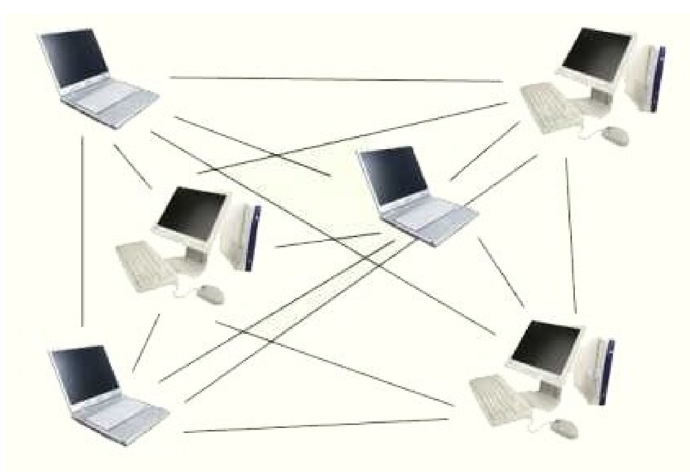

Fig 4: Typical MANET

- **Wireless Mesh Network** - A wireless mesh network (WMN) [26] is a communications network made up of radio nodes organized in a mesh topology.

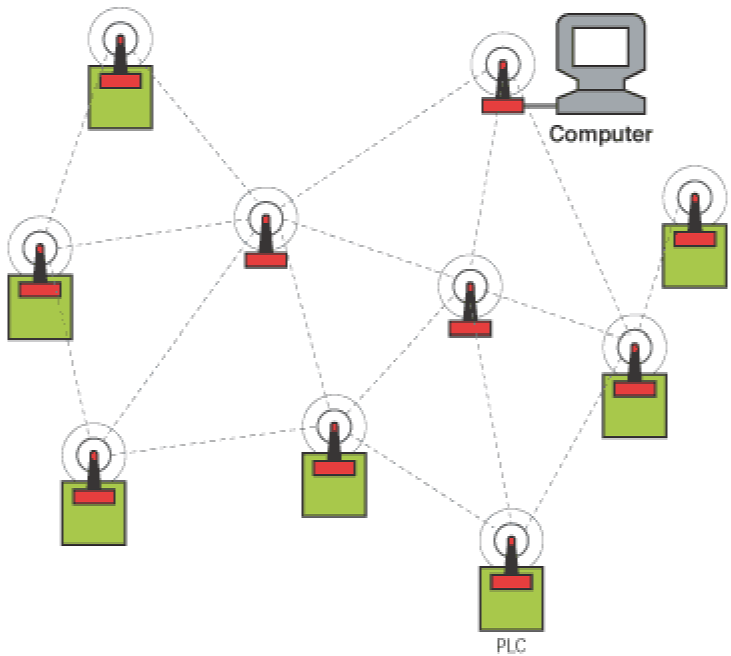


Fig 5: Typical WMN

- **Wireless Sensor Network** - A wireless sensor network (WSN) is a wireless network consisting of spatially distributed autonomous devices using sensors to cooperatively monitor physical or environmental conditions, such as temperature, sound, vibration, pressure, motion or pollutants, at different locations.

# 3.3 MANET

MANET [22] also known as mobile mesh network is a self configuring collection of mobile devices connects via wireless links.

Each device in MANET is connected to the other via wireless links, but since the network is an Ad Hoc network, the node can move in any direction without any constraints. Hence the link between two or more devices changes dynamically. This is the primary challenge of every MANET i.e. to maintain proper communication links with other nodes.

**Types of Mobile Ad Hoc Network:**

- **VANET -** A vehicular ad hoc network (MANET) is used to communicate between vehicles and/or road side equipments.
- **Intelligent VANET** - A vehicular ad hoc network (InVANET) is used to communicate between vehicles and/or road side equipments and behave intelligently incase of collision, minor snags etc.
- **Internet Based MANET** - A internet based MANET (iMANET) is a wireless network consisting of spatially distributed autonomous devices connected via wireless links with the nodes and to a fixed internet gateway.

## 3.4 Clusters

A cluster [24] is a group of linked devices (computer or embedded devices), working together closely so that they form a single node virtually. The components of a cluster are generally, but not always, connected to each other through wireless or wired. Clusters are usually deployed to improve performance and/or availability over that of a single computer, while typically being much more cost-effective than single computers of comparable speed or availability. They can also be defined as loosely coupled computers working together.

**Types of Clusters:**

- **High-availability clusters -** Also known as Failover Clusters [24], are implemented primarily for the purpose of improving the availability of services which the cluster provides. They operate by having redundant nodes, which are then used to provide service when system components fail.

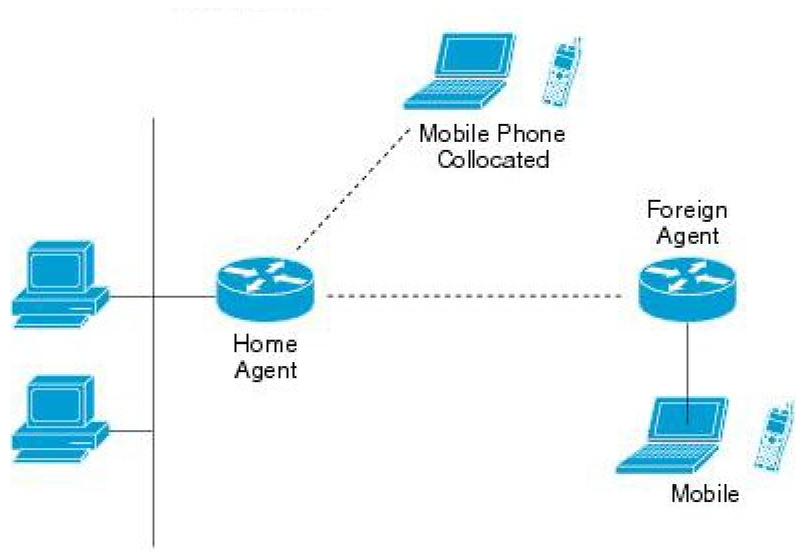


Fig 6: High Availability Clusters

- **Load Balancing Clusters** - Load-balancing [24] when multiple computers are linked together to share computational workload or function as a single virtual computer. Logically, they are multiple machines, but function as a single virtual machine. Requests initiated from the user are managed and distributed among, all the standalone computers to form a cluster. This results in balanced computational work among different machines, of the cluster system.

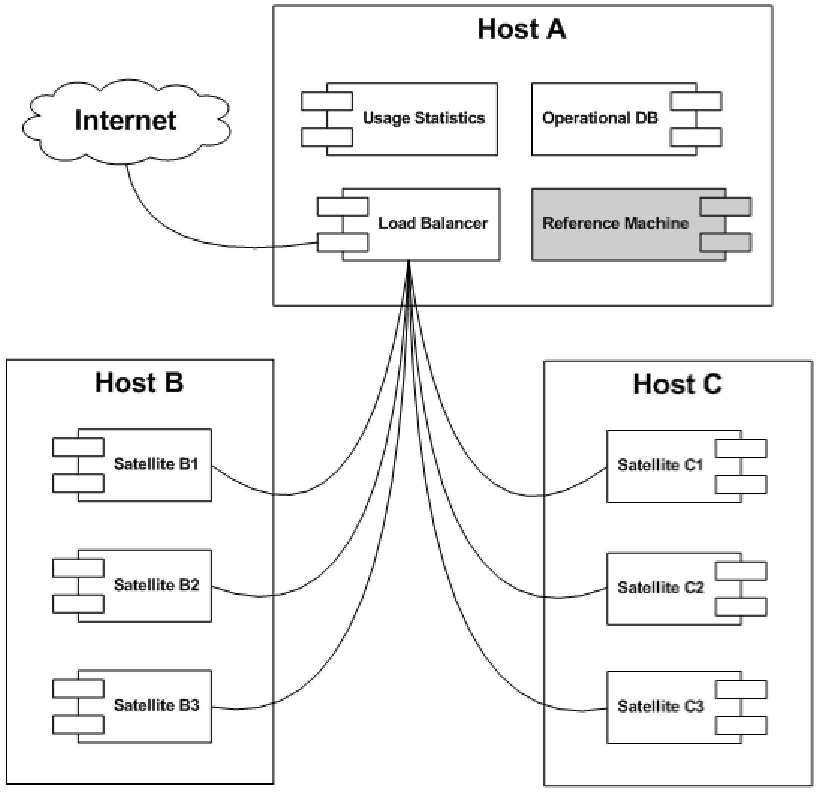


Fig 7: Load Balancing Clusters

- **Compute Clusters** [24] **-** Often clusters are used primarily for computational purposes, rather than handling IO-oriented operations such as web service or databases. For instance, a cluster might support computational simulations of weather or vehicle crashes. The primary distinction within compute clusters is how tightly-coupled the individual nodes are. For instance, a single compute job may require frequent communication among nodes - this implies that the cluster shares a dedicated network, is densely located, and probably has homogenous nodes. This cluster design is usually referred to as Beowulf Cluster [18]. Tightly-coupled compute clusters are designed for work that might traditionally have been called "supercomputing".

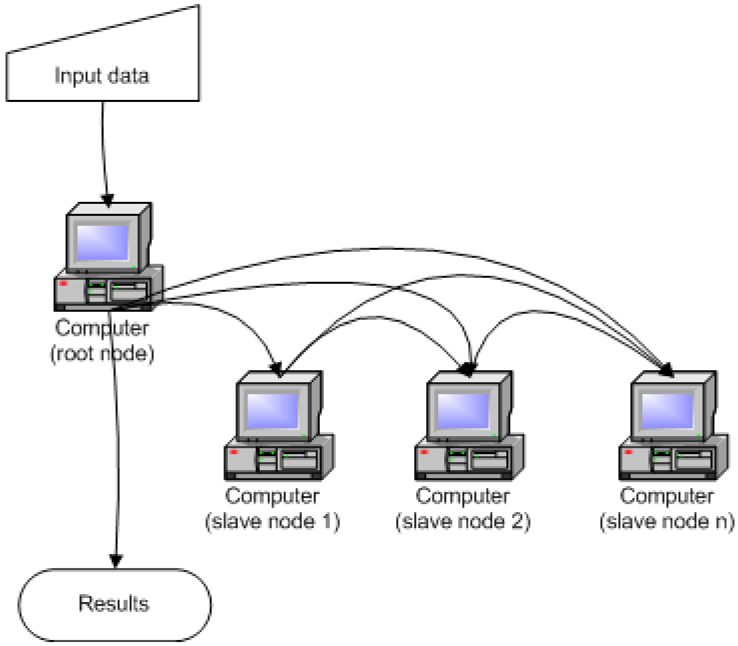


Fig 8: Compute Clusters

- **Grid Clusters -** The other extreme is where a compute job uses one or few nodes, and needs little or no inter-node communication. This latter category is sometimes called "Grid" computing [25]. Grid computing is optimized for workloads which consist of many independent jobs or packets of work, which do not have to share data between the jobs during the computation process. Grids serve to manage the allocation of jobs to computers which will perform the work independently of the rest of the grid cluster.

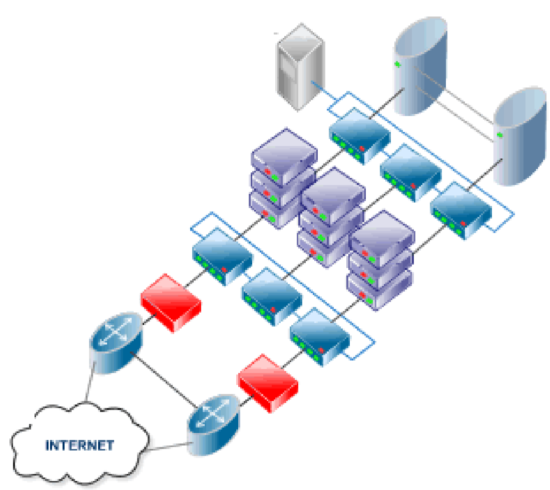


Fig 9: Grid Clusters

## 3.4 Multi – Clusters Architecture

Most hierarchical clustering architectures for mobile radio networks are based on the concept of *Cluster Head.* The Cluster Head acts as a local coordinator of transmissions within the cluster. It differs from the base station concept in current cellular systems, in that it does not have special hardware and in fact is dynamically selected among the set of stations. However, it does extra work with respect to ordinary stations, and therefore it may become the bottleneck of the cluster. To overcome these difficulties, in our approach we eliminate the requirement for a ClusterHead altogether and adopt a fully distributed approach for cluster formation and intra-cluster communications.

In cluster based systems [9] [19], network nodes are partitioned into several groups. In each group, one node is elected to be the cluster-head while the rest of the nodes become ordinary nodes. The cluster size is controlled by the cluster-head's transmission power i.e. its communication range. The ClusterHead coordinates transmissions within the cluster, handles inter-cluster traffic and delivers all packets destined to the cluster; it may also exchange data with nodes that act as gateways to the wired networks. The cluster-based network architectures, the lifetime of the network is strongly related to Cluster Head's failure. Cluster Heads therefore experience high energy consumption and exhaust their energy resources more quickly than ordinary nodes do.

The procedure of cluster formation consists of two phases:

- Cluster-head election
- Assignment of nodes to cluster-heads.

This assignment of nodes to a ClusterHead is one of the major deciding factors for the life of a network. Also after the assignment of these, the balancing of load in terms of the number of nodes serviced by the ClusterHead is important.

# CHAPTER 4
# CLUSTERING ALGORITHMS

## 4.1 Related Work

Ad hoc networks do not use a fixed architecture for communication. In order to provide efficient communication there need to be a wireless backbone [21]. The backbone must change to reflect the changes in the network topology as the nodes move. The algorithm to choose the backbone should be very fast and efficient as they involve mobile nodes powered by battery. Clustering is a solution to the above problem. Even clustering suffers overhead from cluster formation and maintenance. Since the nodes are powered by limited supply of power, the objective is to enhance the existing clustering algorithms to make them a bit more efficient.

Many clustering algorithms [6] [7] [23] [24] are proposed till date. In this thesis we first present the study of some of the popular clustering algorithms which include HC [6] [7], LID [23] and Fuzzy Logic Based approach [24].

## 4.2 Parameter Identification

Before we start discussing the Algorithms, we must first define a few major parameters using which these algorithms have been implemented.

- **Work Budget**: The amount of work that can be done using a fully charged battery by a single node. This can also be referred as the total energy stored in a node.
- **Node Range**: The maximum communication range of a single node. Here we have assumed it to be 25 units.
- **Node ID**: The node IDs.
- **Work Done:** The amount of energy/battery depleted by making one communication with one node.
- **Mobility Pattern**: Non Deterministic i.e. they are independent to move.
- **Sample Area**: The total network area. Here we have assumed it to be 200x200 sq units.

While implementing all these algorithms we have used a set of common algorithms for calculating the distance between the nodes, random work deduction, and mobility of the nodes.

## 4.3 Highest Connectivity Algorithm

The Highest Connectivity algorithm [6] [7] chooses a cluster head based on the degree of connectivity with the other node. So when the algorithm starts executing, the node with the highest connectivity with the other nodes becomes the cluster head. From the remaining set of nodes the next cluster head is chosen using the same conditions/parameters.

The HC algorithm helps to identify the cluster head very quickly, but we face a serious problem of one of the nodes being loaded with more responsibility as compared to the other nodes.

**Parameters**

- SOC : set of all the cluster heads
- CHN: Cluster Head Node
- NT: Total number of nodes in the sample are
- CVR: The nodes covered under a cluster head
- NR: Node Range(25 units)
- $N_i$ : Node under consideration

**Implementation**

```
Begin NodeCHK
        for(i=0;I<=NT;i++)
            for(j=0;j<=NT;j++)
                  if(Ni < NR && Ni != CVR)
                        Ni = CVR for CHNi
                  end if
             end for
           end for
end NodeCHK


Begin HCelect
        for(i=0;i<=SC;i++)
                MAX(Ci)
                CHNi = SOC
        end for
end HCelect
```

Once this has been done we deduct the random work done by the cluster head in communicating with the rest of the nodes.

One major flaw of this election algorithm is that the no of nodes under one cluster head is relatively high as compared to others. Also on increasing the number of nodes in the sample area, this effect is worsened. Since the probability of nodes being close to one another increases. There is also a probability of the same node becoming the ClusterHead again and again.

## 4.4 Lowest ID Algorithm

The Lowest ID algorithm [23] chooses a cluster head based on the PID of the nodes. Hence the node with PID will always become a ClusterHead. This algorithm picks up the node with the lowest ID and checks its connectivity with its neighboring nodes, and using these nodes it forms the cluster. So when the algorithm starts executing, the node with the Lowest ID becomes the ClusterHead and all the nodes within communication range are covered under this ClusterHead. Then the next node with the lowest ID selected to become the ClusterHead is chosen using the same conditions/parameters.

**Parameters**

- SOC : set of all the cluster heads
- CHN: Cluster Head Node
- NT: Total number of nodes in the sample area
- CVR: The nodes covered under a cluster head
- NR: Node Range(25 Units)
- $N_i$ : Node under consideration

**Implementation**

```
Begin NodeCHK
        for(i=0;I<=NT;i++)
            for(j=0;j<=NT;j++)
                  if(Ni< NR && Ni != CVR)
```

```
                    Ni = Covered for Ci
                end if
            end for
        end for
end NodeCHK


Begin LIDelect
    for(i=0;i<=N;i++)
        if(NPID = lowest && Ni != Covered)
        Ni = SC
        Ni = present ClusterHead
    end for
end LIDelect
```

This algorithm is a bit better than the HC algorithm because this allows different nodes to become the ClusterHead. But this algorithm has a flaw; the node with PID 1 will always become the cluster head, will deplete its battery faster and will be the first one to drop out of the network most of the times. Also that a single node can become a Cluster Head.

# CHAPTER 5
# PROPOSED ENHANCEMENT

## 5.1 Proposed Enhancement

Using the above two algorithms and incorporating the use of fuzzy logic [24] and concept of load transfer in the determination of the ClusterHead we propose an enhancement to the whole process of clustering.

The Clustering is divided into three phases: Election of the ClusterHead, Selection of the ClusterHead, and Load Transfer from one ClusterHead to another [25].

**Election of the ClusterHead**

The election of the ClusterHead is done using the LID algorithm.

**Parameters**

- SOC : set of all the cluster heads
- CHN: Cluster Head Node
- NT: Total number of nodes in the sample are
- CVR: The nodes covered under a cluster head
- NR: Node Range(25 Units)
- $N_i$ : Node under consideration
- NC: Set of respective ClusterHead that can take the current node under its own cluster

**Implementation**

```
Begin NodeCHK
          for(i=0;i<=NT;i++)
              for(j=0;j<=NT;j++)
                     if(Ni < NR && Ni != CVR)
                            Ni = CVR for Ci
                            CHNi = NCi
                     end if
               end for
            end for
end NodeCHK
```

```
Begin LIDelect
          for(i=0;i<=N;i++)
                   if(N_PID = lowest && Ni != CVR)
                            Ni = SOC
                            Ni = present Cluster Head
                   end if
          end for
end LIDelect
```

**Selection of the ClusterHead**

The selection of the ClusterHead is based on a Fuzzy decision made by the nodes that will be coming under different ClusterHead. For this fuzzy selection we use a few more parameters, namely work budget left and number of nodes under the ClusterHead. Based on these two values the node will decide its ClusterHead.

**Parameters**

- SOC : set of all the cluster heads
- CHN: Cluster Head Node
- NT: Total number of nodes in the sample are
- CVR: The nodes covered under a cluster head
- NR: Node Range(25 Units)
- $N_i$ : Node under consideration
- NC: Set of respective ClusterHead that can take the current node under its own cluster
- NUN: Number of Nodes Under the ClusterHead
- WB: Work Budget left

**Implementation**

```
Begin FuzzyCHK
        for(i=0;i<=SOC;i++)
                A1 = AVG(WB)
                A2 = AVG(NN)
        end for

        for(i=0;i<=SOC;i++)
               if(Ni belong to NC)
                   T1= MAX(WBi from SOC)
                   T2 = MIN(NNi from SOC)

                   M1 = MOD(AVG(WB)-T1)
                   M2 = MOD(AVG(NN)-T2)
                   M3= M1*M2

                   If(MIN(M3))
                        Ni  = CVR under Ci
                   end if
               end if
          end for
end FuzzyCHK
```

In the above algorithm first calculate the average work budget left for all the ClusterHead and the average number of nodes under a ClusterHead. We then retrieve the ClusterHead with the maximum work budget left that is the maximum battery life left. We store this value in T1. Similarly we choose the ClusterHead with the least number of nodes under it and store it under the value T2. We then calculate the difference between the maximum WB left and NN. Here we use the MOD function since we want the difference between the values and not the actual value. We then multiply these values and store in an array M3. Now choosing the minimum value from this array will give us the desired ClusterHead for the current node. We multiply because if the WB is high but difference between NN and AVG (NN) is low, we get a lower value as compared to the rest of the combinations. The ClusterHead, having the lowest value will have a relatively higher work budget left and a lower number of nodes under it is selected.

**Load Transfer**

This part of the enhancements is done to reduce the effect of increased load when a new node gets admitted into the cluster. The above algorithm works well, but if the node tries to join at a later stage, the ClusterHeads tends to transfer this node to another ClusterHead which is relatively under loaded.

**Parameters**

- SOC : set of all the cluster heads
- CHN: Cluster Head Node
- NT: Total number of nodes in the sample are
- CVR: The nodes covered under a cluster head
- NR: Node Range(25 Units)
- $N_i$ : Node under consideration
- NC: Set of respective ClusterHead that can take the current node under its own cluster
- NUN: Number of Nodes Under the ClusterHead
- WB: Work Budget left

**Implementation**

```
Begin TransferCHK

        for(i=0;i<=NC;i++)
                if(Ni belong to NC)
                    T1= ( WBi from SC)
                    T2 = ( NUNi from SC)

                    M1 = MOD(AVG(T1-Ti))
                    M2 = MOD(AVG(T2-Ti))
                    M3= M1*M2

                   If(MIN(M3))
                       Transfer Ni to CHNi

                   end if
                 end if
            end for
end TransferCHK
```

When a new request comes to a ClusterHead, the ClusterHead looks for all the ClusterHeads which can take this node under its own cluster. The ClusterHead compares its existing work budget and number of nodes under it. So the ClusterHead with relatively lower load will accept this incoming node.

# CHAPTER 6
# SIMULATION AND RESULTS

## 6.1 Simulation Bed

The test bench that we have used:

- Mobility pattern is non deterministic.
- Movement not more than ½ the range of a node per unit time.
- 200 x 200 Sq. Unit Areas.
- Size of the cluster is not known.
- Total number of nodes = {25, 50, 75,100}.
- Sample time = 100ms.
- Node Range = 25 Units (fixed).
- Work Budget = 5000 Units.
- Random Work deducted with a max work done of 10.
- Programming Language JAVA.

The simulator is a software tool designed in Java. The simulated ad hoc network is composed of slowly changing network topology. The topology is a square area with 200 length and 200 width. The network nodes are randomly distributed. Initial energy (Work Budget) of all the nodes in the network is 5000. The range of each node is 25 units which is taken as input to the program. The movement speed of a node can vary between 0 and ½ the communication range of a node.

The mobile nodes move according to the “random waypoint" model. Each mobile node begins the simulation by remaining stationary for pause time seconds. It then selects a random destination in the defined topology area and moves to that destination at a random speed. The random speed is distributed uniformly between zero and maximum speed of not more than ½ the communication range of a node. Upon reaching the destination, the mobile node pauses again for pause time seconds, selects another destination, and proceeds there as previously described. This movement pattern is repeated for the duration of the simulation.

## 6.2 Results

One of the major parameter where we can easily asses the quality of the algorithm is the average ClusterHead Time. This gives us a fair idea of the network lifetime. Hence we compare our proposed algorithm with the existing algorithm.

**HC Algorithm**

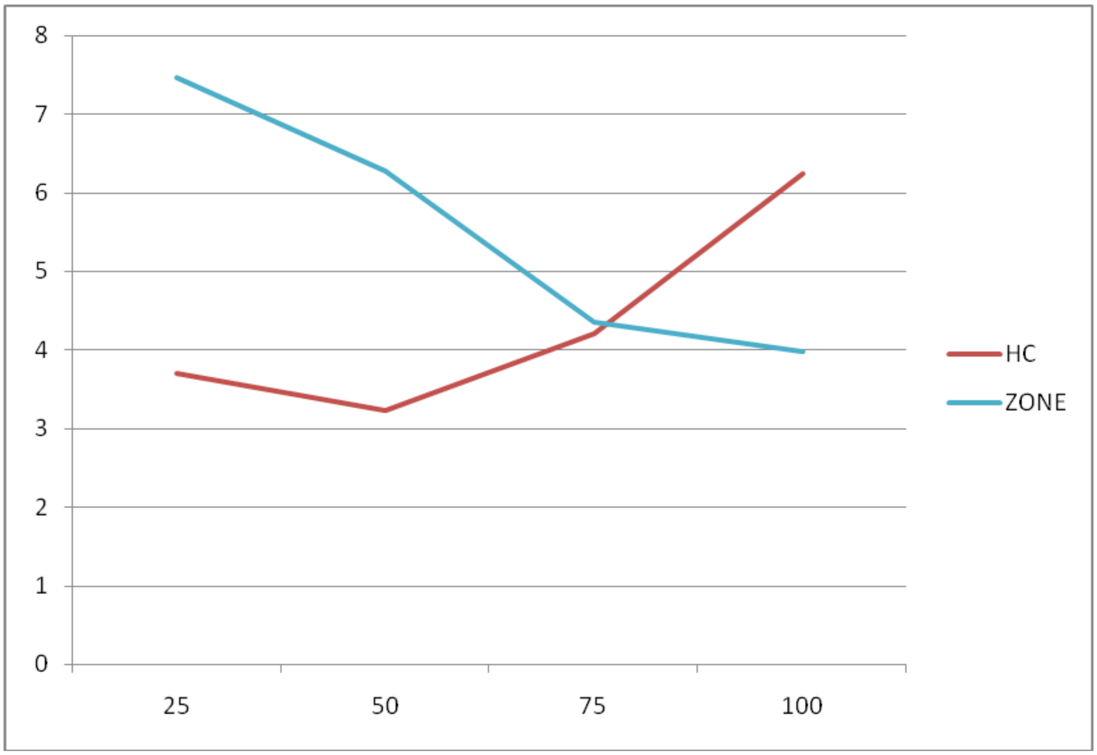


Fig 10: HC vs Proposed Enhancement

The network shows low network lifetime in HC initially because the no of nodes are less. Less nodes amount to lower network lifetime. With the enhancement incorporated we see that the initial network lifetime i.e. with less number nodes has improved because there was a better distribution of nodes under each ClusterHead.

**Lowest ID:**

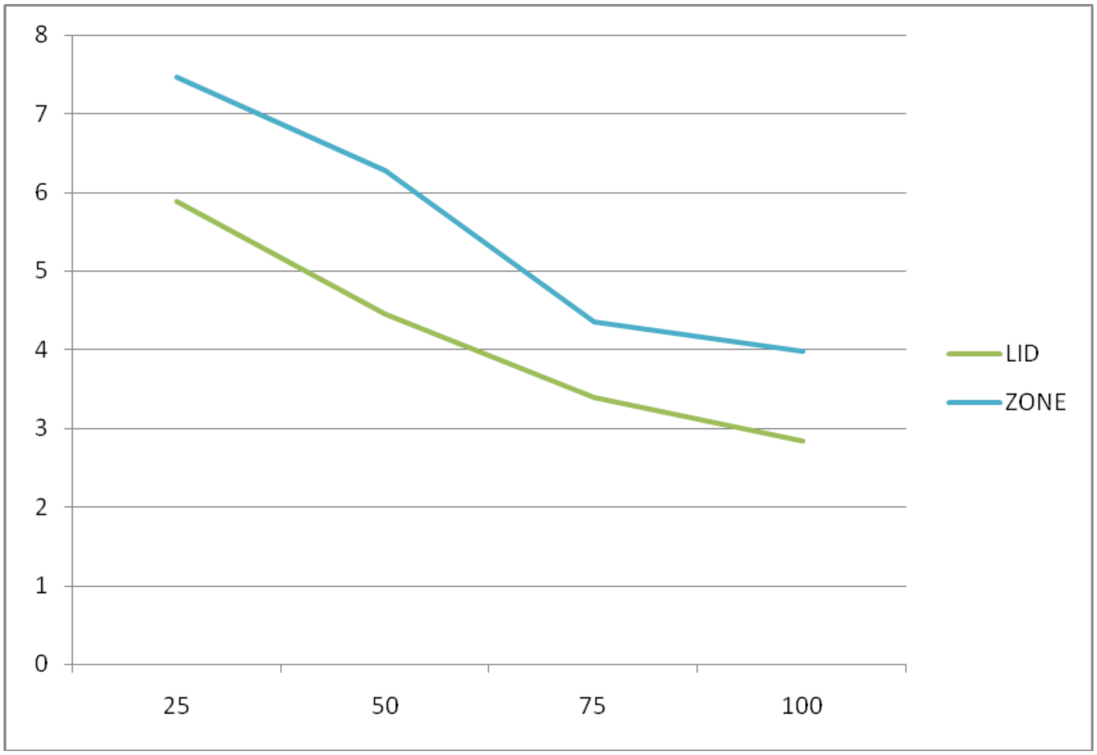


Fig 11: LID vs Proposed Algorithm

The network shows low network lifetime in LID initially because the no of nodes are less. Hence even single nodes are taken as ClusterHead. Less nodes amount to lower network lifetime. With the enhancement incorporated we see that the initial network lifetime i.e. with less number nodes has improved because there was a better distribution of nodes under each ClusterHead. Also it shows a consistent improvement to the overall system.

# CHAPTER 7
# CONCLUSION

Efficient energy usage via clustering i.e. utilizing the available limited amount of energy in deploying the network in the most efficient way is one of the greatest challenges faced by an ad-hoc system. Although there have been certain algorithms proposed to deal with it, however they do not provide a complete energy-efficient network.

The existing algorithm i.e. HC is efficient in building the clusters and deciding upon which nodes will be elected as the cluster head. But the HC algorithm suffers a major flaw related to the way in which it makes these clusters. Since it uses the degree of connected nodes as a parameter, a few nodes are over loaded as compared to the other nodes. This increases the battery consumption as this node has to shoulder a lot of responsibility. Thus, the probability of this node dropping out of the network is high.

The LID algorithm has a better way of making clusters. By choosing the uncovered node having the lowest physical ID as the cluster head, it forms the cluster taking the nodes in direct communication range. This selection based on the physical ID is easy and works with least ambiguity. With this algorithm even a single node can be a cluster head. But this type of election is to out disadvantage since this single node reduces the average cluster head time. Also, the node with physical ID 1 will always be selected as the cluster head, and this will be the first node to drop out of the network unless some other cluster head is serving at least ½ the nodes in the whole sample area.

With the base algorithms being a bit inefficient in clustering in terms of the average Cluster Head time, we have proposed an enhancement. The proposed enhancements helps the existing algorithm LID to maintain a better network as compared to LID alone by distributing load over to a relatively under loaded ClusterHead and helps in the selection of ClusterHead by monitoring two major parameters of the ad hoc network i.e. the work budget left and the number of nodes serviced by the cluster head. This improvement in the network lifetime is due to the change in the election and selection algorithm, and also due to the transfer of node when there is a new admission to the system after it starts running. As the number of nodes in the sample area increases the network lifetime decreases because all the ClusterHead have to shoulder greater responsibility. But this fall in the network lifetime is stabilized as the nodes increase.

# REFFERENCES


[1] "Overview of Wireless Communications", http://www.cambridge.org/us/catalogue/catalogue.asp?isbn=0521837162&ss=exc

[2] "Getting to Know Wireless Networks and Technology", http://www.informit.com/articles/printerfriendly.aspx?p=98132

[3] A. Tanenbaum. *Computer Networks (3rd Edition)*. Prentice Hall, Upper Saddle River, N.J., 1996.

[4] Alan D. Amis, Ravi Prakash. "Load Balancing Clusters in Wireless Ad Hoc Networks", *Proceedings of the 3rd IEEE Symposium on Application-Specific Systems and Software Engineering Technology*, 2000.

[5] "Wireless Networks" , http://en.wikipedia.org/wiki/Wireless_network

[6] M. Gerla and J. T.-C. Tsai. Multicluster, mobile, multimedia radio network. *ACM Baltzer Journal of Wireless Networks*, 1(3):255–265, 1995.

[7] Abhay K. Parekh. "Selecting Routers in Ad-Hoc Wireless Networks". In *ITS*, 1994.

[8] J. Li, and Y. C. Tay M. Jiang, "Cluster Based Routing Protocol," Internet Draft 1999.

[9] S.Kumar C.Chong, "Sensor Networks: Evolution, Opportunities," in *Proc. IEEE*, Aug. 2003, pp. 1247-1256,vol. 91, no. 8.

[10] I. F. Akyildiz et al, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, pp. 393-422, March 2002.

[11] J. Hill, "System Architecture for Wireless Sensor Networks," PhD Thesis Spring 2003.

[12] A. Manjeshwar and D. P. Agrawal, "TEEN : A Protocol for Enhanced Efficiency in Wireless Sensor Networks," in *1st International Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing*, San Francisco, CA, April 2001.

[13] http://en.wikipedia.org/wiki/Wireless_ad_hoc_network

[14] E. M. Royer and C.K. Toh, "A Review of Current Routing Protocols for Ad-Hoc Mobile

Wireless Networks," *IEEE Personal Communications Magazine*, pp. 46–55, April 1999.

[15] C. E. Perkins, *Ad Hoc Networking*.: Addison-Wesley, 2001.

[16] R. Ramanathan and J. Redi, "A Brief Overview of Ad Hoc Networks:Challenges and Directions," *IEEE Communication Magazine*, vol. 40, no. 5, 2002.

[17] Srikanth Krishnamurthy Prasant Mohapatra, Springer Science+Business Media, 2005.

[18] http://en.wikipedia.org/wiki/MANET

[19] Tomas Johansson and Lenka Carr-Motyˇckov´, "On Clustering in Ad Hoc Networks," Division of Computer Science and Networking Lule°a University of Technology, August 17,2003.

[23] Anthony Ephremides, Jeffery Wieselthier, Dennis Baker. "A Design Concept for Reliable Mobile Radio Networks with Frequency Hopping Signaling," in *Proceedings of the IEEE Vol 75*,*No. 1,* January 1987, pp. 56-72.

[24] Shahram Nourizadeh, Y. Q. Song, J. P. Thomesse. " A Location-Unaware Distributed Clustering Algorithm for Mobile Wireless Networks Using Fuzzy Logic", *In 7th IFAC International Conference on Field buses and Networks in Industrial and Embedded Systems (2007) Vol. 7, Part 1*, 2007.

[25] Mukesh Shingal, Niranjan Shivaratri. "Advanced Concepts in Operating Systems", 2009, Chapter 11.