REDUCING POWER CONSUMPTION IN CLUSTER-BASED MOBILE AD HOC NETWORKS

By
Rani Abdullah Al-Maharmah

Supervisor
Dr. Imad Salah

Co-Supervisor
Dr. Nadim Obeid, Prof.

This Thesis was Submitted in Partial Fulfillment of the Requirements for the Master's Degree of Computer Science

Faculty of Graduate Studies
The University of Jordan

August, 2008

COMMITTEE DECISION

This Thesis (REDUCING POWER CONSUMPTION IN CLUSTER-BASED MOBILE AD HOC NETWORKS) was successfully defended and approved on 5 / 8 / 2008.

Examination Committee	**Signature**
Dr. Imad Salah, Chairman **Assoc. Prof. of Computer Networks**	_______________
Dr. Nadim Obeid, Prof., Member **Prof. of Artificial Intelligence**	_______________
Dr. Mohammad Al-Qatawneh, Member **Assoc. Prof. of Computer Networks**	_______________
Dr. Saleh Al-Sharaeh, Member **Assoc. Prof. of Computer Networks**	_______________
Dr. Saleh Al-Oqeili, Prof., External Examiner **Prof. of Computer Networks** **(The Hashemite University)**	_______________

DEDICATION

I dedicate this thesis to my beloved parents Mr. Abdullah Al-Maharmah and Mrs. Maha Al-Maharmah, who gave me an appreciation of learning and taught me the value of perseverance and resolve.

I also dedicate this work to my brothers (Lowai and Bashar) and sisters (Rana, Maes, and Dana) for their unfaltering support and understanding while I was completing this thesis.

ACKNOWLEDGEMENT

This dissertation could not have been written without the blessing of Allah, the lord of the worlds and the king of kings.

Grateful thanks to Dr. Imad Salah, who not only served as my supervisor but also encouraged me throughout my academic program. His extraordinary supervision and invaluable guidance inspire me to complete this work. I wish to thank him for the time he spent on directing me thus he has time constraints. I also would like to thank Prof. Nadim Obeid for his extra supervision and guidance on my work. He inspired me with leading thoughts and ideas that made this work possible, and never accepting less than my best efforts. I thank them all for their valuable participations, insights, and constructive guidance.

Special thanks to my colleagues at the University of Jordan for their support and encouragement throughout this thesis. Finally, many thanks to every one gave hints, offered support, or shared thoughts to improve this work.

Rani A. Al-Maharmah
Amman, 2008

TABLE OF CONTENTS

LIST OF FIGURES AND PLATES

LIST OF TABLES

LIST OF FORMULAS

LIST OF ALGORITHMS

LIST OF ABBREVIATIONS

AP	Access Point
BMT	Border Mobile Terminal
BPL	Battery Power Level
BSS	Basic Service Set
CBLARHM	Cluster Based Location-Aware Routing Protocol for Large Scale Heterogeneous MANET
CH	Cluster Head
IEEE	Institute of Electrical and Electronic Engineers
GloMoSim	Global Mobile Information System Simulator
GPS	Global Positioning System
GSM	Global System for Mobile Communications
GW	Gate Way
MACHM	Multi-Aware Cluster Head Maintenance
MANET	Mobile Ad-hoc NETwork
MT	Mobile Terminal
PCS	Personal Communication Service
PDA	Personal Digital Assistant
RWPM	Random Waypoint Mobility Model
ToA	Time of Arrival
WCA	Weighted Clustering Algorithm
WEAC	Warning Energy Aware Clusterhead

WLAN	Wireless Local Area Network
WMAN	Wireless Metropolitan Area Network
WNIC	Wireless Network Interface Card

REDUCING POWER CONSUMPTION IN CLUSTER-BASED MOBILE AD HOC NETWORKS

By
Rani Abdullah Al-Maharmah

Supervisor
Dr. Imad Salah

Co-Supervisor
Dr. Nadim Obeid, Prof.

ABSTRACT

Node's resources are limited and valuable in Mobile Ad Hoc Networks (MANETs), most importantly, battery power, which is limited due to the relatively small size of mobile nodes. Managing this limited resource is a key challenge in MANET's environments.

In order to achieve scalability, hierarchical solutions were adopted like cluster-based concept. Electing the cluster heads and maintaining them throughout the progress of the ad hoc networks are vital and critical functions.

Many proposals that considered various factors, including the energy factor, have been put forward in order to elect the optimal cluster heads available. We have considered a multi-aware approach namely, Multi-Aware Cluster Head Maintenance (MACHM) for electing cluster heads based on a weighted formula that includes load balancing, geographical, mobility, and energy factors.

MACHM aimed to reduce the total amount of the power consumed by the nodes in the network. Especially, cluster heads, which are more sensitive to power drains because of their important roles in the network.

We have proposed a mechanism to maintain the previously elected cluster heads based on their energy level, that looking for a replacement cluster head from the set of direct neighbors to reduce the effect of changing the cluster head. Simulation results showed that our method outperformed Weighted Clustering Algorithm (WCA) in the number of cluster heads, invoke requests, percentage of cluster heads and all nodes power consumption, and throughput of data messages metrics.

1. Introduction

1.1 Wireless Networks

Wireless networks have significant impact and there are growing demands on wireless communications for many reasons, such as their ease of use, increased computation power (*ex. Laptops, Palmtops, PDAs, etc.*), reduced cost, and mobile capabilities (Hong *et al.*, 2002).

Wireless network term refer to a telecommunication network whose interconnections between nodes is implemented without the use of wires. Wireless telecommunications networks are generally implemented with some type of remote information transmission system that uses electromagnetic waves, such as radio waves, for the carrier and this implementation usually takes place at the physical layer of the network (Tanenbaum, 1997).

Examples on wireless networks include: Wireless Local Area Network (WLAN), Wireless Metropolitan Area Network (WMAN), Global System for Mobile Communications (GSM), Personal Communication Service (PCS), *etc*. Figure 1.1 shows an overview of the range and data rate of various wireless networks types.

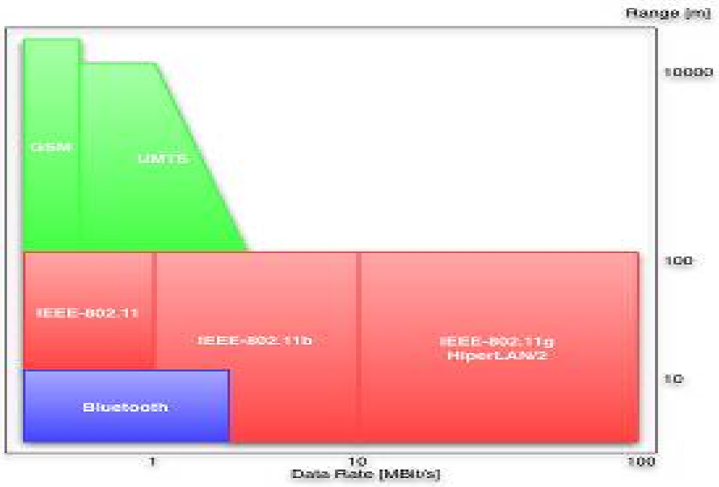


Figure 1.1: Range and Data Rate of Wireless Technologies (URL)

Compatibility issues are very important when dealing with wireless networks. To overcome this, Institute of Electrical and Electronic Engineers (IEEE) developed a set of standards for WLAN computer communication known as IEEE 802.11. The 802.11 family includes over-the-air modulation techniques that use the same basic protocol, the most popular are those defined by the 802.11b and 802.11g protocols (Flickenger *et al.,* 2006).

WLAN based on radio waves to enable communication between devices in a limited area, also known as the Basic Service Set (BSS). This gives users the mobility to move around within a broad coverage area and still be connected to the network. All components that can connect into a wireless medium in a network are referred to as stations. All stations are equipped with Wireless Network Interface Cards (WNIC) (Flickenger *et al.,* 2006).

There are two types of BSS: infrastructure BSS (*Figure 1.*2) and independent BSS (*Figure 1.*3). An infrastructure BSS can communicate with other stations (*even outside the same BSS*) by communicating through access points (AP), while an independent BSS is an AD-Hoc Network where stations can communicate without relaying on access points (Halsall, 2005).

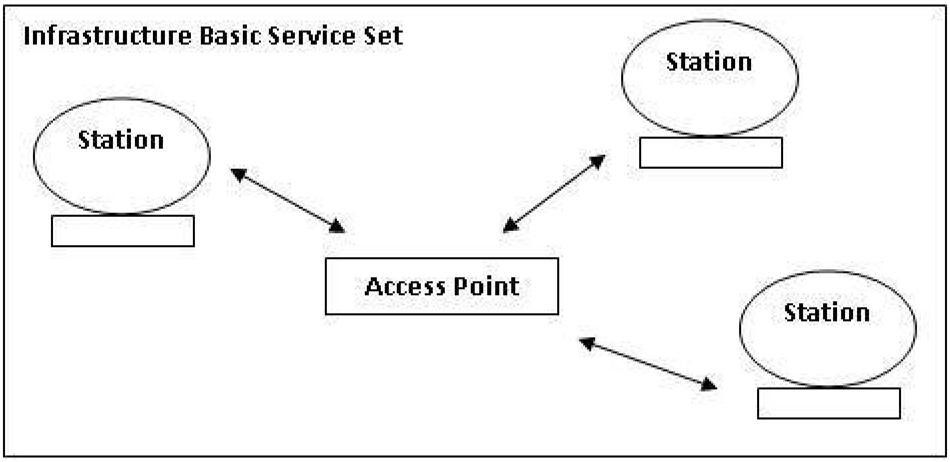


Figure 1.2: Infrastructure BSS

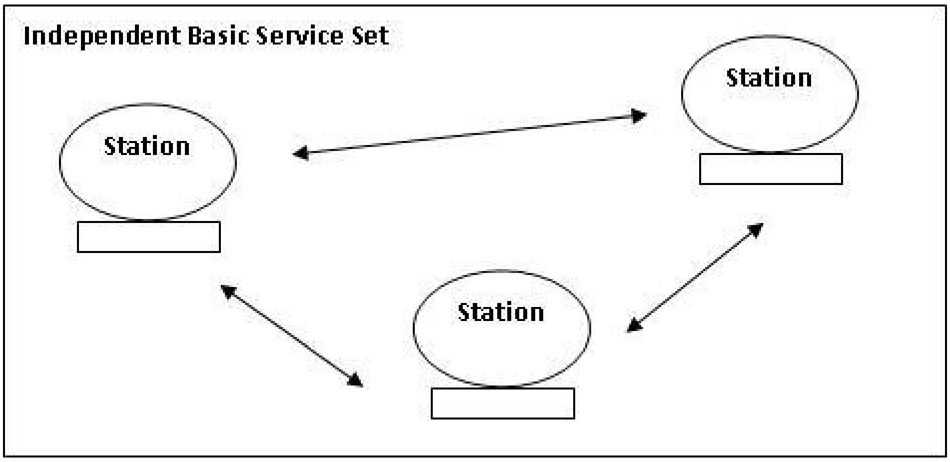


Figure 1.3: Independent BSS

1.2 Mobile Ad Hoc Networks

Mobile **A**d-Hoc **NET**work (MANET) is a multi-hop wireless network, which means that a node (*Device*) can make contact with another node that is not in the range of its transmission through other nodes. Thus**,** mobile nodes play two roles: Senders/Receivers (*of data or control packets*) and Routers.

MANET consists of a number of mobile nodes (*Vary from very small to very large*) (Kuosmanen, 2002). MANETs are self-organizing mobile wireless networks with decentralized control of operations; they do not rely on a preexisting infrastructure (*ex. Access Point*) to communicate (Sasson *et al*., 2005).

Nodes are free to move randomly and organize themselves arbitrarily; thus, the network's wireless topology may change rapidly and unpredictably. Nodes are entering and exiting the range of other nodes, the network has to be able to react on these topology changes and fix broken connections or calculate new routes through the network.

MANET is useful in areas where the fixed (*infrastructure-based*) solutions seem inflexible. This means various applications and implementations; examples include: military, emergency, and conferencing applications (Latiff *et al.,* 2005).

One of the hardest and largest applications is utilized in military environments. The way wars are being fought today has changed drastically. There is no fixed infrastructure when operating in a foreign country. Even when defending own country these infrastructures are most likely to be damaged or destroyed by enemy forces, so it is more critical than in civilian applications.

Another huge public application may be in the area of emergency services (*Firefighters, Police, etc*), sometimes they have to operate in areas where no information infrastructure is present and operations still need to be coordinated. Also, MANET is useful in conferences when there are no fixed office network infrastructures. MANET will enable exchanging data and supporting cooperative work.

However, MANET's environment suffers from classical challenges like limited bandwidth because all wireless communication links share the same medium; these interferences reduce the efficiency of the network. Limited battery power is also a challenge, where mobile devices are battery powered. Routing calculations and sending control and data packages through the network drains energy. Other hazards are short radio coverage, frequent topology changes, and limited security. Wireless communication is much more vulnerable to security issues (*denial-of-service attacks, bugging, etc.*) than hard-wired connections since the physical media cannot be protected from foreign access (Hong *et al.*, 2002).

1.3 Routing Protocols Classifications

Routing is the problem of delivering packets from source to destination (*Unicast*) or destinations (*Multicast*). Traditional classification is (Kuosmanen, 2002):

- Proactive Routing Protocols (Table-Driven).
- Reactive Routing Protocols (On-Demand).

Table-Driven routing protocols try to maintain consistent, up-to-date routing information from each node to every other node. In table-driven the routes are known immediately when they are needed but at a high overhead cost to collect the routing information. On-Demand routing protocols create routes only when these are needed. This results in a reduced routing overhead cost but at the expense of a route establishment delay.

The previous classification has though some drawbacks, like its generality and lack of classing (Kuosmanen, 2002), so other classifications exist.

Other classification is presented in (Hong *et al.*, 2002):

- Flat Routing Protocols.
- Hierarchical Routing Protocols.
- Geographic Routing Protocols (Position-Based).

Flat Routing Protocols adopt a flat addressing scheme, which means that each node participating in routing plays an equal role. In contrast, Hierarchical Routing Protocols usually assigns different roles to network nodes (*ex. Head, Gateway, etc.*). Geographic Routing Protocols, however, get the advantage of nodes location information knowledge and use different techniques to route packets.

Other classification based on the type of cast is presented in (Kuosmanen, 2002):

- Unicast Routing Protocols.
- Multicast Routing Protocols.
- Geocast Routing Protocols.

In Unicast Routing Protocols, one source transmits data packets to one destination, i.e., it is one-to-one communication**.** In Multicast Routing Protocols, the source sends data packets to more than one destination (Multicast Group) by constructing routing tree or mesh, i.e., it is one-to-many communication. In Geocast Routing Protocols the source transmits data packets to a group of nodes (Geocast Group) which are situated on at specified geographical area.

The main difference between Multicast and Geocast is in the first the group members join the multicast group explicitly (*ex. By sending a JOIN_PACKET*), while in the latest the group members are known implicitly (*ex. By their locations*).

Other classification is (Mauve *et al.*, 2001):

- Topology-Based Routing Protocols.
    - Proactive.
    - Reactive.
    - Hybrid.
- Position-Based Routing Protocols.

Topology-Based Routing Protocols use the information about the links that exist in the network to perform packet forwarding. Therefore, it employs the principle that every node in the network maintains large-scale topology information. This large-scale topology information may be gathered proactively or reactively (*as mentioned before in the first classification*) or hybrid (*as Hierarchical Routing Protocols*). Position-Based Routing Protocols use additional information about nodes locations and may not need to construct a large-scale topology view.

1.4 Cluster-Based Routing Protocols

Scalability is an important feature in mobile ad hoc networks, but when the number of nodes is large (*Beyond certain threshold*) (Wang *et al.,* 2007), it is inefficient to use flat routing technique, due to the cost and overhead of exchanging control and data packets in the presence of limited bandwidth and battery power (Cano and Manzoni, 2002).

One way to support efficient communication between nodes is to develop wireless backbone architecture (Johansson and Carr-Motyckova, 2003). This means that certain nodes must be selected to form the backbone. Over time, the backbone must change to reflect the changes in the network topology as nodes move around.

One possible solution is to use the ideas of clustering as a hierarchical routing approach. The main idea is to group the nearby nodes into logical groups known as clusters, then assigning nodes different functionalities inside and outside a group (*cluster*) (Hong *et al.*, 2002).

Each group contains a special node that acts as a leader based on some criteria (*ex. Highest ID, Lowest ID, Connectivity, Power Level, Randomly, etc.*) to label the cluster and to communicate to other nodes on behalf of the cluster (Sinha *et al.,* 2001). Other terms used to express the leader are: cluster head (CH) (Haas and Tabrizi, 1998), coordinator (Chen *et al.,* 2001), core (Sivakumar *et al.*, 1999), member of dominating set (Wu *et al.,* 2002), and backbone network (Liang and Haas, 2000).

Electing the cluster head is a critical step that affects the whole performance of the cluster-based routing protocol, therefore different approaches exist for the election process. In Highest ID criterion each node in the network assigned a distinct ID (*ex. IP Address*), after nodes exchanged their IDs with their direct nodes, the node with the highest ID will win the election. Lowest ID works the same way but the node with the lowest ID won. Other approaches depend on the Connectivity factor which means the number of direct neighbors attached to the node. Battery's power level is also a candidate choice; however other approaches simply choose the cluster head randomly (Wang *et al.,* 2006).

This hierarchal approach enables mixing of proactive and reactive routing techniques. Proactive routing can be used inside the clusters where the number of nodes is relatively smaller, while the reactive routing is used in communication between different clusters.

Additional to the cluster head; clusters contain ordinary nodes known as members, and gateways (GW) that are ordinary nodes have neighbors from different clusters so they enables the communication between clusters. Figure 1.4 shows an example.

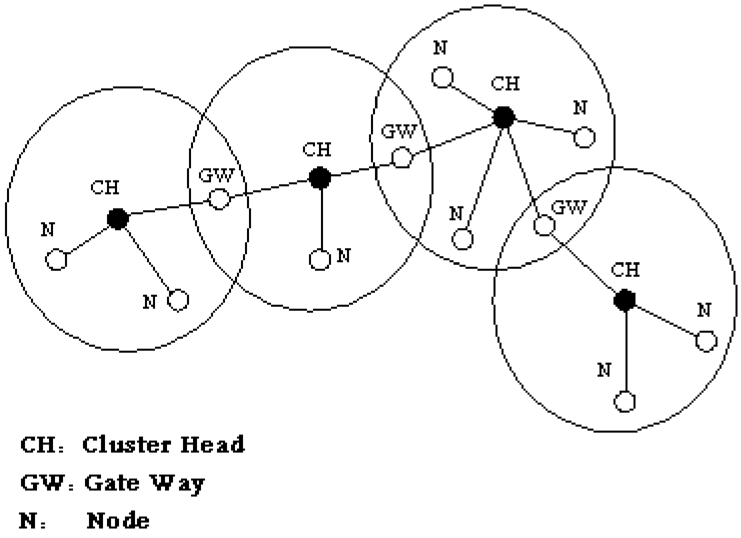


Figure 1.4: Clustering Example

1.5 Problem Statement

As mentioned before, one of the key challenges in MANET's environment is the limited node's battery power. Despite the advancement made in battery technology regarding size and/or power capacity, power consumption remains an important factor to be considered.

In cluster-based routing protocols, cluster heads consume more power than other nodes because they have special roles. Therefore, it is useful to consider the power consumption metric in the process of cluster head election.

(Wu *et al.*, 2007) used the following formula to estimate the node's power consumed to transmit a k-bit message a distance d using the radio model:

$$E_{Tx}(k,d) = (E_{elec} \times k) + (E_{amp} \times k \times d^2) \tag{1.1}$$

Where:

$E_{Tx}(k,d)$: The cost for transmitting a k-bit message to a distance d.

E_{elec}: The power consumption of the circuit itself.

E_{amp}: The power consumed by the amplifier to transmitting packets.

k: The size of message in bits.

d: The distance between sender and receiver.

Consequently, the total power consumption of the cluster head can be calculated by the formula:

$$TPC(CH_X) = \sum E_{TCHx}(k,d) \tag{1.2}$$

Where:

$TPC(CH_X)$: The total power consumed by the cluster head X.

$\sum E_{TCHx}(k,d)$: The summation of the cost of transmitting a k-bit messages to a distances d by the cluster head X.

k: The size of message in bits.

d: The distance between sender and receiver.

Our aim is to find an appropriate approach that will reduce the value of Formula (1.2).

It is clear that reducing node's power consumption will increase the total lifetime of the whole network, and keeps it operative for long as possible, which is desirable

A general survey on clustering and power-aware routing protocols for ad hoc networks can be found in (Yu and Chong, 2005), and (Jones *et al.*, 2001) respectively.

Existing solutions suffer from many problems and limitations such as using only node's power level as a criterion to elect the cluster head will not guarantee an overall performance for the cluster-based routing protocol especially in the cluster head maintenance stage.

There are cluster-based routing protocols that either combine cluster-based with only power-aware ideas like Warning Energy Aware Clusterhead (WEAC) (Sheltami and Mouftah, 2003) or cluster-based with only location-aware ideas like Cluster Based Location-Aware Routing Protocol for Large Scale Heterogeneous MANET (CBLARHM) (Wang *et al.*, 2007).

In our approach we make an attempt to take an advantage of the three concepts (Cluster-based, Power-aware, and Location-aware) at the same time to produce a multi-aware cluster-based routing protocol that deals with the cluster head election and maintenance efficiently. This in result will span the total operative time of the whole network.

1.6 Thesis Organization

The rest of this thesis is organized as follows. In chapter 2 related works will be presented, our multi-aware cluster head maintenance technique will be introduced in chapter 3. Simulation environment and the related results taken from the simulation will be presented and analyzed in chapter 4. Chapter 5 gives the conclusions and suggests the future work.

2. Literature Review

2.1 Introduction

There have been many interests in the area of clustering and power-aware routing in the ad hoc environment, because of the limited energy and node's battery power challenge. In this chapter we will present a brief literature review for the methods and approaches that have been proposed.

2.2 Related Works

2.2.1 Power-Aware Clustering

(Kawadia and Kumar, 2003) proposed CLUSTERPOW algorithm in which dynamic and implicit clustering is suggested on the basis of transmit power level. The transmit power level is the power level required to transmit each packet. The transmit power level to a node inside the cluster is less as compared to the level required to send a node outside the cluster. So here the clustering is performed keeping the nodes with lower transmit power level together. Figure 2.1 shows an example.

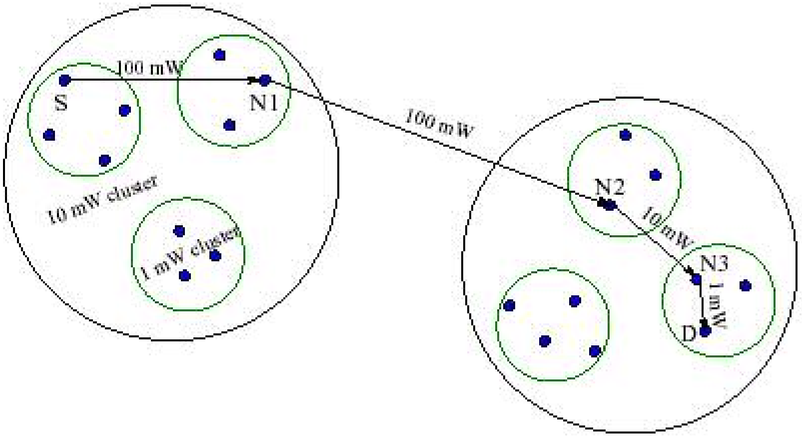
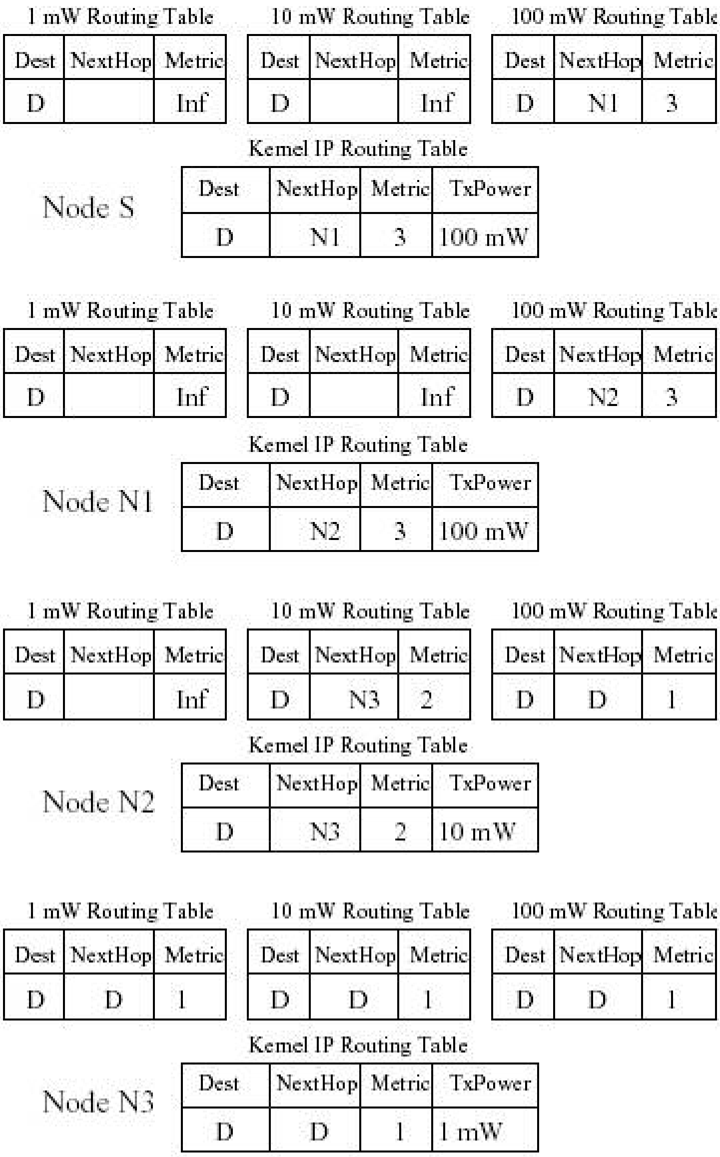


Figure 2.1: Routing by CLUSTERPOW (Kawadia and Kumar, 2003)

In (Kawadia and Kumar, 2003) the primary drawback is that there is no cluster head or cluster gateway. Each node here has routing tables corresponding to different transmit power levels. The routing table for a power level *Pi* in a node is built by communicating with the peer routing table of the same power level at another node. Figure 2.2 shows the tables for the nodes of the network example in Figure 2.1.



Node S

1 mW Routing Table

Dest	NextHop	Metric
D		Inf

10 mW Routing Table

Dest	NextHop	Metric
D		Inf

100 mW Routing Table

Dest	NextHop	Metric
D	N1	3

Kernel IP Routing Table

Dest	NextHop	Metric	TxPower
D	N1	3	100 mW

Node N1

1 mW Routing Table

Dest	NextHop	Metric
D		Inf

10 mW Routing Table

Dest	NextHop	Metric
D		Inf

100 mW Routing Table

Dest	NextHop	Metric
D	N2	3

Kernel IP Routing Table

Dest	NextHop	Metric	TxPower
D	N2	3	100 mW

Node N2

1 mW Routing Table

Dest	NextHop	Metric
D		Inf

10 mW Routing Table

Dest	NextHop	Metric
D	N3	2

100 mW Routing Table

Dest	NextHop	Metric
D	D	1

Kernel IP Routing Table

Dest	NextHop	Metric	TxPower
D	N3	2	10 mW

Node N3

1 mW Routing Table

Dest	NextHop	Metric
D	D	1

10 mW Routing Table

Dest	NextHop	Metric
D	D	1

100 mW Routing Table

Dest	NextHop	Metric
D	D	1

Kernel IP Routing Table

Dest	NextHop	Metric	TxPower
D	D	1	1 mW

Figure 2.2: Routing Tables for all Power Levels, and the Kernel IP Routing Table (Kawadia and Kumar, 2003)

The next hop to route the packet is determined by consulting the lowest power routing table through which the destination is reachable. Thus, this suggests that each node should know the route to other nodes and also know the transmit power level at which a destination node is reachable. This leads to the overhead of collecting the state information and building many routing tables for each power level in a node.

2.2.2 Warning Energy Aware Clusterhead

(Sheltami and Mouftah, 2003) proposed a novel cluster-based infrastructure creation protocol namely: Warning Energy Aware Clusterhead (WEAC). WEAC aims to establish a dynamic wireless mobile infrastructure that scales well to large networks in terms of stability, load balancing, and energy saving.

In their scheme, they used the term Mobile Terminals (MTs) for network nodes or devices. Based on their Battery Power Level (BPL), some MTs are elected to be in charge of other MTs within their transmission range. (Sheltami and Mouftah, 2003) classified MTs to Clusterhead, Zone_MT, Free_MT, and Gateway or Border Mobile Terminal (BMT). They used three power levels thresholds (*THRESHOLD_1, THRESHOLD_2, and THRESHOLD_3*) to characterize the MTs and their role. WEAC's power levels classing is shown in Figure 2.3.

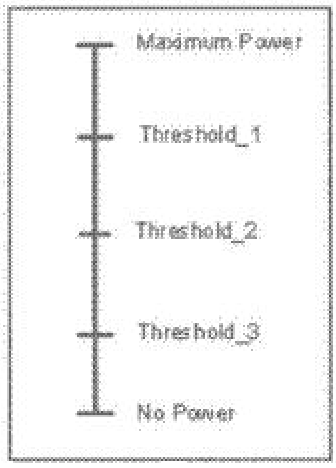


Figure 2.3: Four Power Levels (Sheltami and Mouftah, 2003)

WEAC used the previous thresholds to control the states of MTs and deciding when to send a warning message to look for another cluster head. In WEAC the mission of finding new cluster head is done by ordinary nodes (*Zone_MT and BMT*), which is a main disadvantage for many reasons. The collected and learned information by the previous cluster head will be wasted, and the scope of finding the new cluster head will be expanded, which will put an extra overhead in the network.

2.2.3 Self-Positioning

In (Mellier *et al.*, 2006) nodes execute a distributed algorithm of localization in order to build a global coordinates system. In this global coordinates system each node knows its coordinates. They use a distributed, an infrastructure-free positioning algorithm that does not rely on Global Positioning System (GPS).

(Mellier *et al.*, 2006) goal is to enable nodes to find their positions within the network area using only their local information. They used range measurements between the nodes to build a network coordinates system. The Time of Arrival (ToA) method used to obtain the distance between two mobile devices.

ToA technology used the absolute time of arrival at a certain base station. The distance can be directly calculated because signals travel with a known velocity. However, there still many issues to consider like resolving the precise position between two nodes depending on a third base station, and clock synchronization problem (Tanenbaum, 1995).

In (Mellier *et al.*, 2006), the node becomes the center of its own coordinate system with the position (0,0) and the position of its neighbors are computed accordingly. They used many equations to calculate the distances and build their relative local coordinates system. Figure 2.4 shows an example.

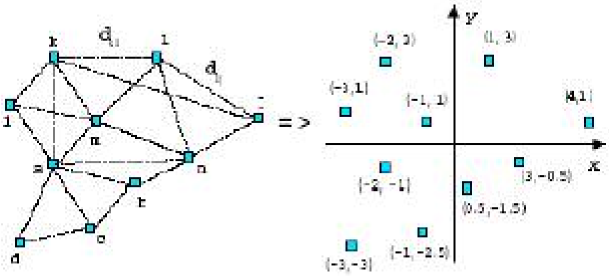


Figure 2.4: Local Coordinates System Example (Mellier *et al.*, 2006)

After the nodes have determined their local coordinate systems, network coordinates system is derived by adjusting the direction of their local coordinate systems by rotating and mirroring.

2.2.4 Weighted Clustering Algorithm

(Chatterjee *et al.,* 2002) proposed an on-demand distributed algorithm for multi-hop packet radio networks known as Weighted Clustering Algorithm (WCA). The proposed weight-based distributed clustering algorithm takes into consideration the ideal degree, transmission power, mobility, and battery power of mobile nodes.

(Chatterjee *et al.,* 2002) used the non-periodic procedure for cluster head election on-demand, to reduce the computation and communication costs. Their algorithm is adaptively invoked based on the mobility of the nodes; they try to delay the cluster head election procedure as long as possible.

In WCA load balancing achieved by specifying a pre-defined threshold on the number of the nodes that a cluster head can handle ideally. Cluster heads in their scheme works in dual power mode to obtain connectivity, the cluster heads can operate at a higher power mode (*resulting in a higher transmission range*) for inter-cluster communication while they use lower power for intra-cluster communication.

The main philosophy in (Chatterjee *et al.,* 2002) is choosing an optimal number of cluster heads, which will yield high throughput but incur as low latency as possible by using a combined weight metric. There is several system parameters, depending on specific applications, any or all of those parameters can be used in the metric to elect the cluster heads. When the elected cluster head moved fast, the nodes may be detached from the cluster head and a result, a reaffiliation occurs. Reaffiliation takes place when one of the ordinary nodes moves out of a cluster and joins another existing cluster.

WCA works at system activation time, where every node broadcasts its ID. It is assumed that a node receiving a broadcast from another node can estimate their mutual distance from the strength of the signal received, or through GPS. Once the neighbors list for each node is ready, the clustering algorithm chooses the cluster head for the first time. Each node then maintains its status (*i.e., ORDINARY, GATEWAY*).

Due to the dynamic nature of the ad hoc networks, the nodes as well as the cluster heads tend to move in different directions, thus disorganizing the stability of the configured system. So, (Chatterjee *et al.,* 2002) used an update policy to handle this situation. The update may result in formation of new clusters and possible change of point of attachment of nodes from one cluster head to another one.

The update policy works as follows, when the mutual separation between the node and its cluster head increases, the signal strength decreases. In that case, the mobile has to notify its current cluster head that it is no longer able to attach itself to that cluster head. The cluster head tries to hand-over the node to a neighboring cluster. If the node goes into a region not covered by any cluster head, then the cluster head election algorithm is invoked and the new dominant set (*The set of cluster heads*) is obtained.

In WCA, the main drawback is the invoking request done at the update policy. Moving of one node caused the whole network to re-calculate the weights and changing the set of cluster heads, even if there are stable clusters. It is obvious that a restriction on the effect and scope of change is required.

3. Multi-Aware Cluster Head Maintenance

3.1 Overview

In this chapter we will explore in details our proposed method. Namely, Multi-Aware Cluster Head Maintenance (MACHM), which aims to reduce the total amount of the power consumed by the nodes in the network. Especially, cluster heads, which are more sensitive to power drains for their roles in the network.

Like WCA, our method will involve in cluster head election, cluster formation stage, and cluster head re-election procedures. To be more efficient, many factors will be considered in all stages, like: the ideal number of nodes that a cluster can handle (*Load Balancing Consideration*), the distance between the node and its neighbors (*Geographical Consideration*), the speed of nodes (*Mobility Consideration*), and most importantly the node's battery power (*Energy Consideration*).

Cluster head election, cluster formation, and cluster head re-election are critical functions in any Cluster-Based Routing Protocol. This is because they will affect the whole performance of the network and its functionality, so the previous factors will be used in a re-combined weighted formula. This formula will give us a more powerful indication about node's status.

Nodes will be assigned weight values, and their behavior will depend on it. These values will help us to take different decisions and actions throughout the network lifetime as will be shown in next sections.

3.2 Some Features of MACHM

Like WCA, our scheme will invoke cluster head election and clusters formation at the time of system activation, but unlike WCA not all the nodes will participate in the cluster head election. The decision of participating or not will depend on the initial node's battery power value as we will see later.

To ensure the correct collection of neighborhood information, HELLO messages will be exchanged periodically between nodes. In order to establish stable clusters different factors will be involved.

Every node will calculate its degree, which is defined as the number of node's direct neighbors. Then degree difference will be computed as the node's degree minus a pre-defined threshold on the number of nodes that a cluster can handle ideally. WCA use this to achieve load balancing throughout the network.

Distances between node and its neighbors are very important, because it will give us a hint about the central node, which has less summation of distances with its neighbors. This is desired because we want the cluster head to be the nearest to the core of the cluster. The ideas of self positioning will be used to determine the set of first level neighbors, then determining their coordinates, which will enable calculating the required distances.

Mobility is great challenge in ad hoc environment, which affects node's location and neighborhood information essential for clustering, so we try to consider it by measuring the running average of the speed for every node.

In MACHM our main focus will be on node's battery power. The way we deal with battery power factor is different; we consider this factor by using the two levels with the threshold shown in Figure 3.1.

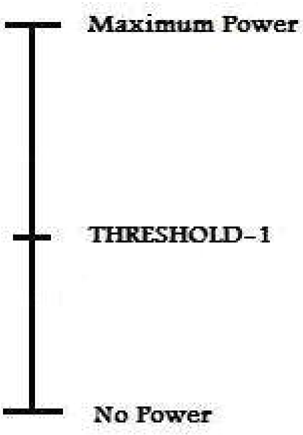


Figure 3.1: Battery Power Levels

MACHM is not limited to consider the consumed battery power as an indicator, but also checks the initial and current battery power values, to take more precise decisions.

Every node will then be able to calculate its weight value by using this weighted formula:

*Node's Total Weight = Weight-1 * Factor-1 + Weight-2 * Factor-2 + Weight-3 * Factor-3 + Weight-4 * Factor-4* (3.1)

Where:

Weight-1, *Weight-2, Weight-3, and Weight-4*:	Constant values
Factor-1, *Factor-2, Factor-3, and Factor-4*:	The considered parameters

The derivation and formal description of Formula (3.1) will be given later in this chapter. The previous calculated value will be exchanged to determine the cluster heads at system initialization time, then later in the cluster head re-election procedure. While the elected cluster head's battery power level is over THRESHOLD-1 nothing is needed to be done. When cluster head's battery power level going below THRESHOLD-1, it will ask direct neighbors to calculate their weights again, to reduce the scope of search.

The choice of the candidate cluster head will depends on the previous weighted formula. After this step, the candidate cluster head will take responsibilities of the original one and the original node will not be the cluster head any more, unless if the current cluster head is better than its neighbors after the invoking request processed. The whole procedure will be applied again to the new cluster head.

3.3 MACHM Control Messages

In order to implement the previously described method, nodes needed to gather different pieces of information relevant to their neighbors. This collection can be done by exchanging control messages, which are broadcasted periodically or evently.

Every control message intends to provide certain piece of knowledge or invoking certain action. In our case, we have to exchange different types of these control messages to enable nodes to work probably. Next subsections will discuss in details the purpose and the structure of the control messages we used.

3.3.1 Hello Message

One of the important knowledge nodes needed to know is their direct neighbors. Nodes use this information to obtain their connectivity, but node's mobility causes frequent changes to node's neighbors. HELLO messages offer the way to overcome this challenge.

The easiest way for nodes to discover their neighbors is through HELLO messages; they are only delivered to the immediate neighbors and are not passed any further. HELLO messages are usually simple and light, the reason is we need to send them in close intervals, without heavily loading the network. Table 3.1 shows the structure of HELLO message.

Table 3.1: HELLO Message Fields

HELLO Message
Type
Source_Node_ID
Cluster_ID

Type field helps the receiving node to determine the kind of the control message, which is in this case a HELLO message, while Source_Node_ID field contains the identifier of the sending node, usually an IP-Address of the initiator node. Cluster_ID field gives the identifier of the cluster which the node belongs to; usually this is the id of the cluster head node.

3.3.2 Weight Message

As we mentioned before, the result of Formula (3.1) is used in cluster head election and cluster head re-election procedures. That requires neighboring nodes know this value to act based on it. This is the roll of WEIGHT messages.

After calculating node's weight each one can exchange its weight value through this control message, the structure and fields of WEIGHT message are shown in Table 3.2.

Table 3.2: WEIGHT Message Fields

WEIGHT Message
Type
Source_Node_ID
Node_Weight
Sequence_Number

Type field tells the receiving node the kind of the control message, which is WEIGHT message in this case. Source_Node_ID field contains the identifier of the control message initiator. Node_Weight field contains the calculated weight value that we need to share with neighbors.

Sequence_Number field is very useful; it can be used to determine the newest versions of WEIGHT messages, and to recognize different WEIGHT messages send by the same node. The combination of Source_Node_ID and Sequence_Number fields is unique.

3.3.3 Adjacent Cluster Head Message

In cluster-based techniques there are two types of routing that can be used separately depending on the cluster level. Proactive techniques can be used inside the cluster when the number of cluster members is relatively small, this keeps the routing done locally and fast, but when the destination is outside the cluster reactive techniques can be used.

Table 3.3: ADJACENT_CLUSTER_HEAD Message Fields

ADJACENT_CLUSTER_HEAD Message
Type
Source_Node_ID
Cluster_Head_ID
Adjacent_Cluster_Head_ID
Sequence_Number

That requires the cluster head where the source, knows other cluster head nodes to be able to forward the routing request to them. This can be accomplished by using ADJACENT_CLUSTER_HEAD messages send by gateway nodes (*Nodes that* *have neighbors from different clusters*) to their cluster head. The structure of ADJACENT_ CLUSTER_HEAD message was shown in Table 3.3.

As mentioned before, type field used to recognize the kind of control message received, Source_Node_ID tells the identifier of the sending node, and this can be used by the cluster head to change the sending node's status to gateway if it is not. Cluster_Head_ID field represents the identifier of the destination node we intend to send the information to.

Adjacent_Cluster_Head_ID field contains the identifier of the neighboring cluster learned by the gateway node; cluster head use this knowledge to recognize the adjacent clusters when it needs to request them. Finally, Sequence_Number field helps the cluster head to manage different ADJACENT_CLUSTER_HEAD messages send by the same gateway node, because again the combination of Source_Node_ID and Sequence_Number fields is unique.

3.3.4 Invoke Request Message

In MACHM when the current cluster head's battery power level goes below THRESHOLD-1, cluster head needs to look for a new candidate cluster head that will take its responsibility.

To have a better choice, the candidate will be from the first-level neighbors set of the current cluster head, then the neighbor with the lowest weight value will be the replacement cluster head. INVOKE_REQUEST message is used by the cluster head for this purpose; this control message informs direct neighbors that a re-calculation of their weights is needed. Table 3.4 shows the structure of INVOKE_REQUEST message.

Table 3.4: INVOKE_REQUEST Message Fields

INVOKE_REQUEST Message
Type
Source_Node_Id
Sequence_Number

Type field is used to determine the kind of control message, which in this case is INVOKE_REQUEST message. Source_Node_ID gives the identifier of the request message initiator, receivers of this control message can use this field for checking purposes before responding, because it should contains the cluster head's identifier. Sequence_Number field helps the receivers to distinguish between new and old requests.

3.4 MACHM Tables

After exploring the types of control messages we used in MACHM, we will describe the complementary elements in this section, which are tables. Every control message provides a specific knowledge, to enable nodes tracing this knowledge; they will keep it in a data structure that can be accessed rapidly.

Usually, we use the term table when we want to represent this data structure. The objective and role of every table along with its structure and fields will be described in the next subsections.

3.4.1 Neighbors Table

When receiving the HELLO message described before, node wants to store the learned information. Therefore, it will use the NEIGHBORS table for this mission. Every entry in this table will have a timer assigned to it, when this timer expires before updating the entry, the corresponding entry is removed from the NEIGHBORS table, where Table 3.5 shows its components.

Table 3.5: NEIGHBORS Table Fields

NEIGHBORS Table
Neighbor_ID
Neighbor_Cluster_ID
Neighbor_Distance
Time_Of_Recieve

Neighbor_ID field contains the identifier of the received HELLO message originator, while Neighbor_Cluster_ID label the cluster identifier of this neighbor. Neighbor_Distance field contains the calculated value by the receiving node of the distance difference between itself and its neighbor; this can be done by using the ToA technique, because radio signals travel with known velocity. Time_Of_Recieve field can be used by the node along with a timer to check entries validity.

3.4.2 Adjacent Cluster Heads Table

This table gets its entries from ADJACENT_CLUSTER_HEAD control message discussed previously; only cluster heads need to maintain this table. They can use it when they want to communicate with other clusters for any reason. Table 3.6 shows its structure.

Table 3.6: ADJACENT_CLUSTER_HEADS Table Fields

ADJACENT_CLUSTER_HEADS Table
Adjacent_Cluster_Head_ID
Gateway_Node_ID
Time_Of_Recieve

Adjacent_Cluster_Head_ID field contains the identifier of the neighbor's cluster head extracted from ADJACENT_CLUSTER_HEAD control message. Where the Gateway_Node_ID field helps the cluster head to know the require node to communicate with the required neighbor cluster. Again, Time_Of_Recieve field can be used by the cluster head along with a timer to check entries validity.

3.4.3 Received Packets Table

As we noticed, there is variety of packets sent around. For more efficient handling of these packets, nodes can use RECEIVED_PACKETS table to monitor them. Using this style node can tell the difference between new and duplicate received messages. The structure and components of RECEIVED_PACKETS table are presented in Table 3.7.

Table 3.7: RECEIVED_PACKETS Table Fields

RECEIVED_PACKETS Table
Source_Node_ID
Sequence_Number
Time_Of_Receive

Source_Node_ID and Sequence_Number combination produce the appropriate uniqueness, which can be used by the node to mark the different received packets. In this table the use of Time_Of_Receive field is relatively different; we mainly use this field to shorten the table after enough periods to avoid huge expanding in RECEIVED_PACKETS table size.

3.5 MACHM Parameters

From the previous discussion, we concluded that MACHM involved in different missions and tasks. Every aspect of the method needs specific parameter or argument to control its behavior. This raises the need for MACHM to adjust many parameters, to get the most possible suitable performance of the method.

In this section, we will survey the parameters we used and their goal, where in the next chapter, the chosen values of these parameters will be presented. Table 3.8 shows MACHM parameters.

Table 3.8: MULTI-AWARE CLUSTER HEAD MAINTENANCE Parameters

MULTI-AWARE CLUSTER HEAD MAINTENANCE PARAMETERS
Neighbors_Table_Timer
Adjacent_Cluster_Heads_Table_Timer
Received_Packets_Table_Timer
Maximum_Sequence_Number
HELLO_Message_Timer
Cluster_Head_Handling
Weight_1
Weight_2
Weight_3
Weight_4
Battery_Power_Threshold_1

Neighbors_Table_Timer, Adjacent_Cluster_Heads_Table_Timer, and Received_Packets_Table_Timer parameters defines the maximum time a row or entry in the NEIGHBORS, ADJACENT_CLUSTER_HEADS, and RECEIVED_PACKETS tables can resides without being reset if no update received for this entry.

Maximum_Sequence_Number parameter determines the maximum sequential number for packets unique identifier nodes can assign to their sent packets. HELLO_Message_Timer parameter defines the time interval a node will send HELLO messages to neighbors, in order to discover them and maintain its connectivity. While Cluster_Head_Handling parameter used to define the assumed number of nodes that a cluster head can handle efficiently; we use this parameter to adopt load balancing property.

Weight_1 parameter used to give a percentage of total weight to node's degree difference factor, while Weight_2 parameter does the same for the node's summation of distances with all its neighbors.Weight_3 parameter used to give heavy to the running average of the node's speed, which measures the mobility, and finally, Weight_4 parameter used with node's battery power factor.

Battery_Power_Threshold_1 parameter used to decide when a node can continue acting as a cluster head (*Over Battery_Power_Threshold_1*), or invoking for a cluster head re-election (*Below Battery_Power_Threshold_1*), then handover to the new cluster head.

3.6 Operational Description of MACHM

In this section, operational description and pseudo code for our previous method will be given, which will cover all the actions that will take place in the different situations and conditions discussed before.

WCA assumed that all nodes have the same battery power values at system activation time, which is not applicable in real life situations. However, MACHM assumes that all nodes will start with different battery power values assigned to them.

At system invocation time, every node will test its initial battery power value against Battery_Power_Threshold_1. If its value over this threshold, ALLOW flag will be set to 1, we use this flag to determine the nodes that are allowed to participate in the cluster head election and cluster head re-election procedures, otherwise it will be set to 0, as shown in Algorithm 3.1.

```
    if (hierarchyStats->InitialBatteryPower > BPT1)
    {
    hierarchyStats->Allow = 1;
    #ifdef DEBUG
      printf("Node allowed to participate in cluster head election and re-
election\n\n");
    #endif
    }
    else
    {
    hierarchyStats->Allow = 0;
    #ifdef DEBUG
      printf("Node not allowed to participate in cluster head election and re-
election\n\n");
    #endif
    }
```

Algorithm 3.1: ALLOW Flag Assignment

Next step, every node allowed to participate in cluster head election will calculate its weight by using the previously presented Formula (3.1). To do this, every factor should be calculated first. To start with, node will use its NEIGHBORS table to determine its degree D_n (*The number of neighbors*) as:

$$D_n = | Neighbors(n) | \qquad (3.2)$$

Where:

D_n:	The degree of node *n*.
Neighbors(*n*):	The number of node *n* neighbors.
$\|\quad\|$:	The absolute value.

As (Chatterjee *et al.,* 2002), every node will compute its degree difference Δ_n, which defined as node's degree minus a predefined threshold α, representing the ideal number of nodes that a cluster head can handle ideally.

$$\Delta_n = | D_n - \alpha | \qquad (3.3)$$

Where:

Δ_n:	The degree difference of node *n*.
D_n:	The degree of node *n*.
α:	The predefined threshold.
$\|\quad\|$:	The absolute value.

Next, every node will compute the sum of the distances L_n, with all its neighbors, this gives an indication about stable nodes that are central to its neighbors.

$$L_n = \sum \{Distance\ (n, n')\} \qquad (3.4)$$

Where:

L_n:	The sum of distances of node *n* with all its neighbors.
$\sum$:	The summation.
$Distance(n, n')$:	The distance between node *n* and its neighbor node *n'*.

To consider mobility factor, every node will compute the running average of the speed for it until the current time of the simulation as calculated in (Chatterjee *et al.*, 2002):

$$M_n = (\sum \surd\, (X_t - X_{t-1})^2 + (Y_t - Y_{t-1})^2) * 1 / T \qquad (3.5)$$

Where:

M_n:	The average speed of node *n*.
$\sum$:	The summation.
$\surd$:	The square root.
X_t:	The *x* coordinates in time *t*.
X_{t-1}:	The *x* coordinates in the previous time slot *t-1*.
Y_t:	The *y* coordinates in time *t*.
Y_{t-1}:	The *y* coordinates in the previous time slot *t-1*.
T:	The current elapsed time of simulation.

Unlike WCA, we use different battery power factor concept. Every node will indicate its current battery power value *Pn*, as:

$$P_n = Initial(P_n) - Consumed(P_n, T) \qquad (3.6)$$

Where:

P_n:	The current battery power value of node *n*.
$Initial(P_n)$:	The initial battery power value of node *n*.
$Consumed(P_n, T)$:	The consumed battery power of node *n* until simulation time *T*.

Finally, every node has all the needed factors to calculate its combined weight W_n. Algorithm 3.2 shows the whole actions and steps taken to calculate node's weights as:

$$W_n = w1 * \Delta_n + w2 * L_n + w3 * M_n + w4 * P_n \qquad (3.7)$$

Where:

W_n:	The combined weight of node *n*.
Δ_n:	The degree difference of node *n*.
L_n:	The sum of distances of node *n* with all its neighbors.
M_n:	The average speed of node *n*.
P_n:	The current battery power value of node *n*.
$w1, w2, w3, w4$:	Constant values.

```
if (hierarchyStats->Allow = = 1)
{
Dn = hierarchyTables->NEIGHBORS_Table->size;
Degree_Difference = abs (Dn - ALPHA);
current = hierarchyTables->NEIGHBORS_Table->head;
                while (current != NULL)
                {
                  Ln += current->Neighbor_Distance;

                  previous = current;
                  current = current->next;
                }
Mn = (sqrt (SQUARE(node->position.x - node->serviceData.prevPosition.x)
         +
                SQUARE(node->position.y - node->serviceData.prevPosition.y)) )
        / Simulation_Time;
Pn = hierarchyStats->BatteryPower;
Wn = W1 * Degree_Difference + W2 * Ln + W3 * Mn + W4 * Pn;
}
```

Algorithm 3.2: Node's Combined Weight Calculation

After wining the election, the chosen cluster head will do its usual functions besides monitoring its battery power level regularly. While it is battery power value is over Battery_Power_Threshold_1 nothing is needed to be done, and the node can keep running as cluster head. WCA considered node's mobility as the triggering action, while in MACHM we go with the node's battery power factor as the triggering action.

When the node's battery power level reaches Battery_Power_Threshold_1 or below a trigger will be activated to ask direct neighbors for a cluster head re-election procedure. The re-election procedure is the same, but this time its scope will be limited to the direct neighbors of the current cluster head, to keep the changes as minimum as possible. Algorithm 3.3 shows this action.

```
if (node->serviceData.clusterElement = = CLUSTERHEAD)
   {
 /* Check if the cluster head's battery power is over THRESHOLD-1 or below */

    if ((hierarchyStats->BatteryPower <= BPT1) &&
         (hierarchyStats->BatteryPower > 0))
                {
                #ifdef DEBUG
                    printf("Cluster Head Node %d: Going Below Battery Power
THRESHOLD-1  %d \n", node->nodeAddr, hierarchyStats->BatteryPower);
                #endif

                /* Invoke request */

                InvokeClusterHeadAlgotforNode(node);

                seqNum = node->serviceData.seqNum++ % MAXSEQNUM;
                InsertPacketTable(InvokeReqTab, node->nodeAddr, seqNum);
                }
    }
```

Algorithm 3.3: Cluster Head Re-election Trigger

This cycle will be repeated for the new cluster head as discussed before. In next chapter we will show the experimental results for our method.

4. Results and Analysis

4.1 Introduction

In this chapter, we will present in details the simulation process we followed. To start with, an overview of the network simulator used in our experiments will be given. Next, different scenarios will be discussed along with their simulation setup parameters. To check the performance of our method, certain metrics should be chosen to give us the right indication about our goals, these metrics will be explored. Finally, simulation results along with their analysis will be given.

4.2 Network Simulator

MACHM and WCA have been simulated using GloMoSim network simulator under UNIX platform. Global Mobile Information System Simulator (GloMoSim) is a scalable simulation environment for large wireless and wired communication networks (Gerla *et al*., 1999). GloMoSim uses a parallel discrete-event simulation capability provided by Parsec.

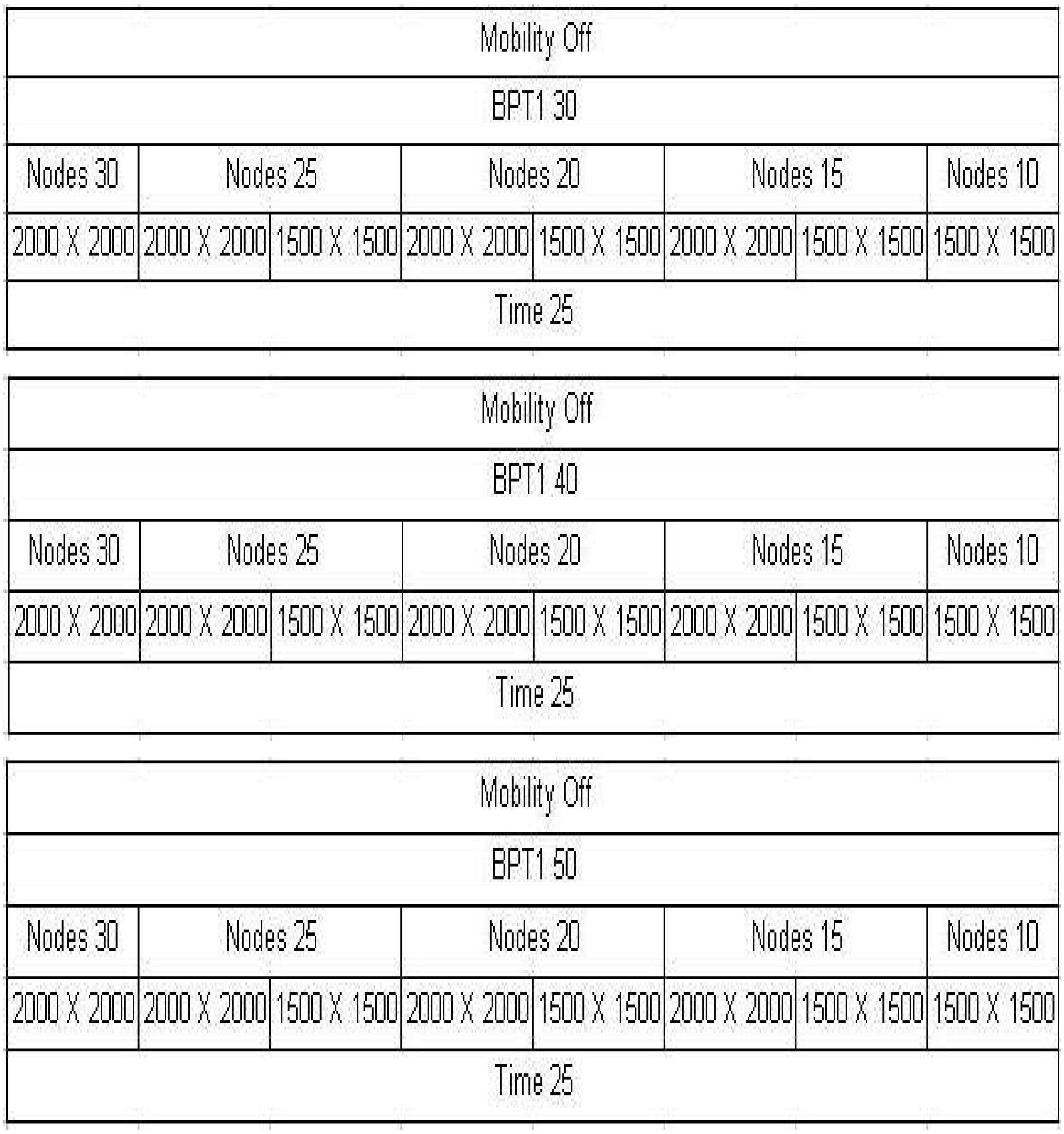
GloMoSim is designed in a layered approach, it can simulate networks with up to thousand nodes linked by a heterogeneous communications capability that includes multicast, asymmetric communications using direct satellite broadcasts, multi-hop wireless communications using ad-hoc networking, and traditional Internet protocols (Gerla *et al*., 1999). GloMoSim provides simulation statistics at all layers, which helped us to trace the behavior of both methods at different layers and levels. Further information about the simulator can be found in Appendix A.

We ran the simulations on a machine having the following capabilities, Fedora™ Core Release 5 Operating System, Intel® Core™ Duo Central Processing Unit, T2450 @ 2.00GHz, 0.99 GB of Random Access Memory, and 120 GB Hard Disk.

4.3 Scenarios and Simulation Setup Parameters

We divided the simulation study sample into two groups; this helped us to trace different scenarios and conditions. The first group of simulations done when the mobility is off whiles the other group tested when the mobility is on. Inside each group we divide the simulations depending on the BPT1 value. For every BPT1 value we tested different number of nodes in different sizes of simulation area within 25 minutes simulation time. Tables 4.1 and 4.2 summarize the simulation study sample groups.

Table 4.1: Simulation Study Sample Group1

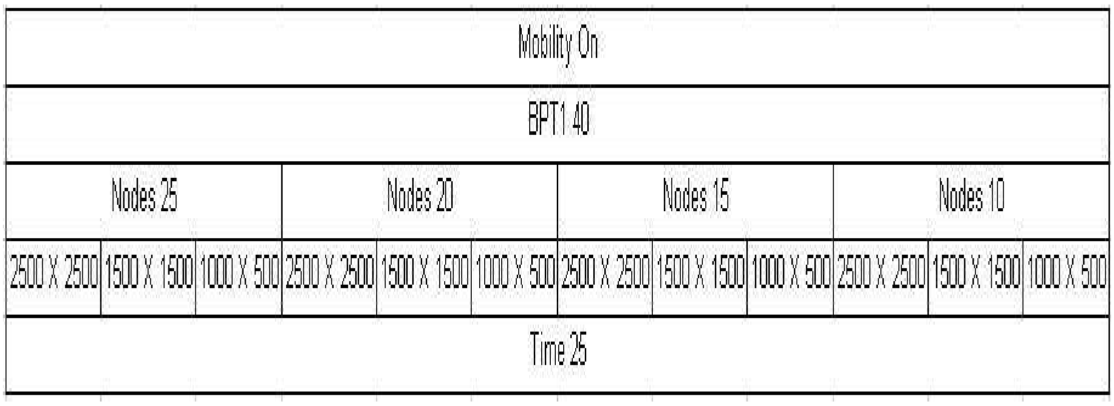


Mobility Off							
BPT1 30							
Nodes 30	Nodes 25		Nodes 20		Nodes 15		Nodes 10
2000 X 2000	2000 X 2000	1500 X 1500	2000 X 2000	1500 X 1500	2000 X 2000	1500 X 1500	1500 X 1500
Time 25							

Mobility Off							
BPT1 40							
Nodes 30	Nodes 25		Nodes 20		Nodes 15		Nodes 10
2000 X 2000	2000 X 2000	1500 X 1500	2000 X 2000	1500 X 1500	2000 X 2000	1500 X 1500	1500 X 1500
Time 25							

Mobility Off							
BPT1 50							
Nodes 30	Nodes 25		Nodes 20		Nodes 15		Nodes 10
2000 X 2000	2000 X 2000	1500 X 1500	2000 X 2000	1500 X 1500	2000 X 2000	1500 X 1500	1500 X 1500
Time 25							

Table 4.2: Simulation Study Sample Group2



Mobility On											
BPT1 40											
Nodes 25			Nodes 20			Nodes 15			Nodes 10		
2500 X 2500	1500 X 1500	1000 X 500	2500 X 2500	1500 X 1500	1000 X 500	2500 X 2500	1500 X 1500	1000 X 500	2500 X 2500	1500 X 1500	1000 X 500
Time 25											

For both MACHM and WCA, we use different simulation setup parameters for each simulation study sample group, including: time interval between HELLO messages, maximum time allowed for entries to remain in NEIGHBORS or ADJACENT_CLUSTER_HEADS or RECEIVED_PACKETS tables without being updated, maximum sequential number to be used in packets, number of nodes that a cluster head can handle ideally, the combined weights W1, W2, W3, and W4, and finally, the mobility model used. Tables 4.3 and 4.4 show these parameters for each group.

Table 4.3: Simulation Setup Parameters for Group1

Parameter	Value
HELLOTIME	**250 ms**
NEIGHBORTABLETIMER	**500 ms**
CLUSTERHEADTABLETIMER	**1000 ms**
PACKETLIFETIME	**5 s**
MAXSEQNUM	**50000**
ALPHA	**5**
W1	**0.2**
W2	**0.2**
W3	**0.1**
W4	**0.5**
MOBILITY	**None**

Table 4.4: Simulation Setup Parameters for Group2

Parameter	Value
HELLOTIME	**250 ms**
NEIGHBORTABLETIMER	**500 ms**
CLUSTERHEADTABLETIMER	**1000 ms**
PACKETLIFETIME	**5 s**
MAXSEQNUM	**50000**
ALPHA	**5**
W1	**0.1**
W2	**0.1**
W3	**0.4**
W4	**0.4**
MOBILITY	**RANDOM-WAYPOINT**
PAUSE	**30 s**
MINSPEED	**0 m/s**
MAXSPEED	**10 m/s**

In order to get consistent results, we used different weights for each simulation study sample group. When the mobility is off we gave a higher weight to the battery power metric (*W4*) equaled to 50%, while giving the mobility metric (*W3*) lower weight equaled to 10%. This is because the effect of nodes movement is absent in this group, but when the mobility is on we set equal weights to battery power and mobility metrics (*W4 and W3 respectively*) equaled to 40% to be fair as possible.

The mobility model used is random waypoint, which works as follows: a node randomly selects a destination from the physical terrain. It moves in the direction of the destination in a speed uniformly chosen between MINSPEED and MAXSPEED (meter/sec). After it reaches its destination, the node stays there for PAUSE time period. At the application layer, TELNET has been used as the data messages generator.

4.4 Performance Metrics

In this section, we will show the performance metrics we used in our simulation, to compare the performance of MACHM against WCA. The choice of performance metrics considers the goal which we aim to achieve. Next subsections will give a brief description of each metric and the purpose of using it.

4.4.1 The Number of Cluster Heads

This metric measures the number of cluster heads that exist in the network, the purpose of this metric is to discover the relation between the number of existing cluster heads and the power consumed in the network, under different conditions relating to the number of nodes, simulation area size, and the time of simulation.

4.4.2 The Number of Invoke Requests

Overhead can be measured in term of control messages occurred in the network like the INVOKE_REQUEST message. Both MACHM and WCA used the concept of invoke requesting, but the choice of when to do this invoking is different. MACHM activate the invoke request based on battery power factor, while WCA do so on mobility factor. We used the number of invoke requests metric to compare the two choices, and study the effect of it on the overhead and power consumed in the network.

4.4.3 The Percentage of Cluster Heads Consumption

Saving cluster heads battery power represented our main concerns, because it is an essential property, which will extend the life time of the network. In order to study the efficiency of dealing with cluster heads battery power, we will measure the percentage of cluster heads power consumed by each method under different conditions.

4.4.4 The Percentage of All Nodes Consumption

While cluster heads are our main interest, but also the rest of nodes shouldn't be loaded too much, to ensure longer time of operations. We will measure the percentage of all nodes power consumed in the network, to compare the performance of both methods regarding this property.

4.4.5 The Throughput of Data Messages

Throughput defined as the average rate of successful message delivery over a communication channel (*Wireless channel in our case*). The throughput is usually measured in bits per second (*bit/s or bps*), and sometimes in data packets per second or data packets per timeslot (Rappaport, 2002). This metric used to measure the efficiency of both methods in delivering data messages within network.

4.5 Results and Analysis

We start by testing the number of cluster heads MACHM produced against WCA. Figures 4.1 and 4.2 show the results for 1500 X 1500 and 2000 X 2000 simulation area size respectively, when simulated for a 25 minutes while the mobility is off.

Simulation results showed that MACHM achieved less number of cluster heads than WCA, under different conditions. When the simulation area is 1500 X 1500, and the BPT1 equaled to 30, the difference in the number of cluster heads for both methods ranged from 0 (*When the number of nodes is 15*) to 2 (*When the number of nodes is 10 and 25*). The difference extended when the BPT1 equaled to 50, from 2 (*When the number of nodes is 10 and 15*) to 3 (*When the number of nodes is 20 and 25*).

Similar behavior can be noticed when the simulation area size is 2000 X 2000, but with a wider difference in the number of cluster heads. For BPT1 equaled to 30, the difference ranged from 1 (*When the number of nodes is 20*) to 4 (*When the number of nodes is 25 and 30*). When the BPT1 equaled to 50, a wider difference can be recognized, which ranged from 4 (*When the number of nodes is 15 and 20*) to 6 (*When the number of nodes is 25*).

In WCA, we noticed that the number of cluster heads didn't affected by changing the BPT1 for the same simulation area size and time. When the simulation area size is 1500 X 1500, the number of cluster heads equaled to 4, 5, 7, and 9 from a set of nodes equaled to 10, 15, 20, and 25 respectively. This number of cluster heads kept unchanged while varying the BPT1 from 30 to 40 then 50. Same case appeared for 2000 X 2000 simulation area size, but with higher number of cluster heads equaled to 8, 11, 15, and 17 from a set of 15, 20, 25, and 30 nodes respectively.

The previous results are expected and natural because WCA didn't use BPT1 to control the process of electing its cluster heads. WCA is unaware of this parameter and that caused the fixed number of cluster heads, even when the BPT1 parameter value changed through the same simulation area size.

When the mobility is off, network topology will be unchanged and node's battery power drain caused by exchanging control and data packets only. WCA activate its trigger on node's mobility so the chosen cluster heads will not changed throughout the simulation, on the other hand, MACHM will change the set of cluster heads throughout the simulation time based on their energy level.

In MACHM, BPT1 parameter is involved in the process of electing the cluster heads. This resulted in different number of cluster heads for the same simulation area size when changing the BPT1 value. MACHM will limit the number of nodes that allowed participating in the election process depending on their initial battery power value, while in WCA there is not any kind of such restriction.

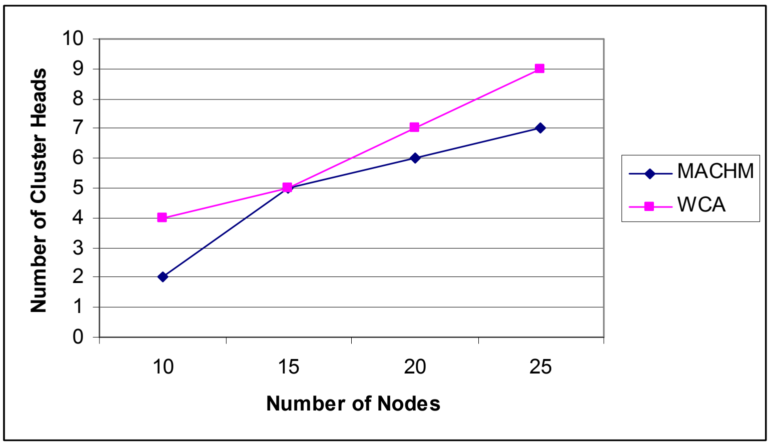


Figure 4.1 (a): BPT1 = 30

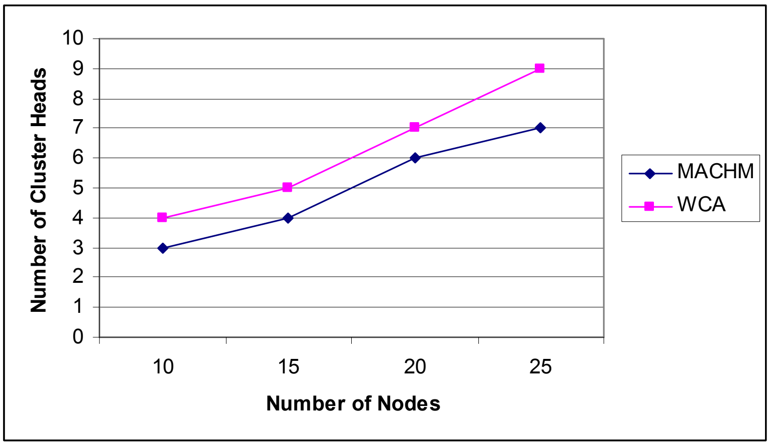


Figure 4.1 (b): BPT1 = 40

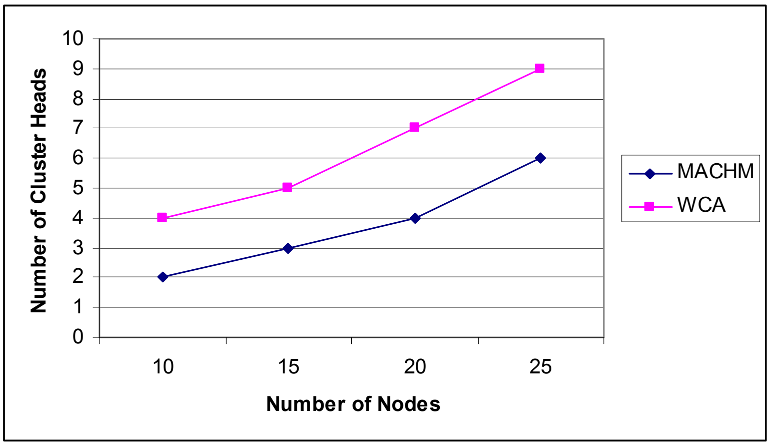


Figure 4.1 (c): BPT1 = 50

Figure 4.1: Number of Cluster Heads for Different BPT1 Values in 1500 X 1500

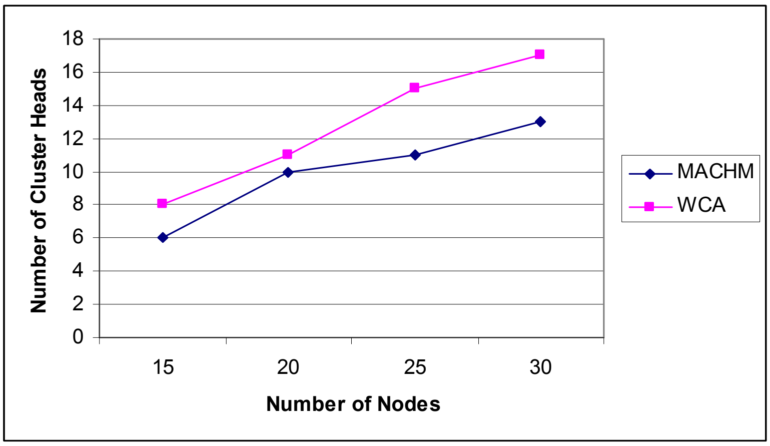


Figure 4.2 (a): BPT1 = 30

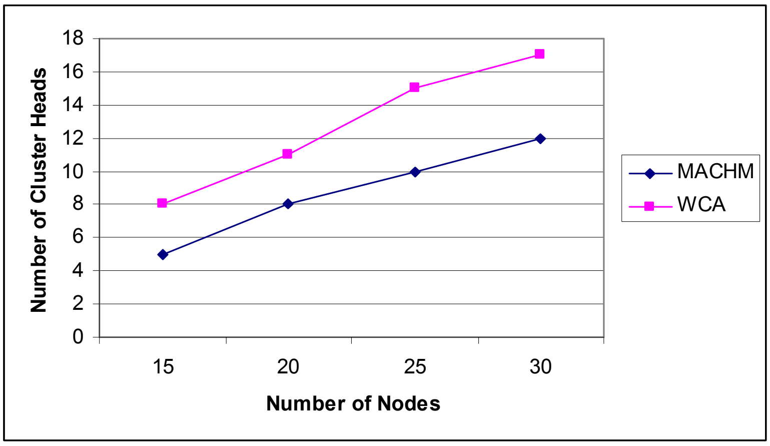


Figure 4.2 (b): BPT1 = 40

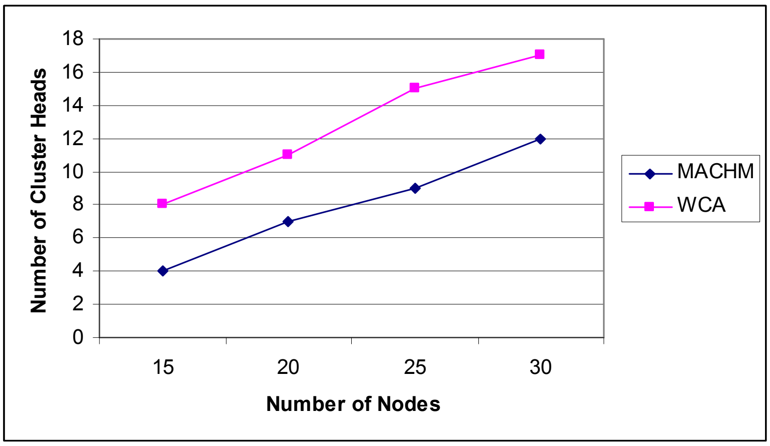


Figure 4.2 (c): BPT1 = 50

Figure 4.2: Number of Cluster Heads for Different BPT1 Values in 2000 X 2000

For MACHM, simulation results showed that increasing the BPT1 value will reduce the number of cluster heads for the same simulation area size. This happened because increasing the BPT1 will decrease the opportunity for nodes to have battery power level values over this parameter. As a result, less number of nodes will enabled their ALLOW flag, which controlled the participating in the election process, and therefore reducing the number of cluster heads in the network.

When the BPT1 equaled to 30 in 1500 X 1500 simulation area size, the number of cluster heads MACHM produced was 2, 5, 6, and 7 from 10, 15, 20, and 25 nodes respectively. This decreased to 2, 3, 4, and 6 for the same number of nodes in the same simulation area size when changing the BPT1 value to 50. Similar results obtained for 2000 X 2000 simulation area size, the number of cluster heads decreased from 6, 10, 11, and 13 for 15, 20, 25, and 30 nodes respectively to 4, 7, 9, and 12.

Another note concerned the simulation area size, increasing the size from 1500 X 1500 to 2000 X 2000 resulted in increasing the number of cluster heads for the same number of nodes at the same BPT1 values, because of node's distribution in larger area. The chance of nodes going to far areas is bigger, which will extend the number of cluster heads to cover this larger area.

As an example on the previous note, notice the number of cluster heads when the BPT1 equaled to 30. The number of cluster heads increased from 5, 6, and 7 for 15, 20, and 25 nodes respectively to 6, 10, and 11 when the simulation area size extended from 1500 X 1500 to 2000 X 2000.

Next, we studied the same metric in the existence of node's mobility. In this case, WCA will change the set of cluster heads throughout the simulation because of nodes movements and separations, which will activate its invoking requests. Figure 4.3 presented the results obtained, which showed that MACHM still achieved less number of cluster heads than WCA.

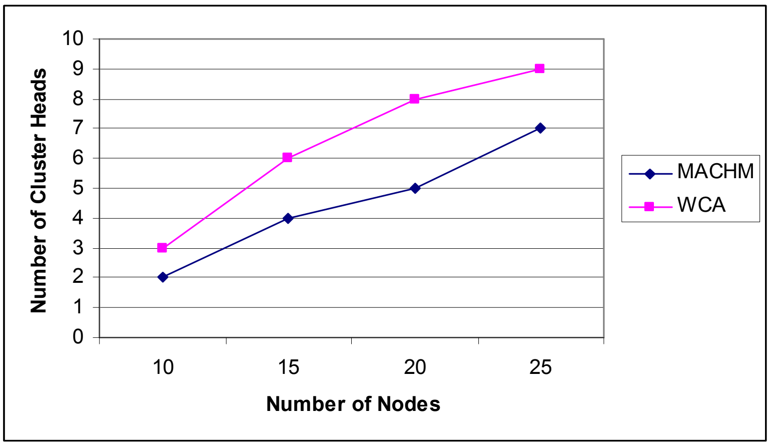


Figure 4.3 (a): Simulation Area = 1000 X 500

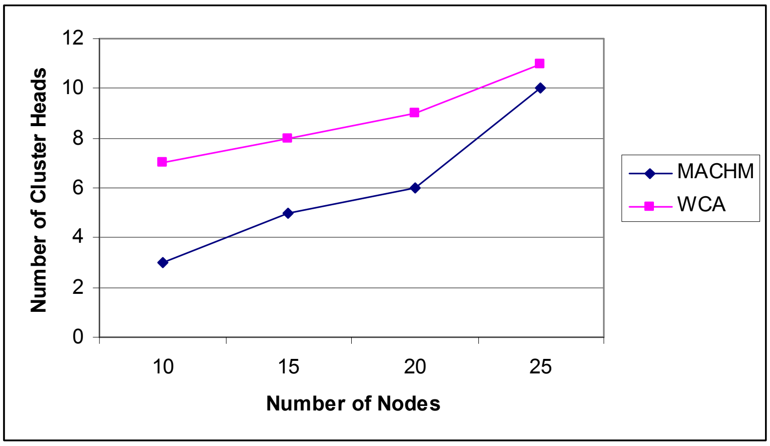


Figure 4.3 (b): Simulation Area = 1500 X 1500

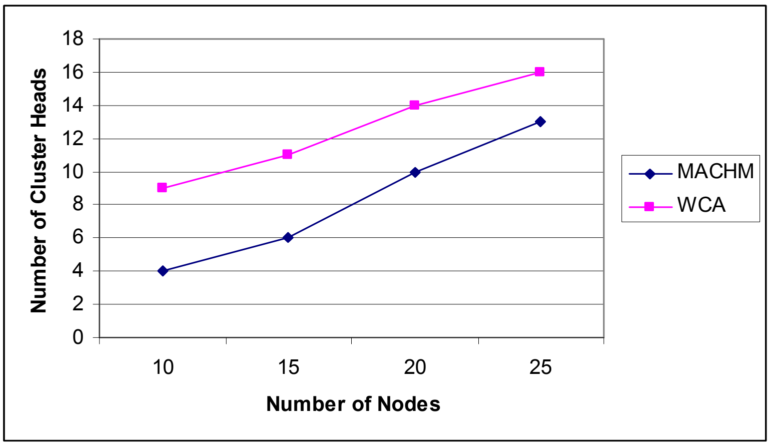


Figure 4.3 (c): Simulation Area = 2500 X 2500

Figure 4.3: Number of Cluster Heads for Different Sizes of Simulation Area

Again, the main reason of getting less number of cluster heads in MACHM, is because of the allow limitation implemented in the method as discussed before. In this group of simulations, we fixed the BPT1 value to 40, while varying the simulation area sizes and the number of nodes in the network. We noticed that the difference in the number of cluster heads increased with the increasing of the simulated area.

For 1000 X 500 simulation area size the difference in the number of cluster heads ranged from 1 (*When the number of nodes is 10*) to 3 (*When the number of nodes is 20*). For 1500 X 1500 simulation area size, wider difference obtained ranged from 1 (*When the number of nodes is 25*) to 4 (*When the number of nodes is 10*). In 2500 X 2500 a wider difference noticed that ranged from 3 (*When the number of nodes is 25*) to 5 (*When the number of nodes is 10 and 15*).

Increasing the size of simulation area resulted in increasing of the number of cluster heads for both methods. This is to enable each method handling the larger area with same number of nodes distributed in. As an example, WCA had 3, 6, 8, and 9 cluster heads for 10, 15, 20, and 25 nodes respectively in 1000 X 500 simulated area. This increased to 7, 8, 9, and 11 cluster heads for 1500 X 1500 simulation area size, then to 9, 11, 14, and 16 when simulated in 2500 X 2500 terrain.

In the case of the number of invoke requests done by each method metric, the only comparison can be done when the mobility is on. This is acceptable because based on WCA philosophy in the case of disabling the mobility, WCA will not do any invoke requests. This will not help us to compare the performance of both methods regarding this metric.

On the other hand, when the mobility is on each method will activate its own invoke requests, and the comparison can be done. Each method adopted different event to activate it's invoke request trigger, WCA on mobility basis and MACHM on battery power basis. Figure 4.4 shows the simulation results for both choices.

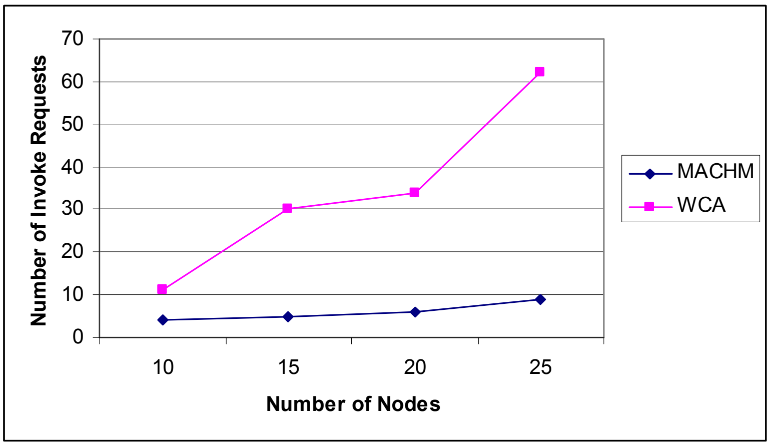


Figure 4.4 (a): Simulation Area = 1000 X 500

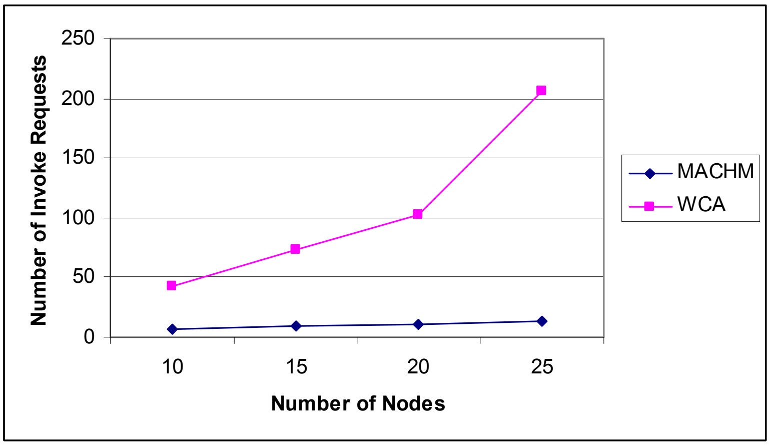


Figure 4.4 (b): Simulation Area = 1500 X 1500

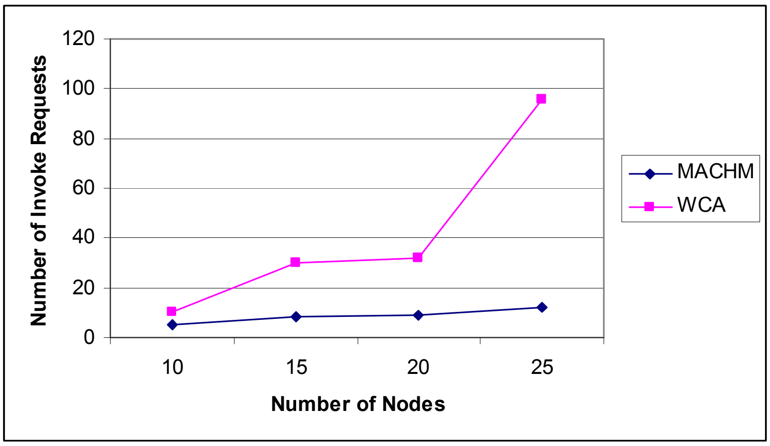


Figure 4.4 (c): Simulation Area = 2500 X 2500

Figure 4.4: Number of Invoke Requests for Different Sizes of Simulation Area

Simulation results showed that MACHM did lower number of invokes requests compared to WCA, at the same simulation area size and number of nodes.

The difference in the number of invoke requests done by each method is obvious at variant conditions. For example, in Figure 4.4 (b) MACHM did 6, 9, 10, and 13 invoke requests for number of nodes equaled to 10, 15, 20 and 25 respectively. At the same setting WCA did 42, 73, 102, and 206 invoke requests, resulting to a difference up to 193 invoke requests, which is relatively huge.

In MANET's environment, the existence of mobility in a wide area caused nodes to move around unpredictably. As a result, frequent topology changes occurred, forcing WCA to activate its update policy represented in the invoke requests. This explained the larger number of invoke requests done by WCA that appeared in the results.

In MACHM the choice of activating the invoke requests on battery power considerations, produced less number of invoke requests in different situations. We can notice that easily, because when the mobility is on, the separation between nodes and their cluster heads will occur more frequently than reaching the BPT1, which happened slowly.

The next group of results considered the percentage of cluster heads battery power consumption. As we mentioned previously, this metric is our main constraint which we aim to reduce as possible. Figures 4.5 and 4.6 show simulation results for 1500 X 1500 and 2000 X 2000 simulation area size respectively, when the mobility is off.

Even when the mobility is off, simulation results showed that MACHM preserved higher percentage of cluster heads battery power than WCA. The difference and improvement of cluster heads battery power consumption varied from scenario to other. In 1500 X 1500 simulation area size when the BPT1 equaled to 30 the range of improvements varied from 22% (*When the number of nodes is 15*) to 41% (*When the* *number of nodes is 20*), where the percentage of battery power consumption is 50% and 43.82% for MACHM, comparing to 63.80% and 73.97%.

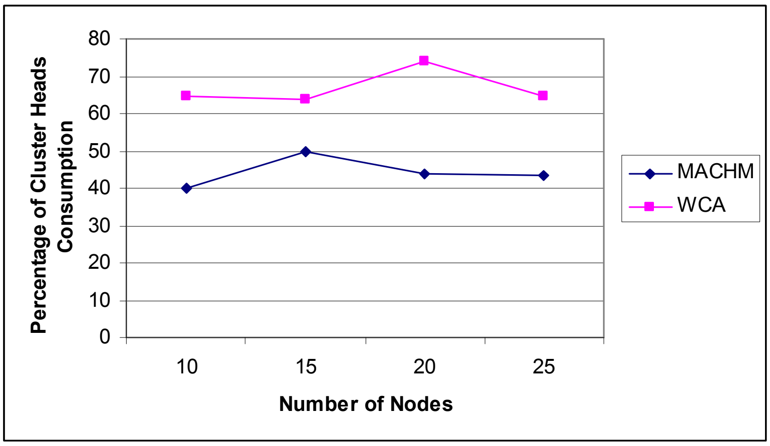


Figure 4.5 (a): BPT1 = 30

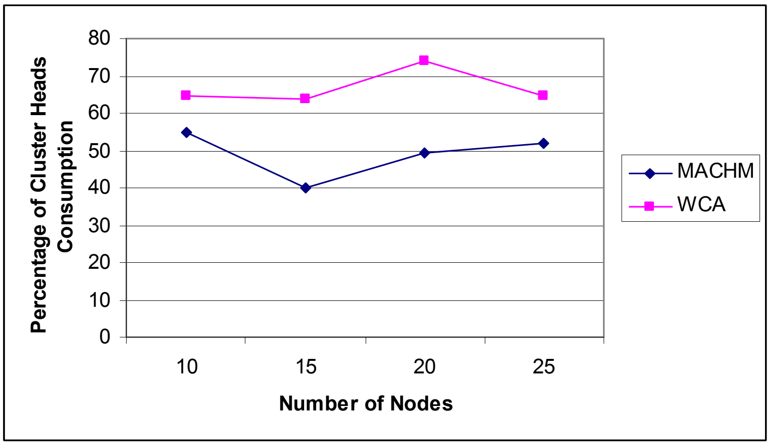


Figure 4.5 (b): BPT1 = 40

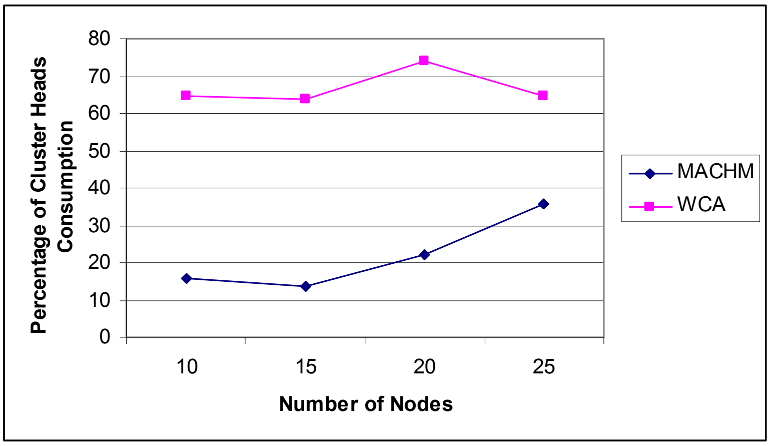


Figure 4.5 (c): BPT1 = 50

Figure 4.5: Percentage of Cluster Heads Consumption for Different BPT1 Values in 1500 X 1500

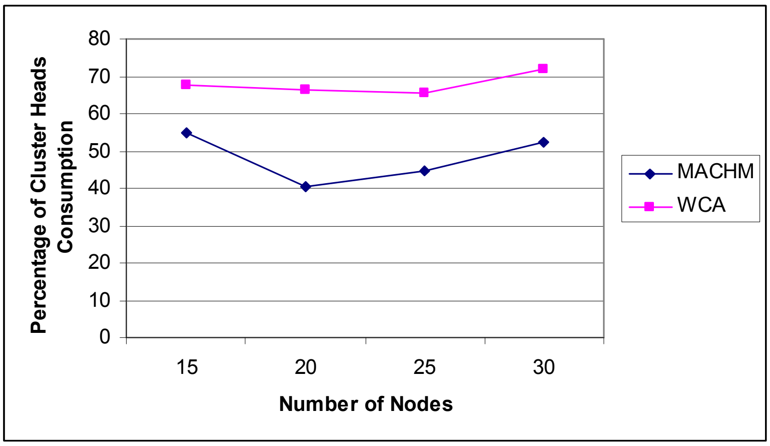


Figure 4.6 (a): BPT1 = 30

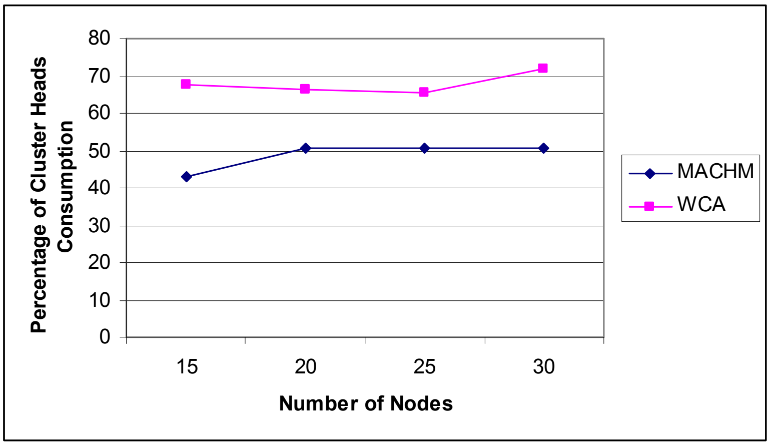


Figure 4.6 (b): BPT1 = 40

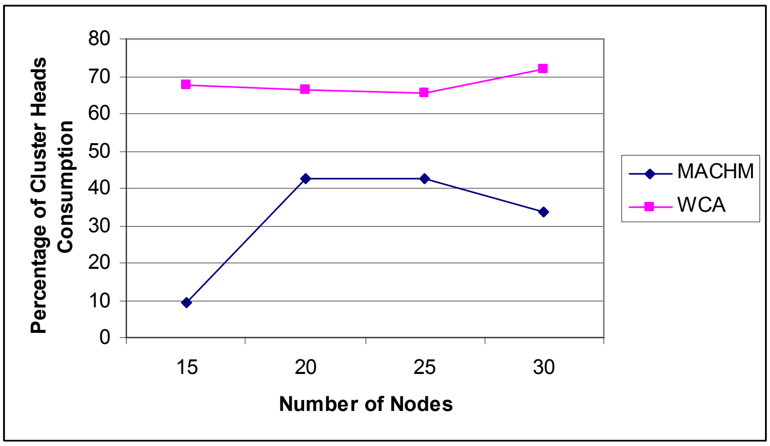


Figure 4.6 (c): BPT1 = 50

Figure 4.6: Percentage of Cluster Heads Consumption for Different BPT1 Values in 2000 X 2000

Better improvement can be noticed in 1500 X 1500 simulation area size when the BPT1 value equaled to 50. MACHM consumed 35.62% (*When the number of* *nodes is 25*) and 13.51% (*When the number of nodes is 15*) of the total cluster heads battery power. Compared to 64.56% (*When the number of nodes is 25*) and 63.80% (*When the number of nodes is 15*) for WCA method, leading to improvements ranged from 45% to 79% respectively.

Similar behavior recognized for 2000 X 2000 simulation area size. When the BPT1 value is 30, MACHM consumed 54.80% (*When the number of nodes is 15*) and 40.52% (*When the number of nodes is 20*) of the total battery power of cluster heads nodes. At the same circumstances, WCA consumed 67.50% and 66.52% of the total cluster heads battery power, resulting to improvements of 19% and 40% respectively.

Wider preservation occurred when the BPT1 value is 50 in 2000 X 2000 simulation area size. WCA consumed 65.47% (*When the number of nodes is 25*) and 67.50% (*When the number of nodes is 15*) compared to 42.60% and 67.50% for MACHM. The improvements in this scenario ranged from 35% to 87% respectively.

These results obtained because MACHM method changed the cluster head when it reaches BPT1 value, prohibiting it from continue acting as cluster head. While in WCA the case is different, WCA is not sensitive to node's battery power. In the absence of mobility and nodes movements none of the elected cluster heads will be changed, causing more battery power drains for these nodes because they operated as cluster heads for longer time.

From the previous simulation results, we can notice that better saving occur on higher BPT1 values. This is happened because at higher BPT1 values, cluster heads will reach the threshold value at less time, causing faster changing of cluster heads, and hence reducing the power consumed. On the other hand, this will increase the number of invoke requests. When minimizing the BPT1 value, more nodes are permitted to engage in the electing process, and the elected cluster heads will keep running longer to reach the lower BPT1.

We studied the percentage of cluster heads battery power consumption in the existence of node's mobility. Figure 4.7 shows the results obtained.

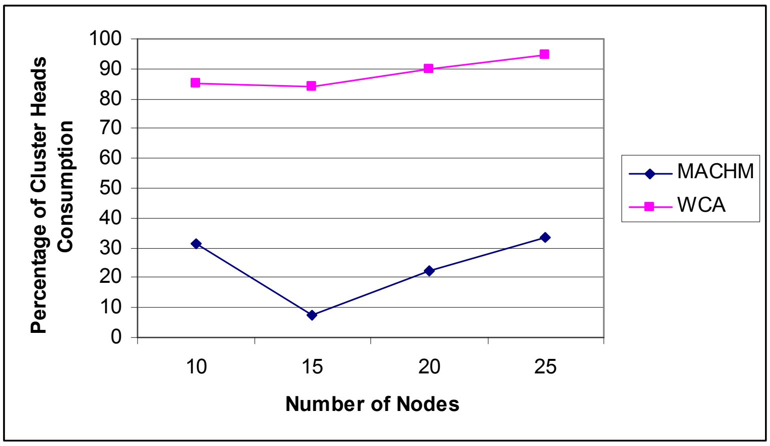


Figure 4.7 (a): Simulation Area = 1000 X 500

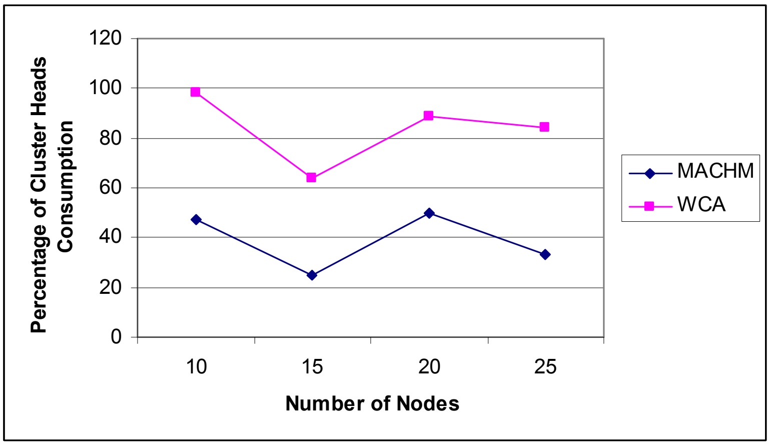


Figure 4.7 (b): Simulation Area = 1500 X 1500

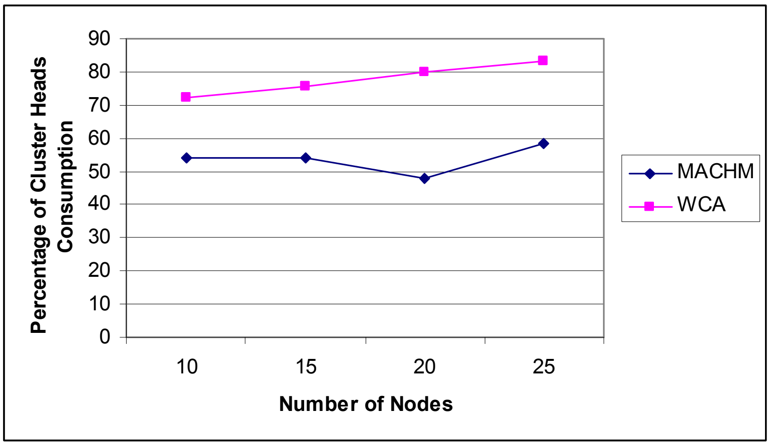


Figure 4.7 (c): Simulation Area = 2500 X 2500

Figure 4.7: Percentage of Cluster Heads Consumption for Different Sizes of Simulation Area

When the mobility is on, results showed that MACHM keep outperforming WCA in the sense of cluster heads battery power drains. For 1000 X 500 simulation area size, MACHM consumed 31.28% (*When the number of nodes is 10*) and 7.66% (*When the number of nodes is 15*) of the total cluster heads battery power, while WCA consumed 85.07% (*When the number of nodes is 10*) and 83.98% (*When the number of nodes is* 15). The improvements obtained ranged from 64% to 91% respectively.

In 1500 X 1500 simulation area size, WCA consumed 88.85% (*When the number of nodes is 20*) and 63.80% (*When the number of nodes is 15*) of the total battery power of cluster heads nodes. At same conditions, MACHM improved the percentage of consumption to 44% and 61%, by consuming 49.76% and 25% respectively.

The main reason of MACHM improvements over WCA is its sensitivity toward cluster heads energy. As mentioned before, MACHM monitor the power level of cluster heads, and then activates its invoking request (*When BPT1 value reached*) to replace the running cluster heads. We notice that, even we gave equal weights (*W3 and W4 = 40%*) for both methods, WCA still can't do better than our method.

This happened because WCA change the set of cluster heads based on mobility monitoring, which happened frequently when the mobility is on. This affect the whole network by re-calculation of nodes weights and restarting of clusters formation process. Unnecessary waste of the collected information occurred, that can be avoided.

MACHM do the re-calculation of nodes weights in a smaller scope, which restricted to the set of first level neighbors for the requesting cluster head. Other stable clusters kept unaware of the cluster head changing happened in other clusters, which reduced the overhead and the percentage of consumption. This ensured that the new cluster head is in the core of the cluster, with best weight value available.

Next, we test the percentage of all nodes battery power consumption metric. Figures 4.8 and 4.9 show the results obtained, when the mobility is off.

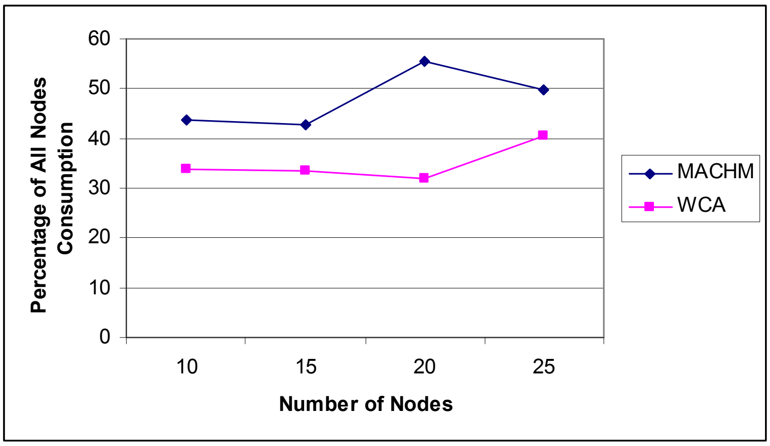


Figure 4.8 (a): BPT1 = 30

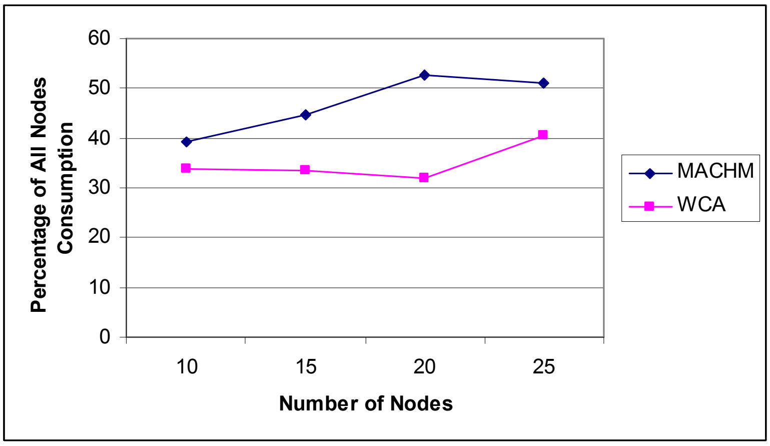


Figure 4.8 (b): BPT1 = 40

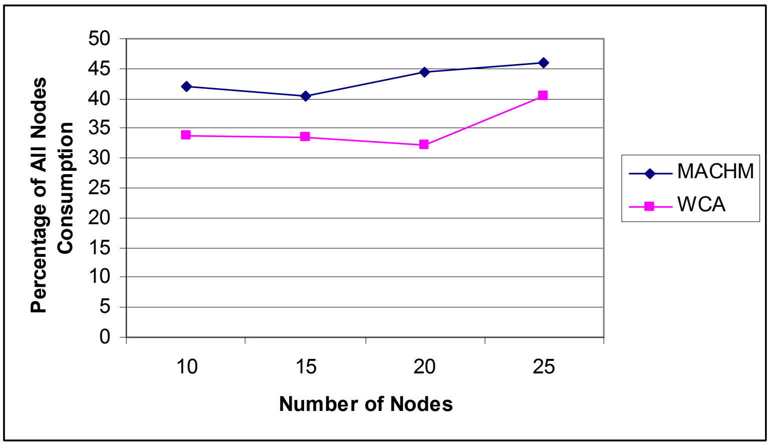


Figure 4.8 (c): BPT1 = 50

Figure 4.8: Percentage of All Nodes Consumption for Different BPT1 Values in 1500 X 1500

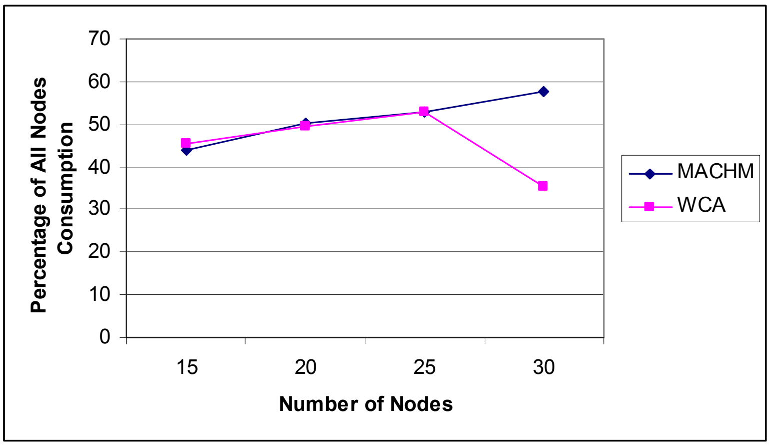


Figure 4.9 (a): BPT1 = 30

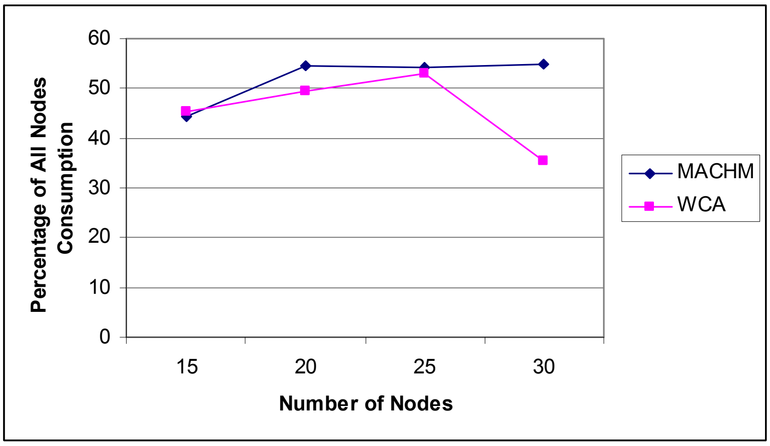


Figure 4.9 (b): BPT1 = 40

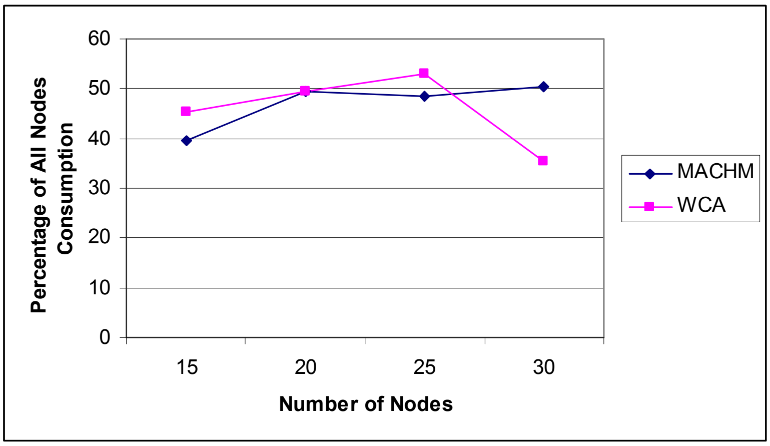


Figure 4.9 (c): BPT1 = 50

Figure 4.9: Percentage of All Nodes Consumption for Different BPT1 Values in 2000 X 2000

The results showed that when the mobility is off, WCA slightly outperformed MACHM in the percentage of all nodes battery power consumption, especially, in lower BPT1 values. In 1500 X 1500 simulation area size, when the number of nodes is 20, WCA consumed 32.07% for BPT1 equaled to 30, 40, and 50. While MACHM consumed 55.44%, 52.72%, and 44.37% for BPT1 equaled to 30, 40, and 50 respectively at the same number of nodes (*Which is 20*).

Expanding the size of simulation area from 1500 X 1500 to 2000 X 2000 resulted to fluctuated performance in this metric. Neither MACHM nor WCA scored absolute advantage in all scenarios, for example when the BPT1 equaled to 30, MACHM consumed 43.92% of all nodes battery power for 15 nodes, compared to 45.30% for WCA. On the other hand, at the same BPT1 but with different number of nodes equaled to 20, MACHM consumed 50.43% compared to consumption of 49.47% for WCA.

Similar results obtained for higher BPT1 values, for example when the BPT1 is 40, MACHM consumed 44.24% of the total battery power of 15 nodes, compared to 45.30% consumption for WCA. At the same BPT1 value, WCA consumed 53.04% of the total battery power of 25 nodes, while MACHM consumed 54.33%.

The previous results obtained, because MACHM and WCA followed different way to activate their invoke requests. In WCA, no invoke requests occurred when the nodes are still and unmoved, therefore, the main load is on cluster heads, keeping the rest of nodes lightly loaded. This explained the slightly outperforming of WCA over MACHM in some conditions for the percentage of all nodes consumption.

While in MACHM the invoke requests are done even when the mobility is off, because it is based on power level monitoring. To study the effect of invoke requests on the consumption of all nodes in the network, Figures 4.10 and 4.11 show the relation between them in MACHM for 1500 X 1500 and 2000 X 2000 simulation area size respectively, when the mobility is off.

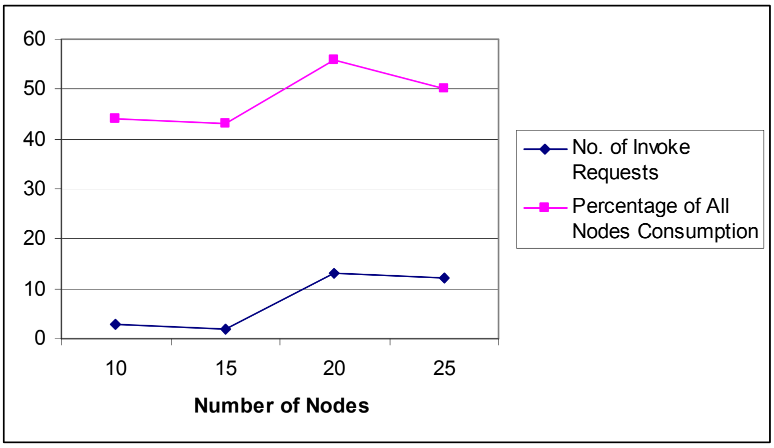


Figure 4.10 (a): BPT1 = 30

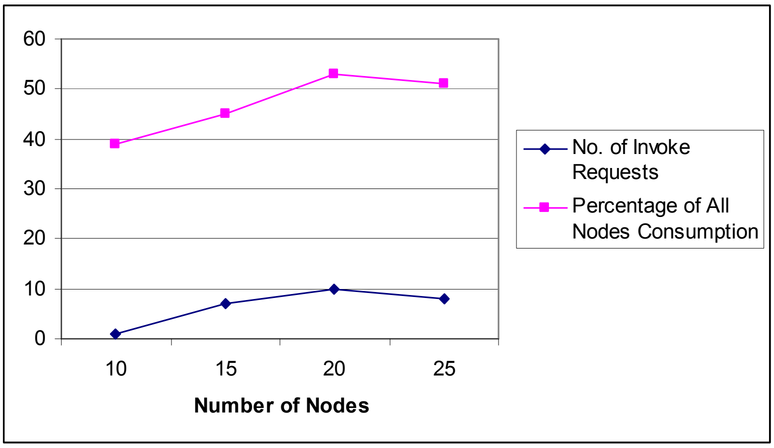


Figure 4.10 (b): BPT1 = 40

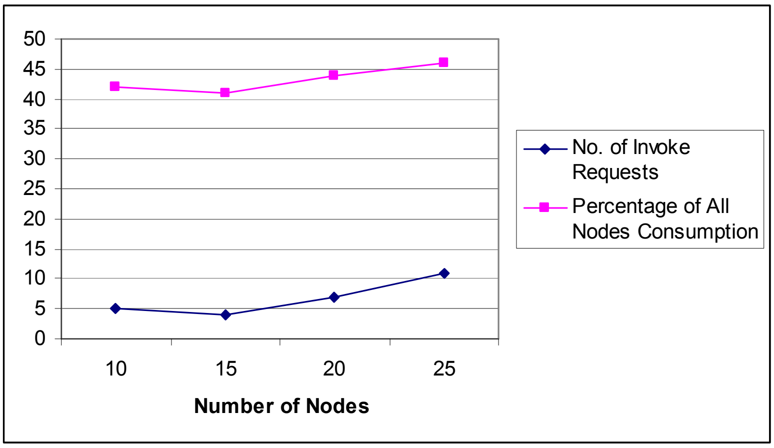


Figure 4.10 (c): BPT1 = 50

Figure 4.10: Invoke Requests vs. Percentage of All Nodes Consumption for Different BPT1 Values in 1500 X 1500

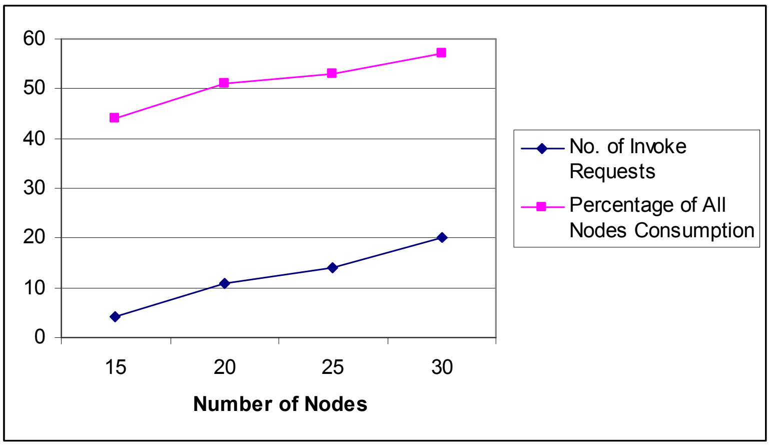


Figure 4.11 (a): BPT1 = 30

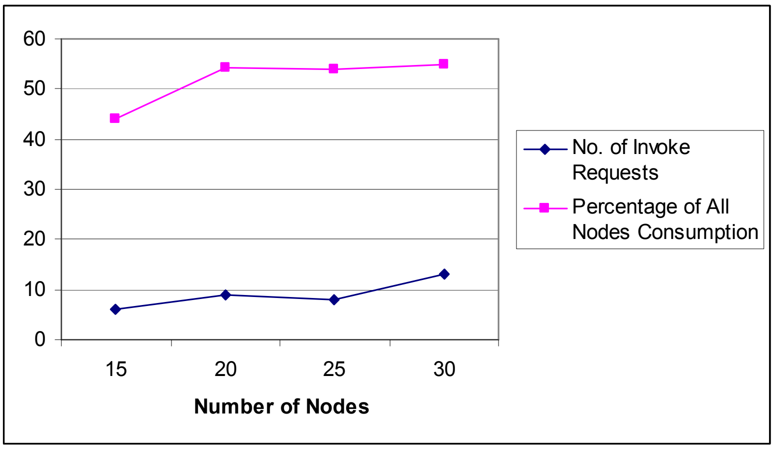


Figure 4.11 (b): BPT1 = 40

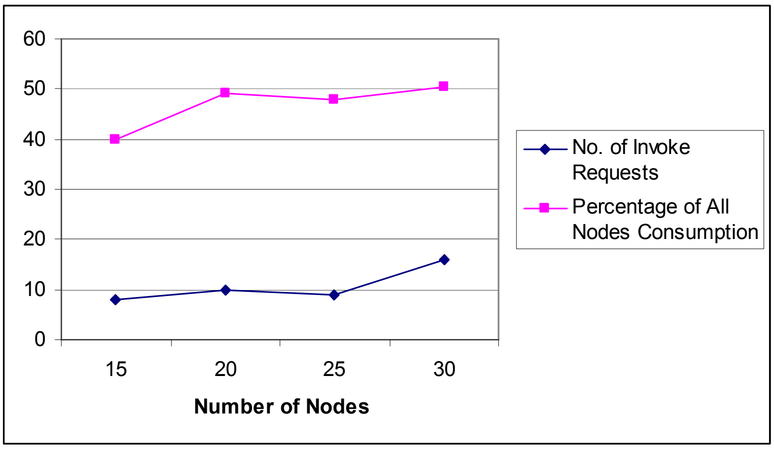


Figure 4.11 (c): BPT1 = 50

Figure 4.11: Invoke Requests vs. Percentage of All Nodes Consumption for Different BPT1 Values in 2000 X 2000

From the previous charts, we can notice the direct relation between the numbers of invoke requests and the percentage of all nodes consumption. The effect of invoke requests control messages is obvious, increasing of invoke requests lead to increased consumption of all nodes battery power, and vise versa.

In 1500 X 1500 simulation area size with BPT1 equaled to 30, MACHM did 4, 3, 13, and 12 invoke requests when the number of nodes is 10, 15, 20 , and 25 respectively, that leads to total consumption of 44.90%, 42.76%, 56.13%, and 51.09%. Similar behavior scored for higher BPT1 values.

Same logic applied for 2000 X 2000 simulation area size with different BPT1 values. For example, when the BPT1 is 40, MACHM did 6, 9, 8, and 14 invoke requests for 15, 20, 25, and 30 nodes respectively, resulted to consumption of 46.07%, 55.39%, 54.60%, and 57% from the total nodes battery power.

In MACHM, invoke requests control messages put an extra load on cluster head's neighbors. A re-calculation of their weights is required, causing extra messages exchanging and battery power draining. This is also, explained the reason of WCA's little advantage on MACHM in dome cases for this metric when the mobility is off.

On the other hand, in the case of nodes mobility WCA will start to do invoke requests based on its update policy. To complete the image, we need to study the percentage of all nodes consumption when the mobility is on. Figure 4.12 shows the simulation results for different sizes of simulation areas.

In this case, simulation results showed different behavior when the nodes start moving around in the terrain. MACHM outperformed WCA and improved the percentage of all nodes battery power consumption.

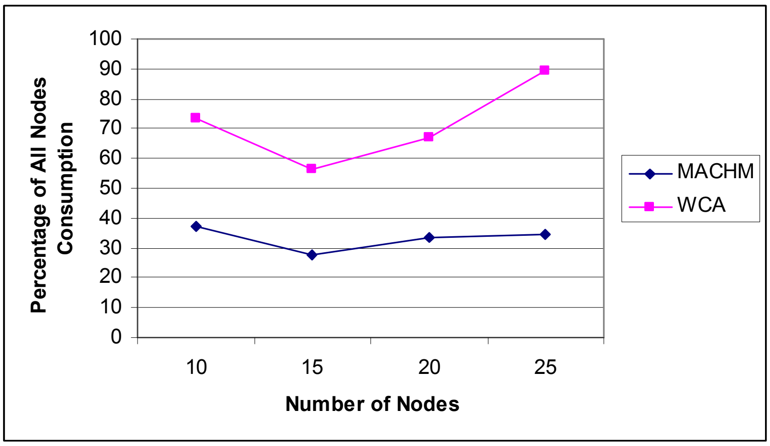


Figure 4.12 (a): Simulation Area = 1000 X 500

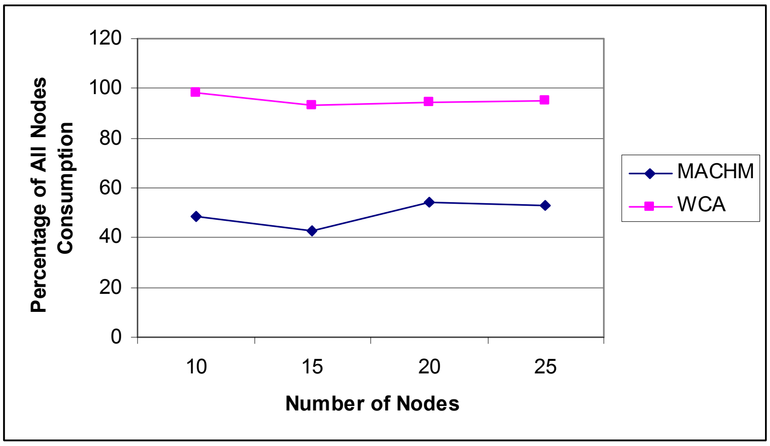


Figure 4.12 (b): Simulation Area = 1500 X 1500

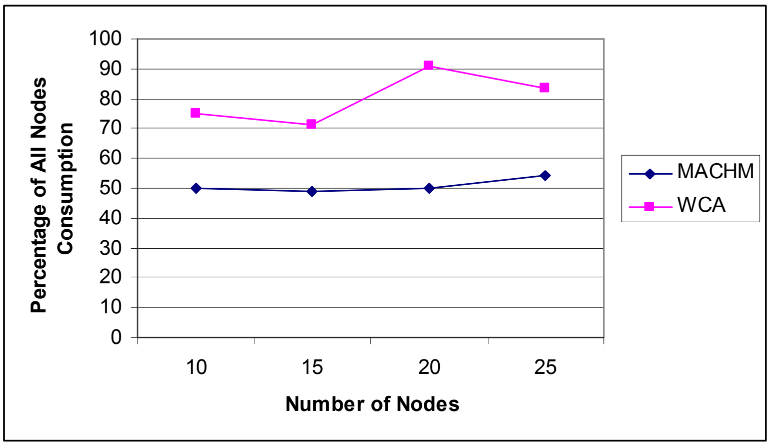


Figure 4.12 (c): Simulation Area = 2500 X 2500

Figure 4.12: Percentage of All Nodes Consumption for Different Sizes of Simulation Area

In 1000 X 500 simulation area size, MACHM consumed 37.20%, 27.71%, 33.30%, and 34.62% of all nodes battery power for number of nodes equaled to 10, 15, 20, and 25 respectively, compared to consumption of 73.62%, 56.39%, 66.78%, and 89.39% in WCA at the same settings.

For 1500 X 1500 simulation area size, the percentage of improvement MACHM achieved over WCA ranged from 43% (*When the number of nodes is 20*) to 55% (*When the number of nodes is 15*), with consumption of 54.39% and 42.75% for MACHM against consumption of 94.37% and 93.39% for WCA.

This happened because WCA starts to activate its invoking requests trigger. In WCA, this is an expensive action that will affect all the nodes in the network. This wider scope caused the previous cluster heads and the rest of nodes to lose extra battery power needed to re-calculate their weights and electing new set of cluster heads, which may be unnecessary for all clusters in the network.

MACHM involved a subset of nodes (*Direct neighbors of old cluster head*), not all the nodes in the network in the update policy. This explained the raising of battery power drains for the rest of nodes happened in WCA, and avoided in MACHM.

Finally, to complete our comparison, we tested the throughput of data messages achieved by MACHM and WCA methods through the progress of simulations. Figures 4.13 and 4.14 show the results obtained for different simulation area sizes when the mobility is off.

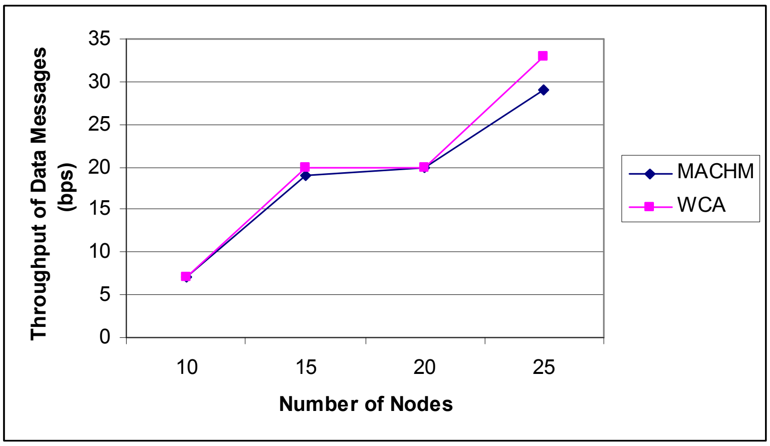


Figure 4.13 (a): BPT1 = 30

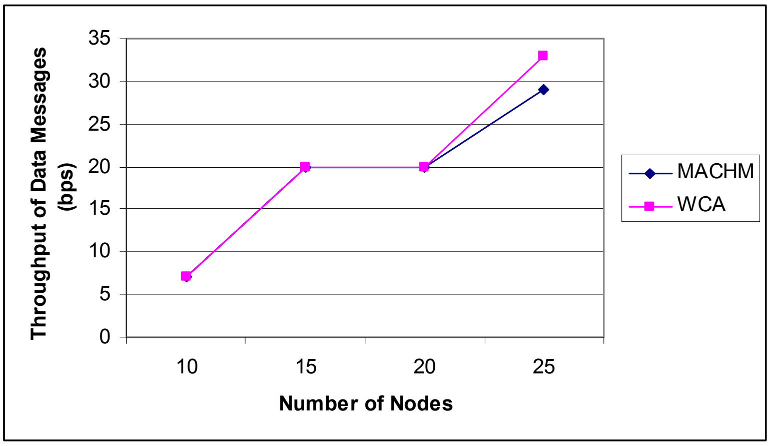


Figure 4.13 (b): BPT1 = 40

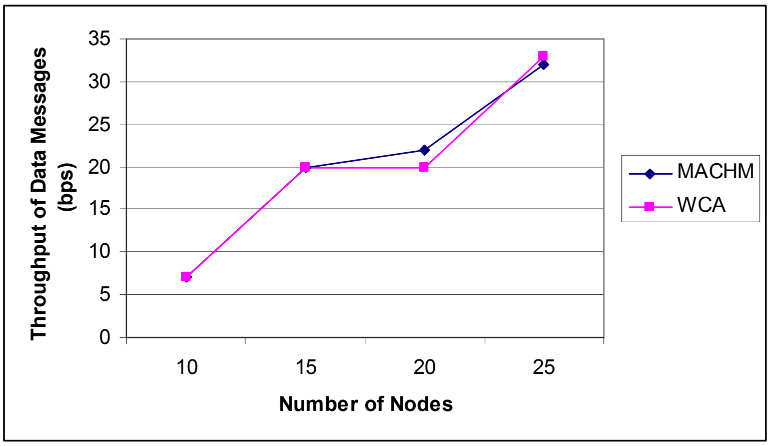


Figure 4.13 (c): BPT1 = 50

Figure 4.13: The Throughput of Data Messages for Different BPT1 Values in 1500 X 1500

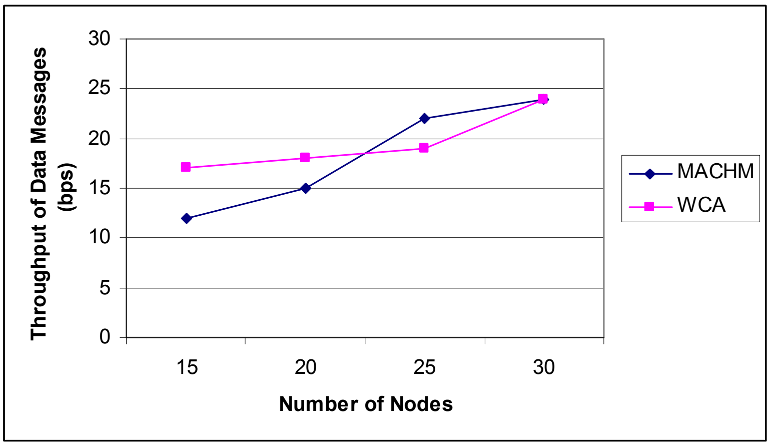


Figure 4.14 (a): BPT1 = 30

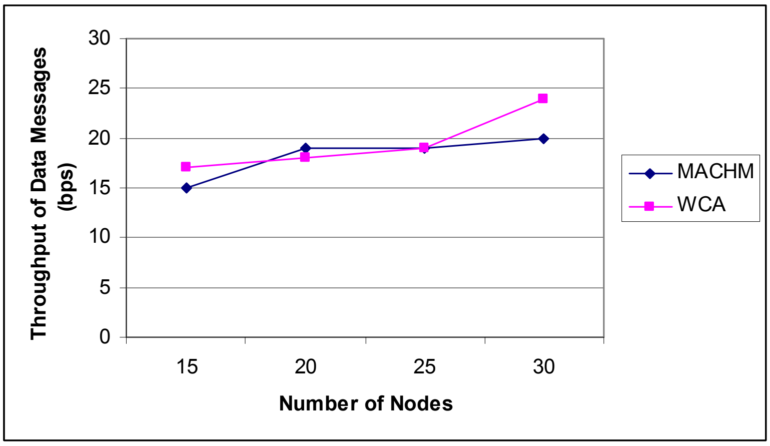


Figure 4.14 (b): BPT1 = 40

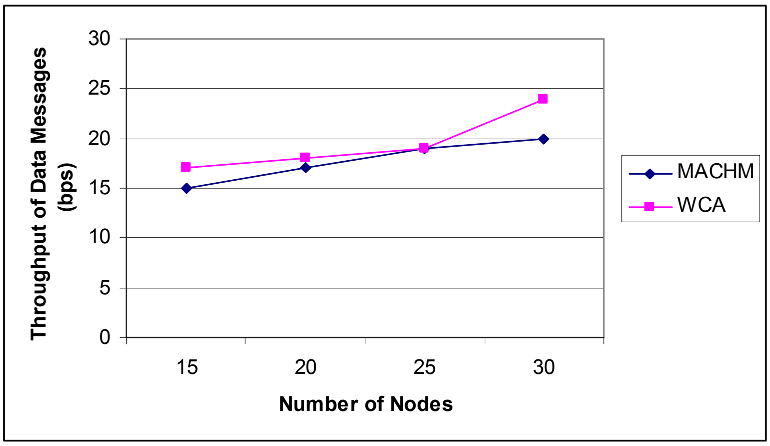


Figure 4.14 (c): BPT1 = 50

Figure 4.14: The Throughput of Data Messages for Different BPT1 Values in 2000 X 2000

Simulation results showed close performance between MACHM and WCA regarding this metric. In 1500 X 1500 simulation area size with BPT1 equaled to 30, both methods achieved identical score equaled to 7 bps (*When the number of nodes is 10*) and 20 bps (*When the number of nodes is 20*). While more matching happened for BPT1 equaled to 40, MACHM and WCA achieved 7 bps, 20 bps, and 20 bps for number of nodes equaled to 10, 15, and 20 respectively.

Increasing the simulation area size to 2000 X 2000, reduced the number of matching and showed bigger difference. For BPT1 equaled to 30, MACHM achieved 22 bps (*When the number of nodes is 25*), while WCA achieved 19 bps. On the other hand, at different number of nodes equaled to 15, MACHM achieved 12 bps compared to 17 bps for WCA.

When the BPT1 is 40 and the number of nodes is 20, MACHM achieved 19 bps compared to 18 bps achieved by WCA. At the same value of BPT1, MACHM scored 20 bps (*When the number of nodes is 30*), while WCA achieved 24 bps for the same number of nodes.

We noticed that increasing the number of nodes at the same terrain resulted to higher value of data messages throughput. For example, in 1500 X 1500 simulation area size with BPT1 equaled to 50, MACHM achieved a series of 7 bps, 20 bps, 22 bps, and 32 bps for number of nodes equaled to 10, 15, 20, and 25 respectively. Similar results obtained for WCA, which achieved a series of 7 bps, 20 bps, 20 bps, and 33 bps at the same network settings.

These results make sense, because increasing the number of nodes covering the same dimensions will increase the chance of finding alternative routes from source to destination, and hence, ascending values of data messages throughput will be scored. Next, we studied the effect of nodes mobility on the throughput of data messages. Figure 4.15 shows simulation results for different network dimensions.

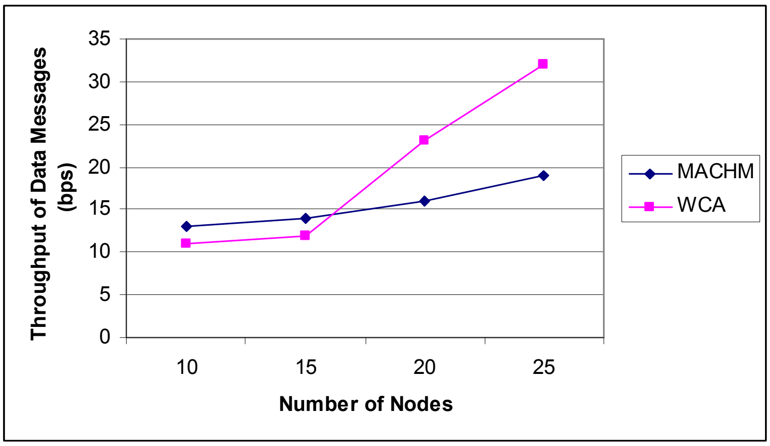


Figure 4.15 (a): Simulation Area = 1000 X 500

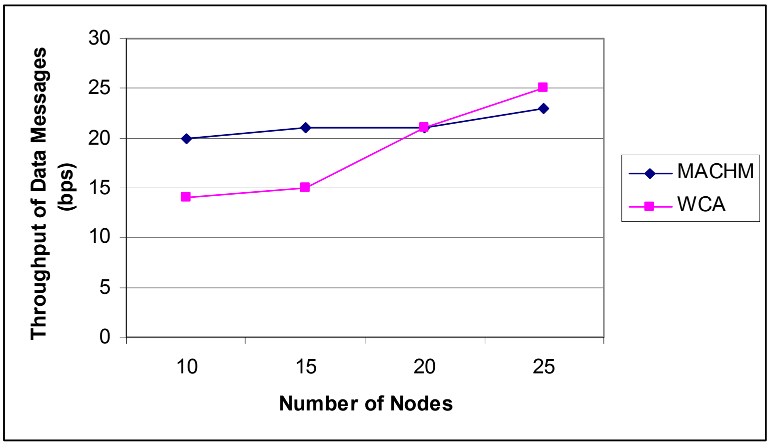


Figure 4.15 (b): Simulation Area = 1500 X 1500

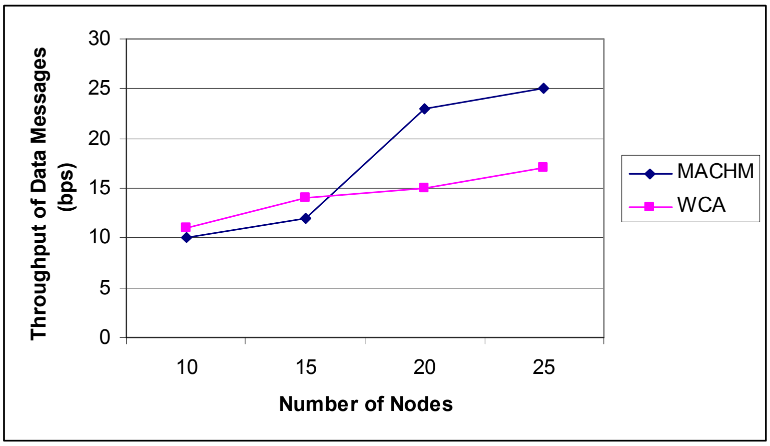


Figure 4.15 (c): Simulation Area = 2500 X 2500

Figure 4.15: Throughput of Data Messages for Different Sizes of Simulation Area

When mobility is on, simulation results showed fluctuated performance for MACHM and WCA, with more obvious difference in the throughput values. For example, in 1000 X 500 simulation area size when the number of nodes is 10, MACHM and WCA scored 13 bps and 11 bps respectively, at same terrain with number of nodes equaled to 25, MACHM and WCA achieved throughput values of 19 bps and 32 bps respectively. Similar differences can be noticed in other sizes of simulation area.

In general, there is not any absolute advantage of any method on the other. MACHM achieved a slight advance in this metric, which can be seen in certain sizes of simulation area with certain number of nodes, while WCA scored the advantage in other conditions and settings.

Refer to Appendix B for the complete simulations run results. We summarized the results of every performance evaluation metric in a single table, in order to compare the performance of each method in an absolute way, we calculate the improvements achieved by MACHM over WCA.

We use different formulas to calculate the improvements, depending on the objective of the metric. When the objective is reduction, like the number of cluster heads, the number of invoke requests, the percentage of cluster heads consumption, and the percentage of all nodes consumption metrics we used this formula:

$$Improvement = (1 - MACHM\_Value / WCA\_Value) * 100 \qquad (4.1)$$

While when the objective of the performance evaluation metric is increasing, like the throughput of data messages metric we used this formula:

$$Improvement = (1 - WCA\_Value / MACHM\_Value) * 100 \qquad (4.2)$$

5. Conclusions and Future Works

5.1 Conclusions

In this study, we have proposed MACHM to reduce the power consumption in cluster-based mobile ad hoc networks. We aim to extend the life time of mobile ad hoc network and keep it operative as long as possible, for scalability reasons we implemented hierarchal solution represented in clustering concept. In cluster-based mobile ad hoc networks the critical key is electing the cluster heads and maintaining them efficiently throughout the progress of the network.

To achieve this goal, we considered a multi-aware approach for electing these cluster heads based on a weighted formula, different factors were included in this formula in a try to elect the most suitable cluster head available. These factors covered the ideal number of nodes that a cluster can handle (*Load Balancing Consideration*), the distance between the node and its neighbors (*Geographical Consideration*), the speed of nodes (*Mobility Consideration*), and most importantly the node's battery power (*Energy Consideration*).

We also proposed a mechanism to maintain the previously elected cluster heads based on their energy level, that looking for a replacement cluster head from the set of direct neighbors to reduce the effect of changing the cluster head. Then we simulate our method against WCA.

In order to compare the performance of MACHM and WCA, different performance metrics were used under different scenarios. These metrics include the number of cluster heads, the number of invoke requests, the percentage of cluster heads consumption, the percentage of all nodes consumption, and the throughput of data messages.

Referring to Table B.1, we can notice that according to the number of cluster heads metric, MACHM achieved less number of cluster heads for the same number of nodes distributed in the same simulation area up to 58%. This means fewer amounts of data considering other cluster heads in the network, but with reasonable members in the same cluster. The improvement achieved by MACHM over WCA in this metric can be noticed easily.

According to Table B.2, we concluded that in WCA selecting the mobility as the trigger activation for the invoke requests resulted in a much higher overhead. We can notice the huge improvement achieved, which reached up to 94%. This showed that the choice of activating the invoking based on battery power considerations in MACHM, minimized the number of invoke requests control messages, and hence the overhead in the network.

As a result, lower number of cluster heads with lower number of invoke requests in MACHM resulted better percentage of the battery power consumed for the cluster heads as Table B.3 showed. We recognized that MACHM achieved our main goal, and reduced the consumption percentage for all conditions tested. The improvement in this metric reached to 91% better than WCA.

On the other hand, we tested the percentage of all nodes consumption to check the distribution of the overhead among the rest of nodes in the network. When the mobility is off, WCA will not do any invoke requests and this explained the reason of its advancing in some scenarios. Table B.4 showed the results obtained for this metric.

As we mentioned before, the main overhead on ordinary nodes caused by the invoke requests generated by the cluster heads, because it required re-calculation of node's weights. MACHM outperformed WCA in this metric when the mobility is on and scored an improvement up to 62%.

Finally, we studied the throughput of data messages achieved by MACHM against WCA, which gave us an indication about the efficiency of each method. After referring to Table B.5, we obtained close performance between MACHM and WCA in this metric, and MACHM was able to achieve an improvement reached 35%.

However, even if MACHM provides lower throughput of data messages than WCA in some cases, it still better in the over all performance, because it provides close values of throughput of data messages for longer time. The extension in the life time of the network came from the preservation in the battery power for the cluster heads and the rest of nodes, which will yields a higher percentage of messages delivery through out the progress of the network.

At the end, we concluded that MACHM outperforms WCA in reducing battery power consumption for the cluster heads and ordinary nodes by reducing the number of cluster heads covering the same simulation area, reducing the overhead of control messages, and presenting good throughput for longer time.

5.2 Future Works

In this study, we constrained on extending network's life time as long as possible, by reducing the node's battery power consumption. While this is an important property, we will investigate the opportunity to provide other properties beside it, like guaranteeing network's connectivity, or adding new factors to the weighted formula, that will cover other aspects of ad hoc networks.

On the other hand, we will study the ability of adding new battery power threshold. The proposed threshold may be used to trigger certain action other than invoking requests if needed.

References

Cano J., and Manzoni P., **'Reducing Energy Consumption in a Clustered MANET using the Intra Cluster Data-Dissemination Protocol (Icdp)'**, Proceedings of the 10th Euromicro Workshop on Parallel, Distributed and Network-based Processing, IEEE, 2002.

Chatterjee M., Das S., and Turgut D., **'WCA: A Weighted Clustering Algorithm for Mobile Ad Hoc Networks'**, Kluwer Academic Publishers, Cluster Computing Vol. 5, pp. 193-204, 2002.

Chen B., Jamieson K., Morris R., and Balakrishnan H., **'Span: An Energy-Efficient Coordination Algorithm for Topology Maintenance in Ad Hoc Wireless Networks'**, Proceedings of 7th ACM International Conference on Mobile Computing and Networking, 2001.

Flickenger R., Aichele C., Fonda C., Forster J., Howard I., Krag T., and Zennaro M., **'Wireless Networking in the Developing World'**, 1st Edition, Limehouse Book Sprint Team, 2006.

Gerla M, Bajaj L, Takai M., Ahuja R, and Bagrodia R., **'GloMoSim: A Scalable Network Simulation Environment'**, Technical Report 990027, University of California, 1999.

Haas Z., and Tabrizi S., **'On Some Challenges and Design Choices in Ad-Hoc Communications'**, Proceedings of IEEE Military Communications Conference, IEEE, 1998.

Halsall F., **'Computer Networking and the Internet'**, 5th Edition, Pearson Education (Addison Wesley), ISBN 0-321-26358-8, 2005.

Hong X., Xu K., and Gerla M., **'Scalable Routing Protocols for Mobile Ad Hoc Networks'**, IEEE Network magazine, Vol.16(4), pp. 11-21, 2002.

Johansson T., and Carr-Motyckova L., **'On Clustering in Ad Hoc Networks'**, Proceedings of Vehicular Tech. Conf. Fall , Swedish National Computer Networking Workshop 2003, (2003).

Jones C., Sivalingam K., Agrawal P., and Chen J., **'A Survey of Energy Efficient Network Protocols for Wireless Networks'**, Wireless Networks, Vol. 7 , Issue 4, 2001.

Kawadia V., and Kumar P., **'Power Control and Clustering in Ad hoc Networks'**, Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies, IEEE, Vol. 1, No. 1, pp. 459-469, 2003.

Kuosmanen P., **'Classification of Ad Hoc Routing Protocols'**, Finnish Defense Forces, Naval Academy, 2002.

Latiff L., Ali A., Ooi C-C., and Fisal N., **'Location-based Geocasting and Forwarding (LGF) Routing Protocol in Mobile Ad Hoc Network'**, Proceedings of the Advanced Industrial Conference on Telecommunications/Service Assurance with Partial and Intermittent Resources Conference/E-Learning on Telecommunications Workshop, IEEE, 2005.

Liang B., and Haas Z., **'Virtual Backbone Generation and Maintenance in Ad Hoc Network Mobility Management'**, Proceedings of the 19th Annual Joint Conference of the IEEE Computer and Communications Societies, IEEE, 2000.

Mauve M., Widmer J., and Hartenstein H., **'A Survey on Position-based Routing in Mobile Ad Hoc Networks'**, IEEE Network Magazine, pp. 30-39, 2001.

Mellier R., Myoupo J., and Sow I., **'GPS-Free Geocasting Algorithms in Mobile Ad hoc Networks'**, Proceedings of IEEE International Conference on Wireless and Mobile Communications, IEEE, 2006.

Rappaport T., **'Wireless Communications'**, Principles and Practice', 2nd Edition, Prentice Hall, ISBN 0130422320, 2002.

Sasson Y., Cavin D., and Schiper A., **'A Location Service Mechanism for Position-Based Multicasting in Wireless Mobile Ad Hoc Networks'**, Proceedings of the 38th Annual Hawaii International Conference on System Sciences, 2005.

Sheltami T., and Mouftah H., **'An Efficient Energy Aware Clusterhead Formation Infrastructure Protocol for MANETs'**, Proceedings of the 8th IEEE International Symposium on Computers and Communication, IEEE, 2003.

Sinha P., Sivakumar R., and Vaduvur B., **'Enhancing Ad Hoc Routing with Dynamic Virtual Infrastructures'**, Proceedings of the 20th Annual Joint Conference of the IEEE Computer and Communications Societies, IEEE, 2001.

Sivakumar R., Sinha P., and Bharghavan V., **'CEDAR: a Core-Extraction Distributed Ad Hoc Routing Algorithm'**, IEEE Journal on Selected Areas in Communication, Vol. 17, No. 8, pp. 1-12, 1999.

Tanenbaum A., **'Computer Networks'**, 3rd Edition, Prentice Hall of India Private Limited, New Delhi, ISBN 81-203-1165-5, 1997.

Tanenbaum A., **'Distributed Operating Systems'**, 11th Indian Reprint, Pearson Education, Inc., ISBN 81-7808-294-2, 1995.

URL: **http://www.iwi.uni-hannover.de/lv/seminar_ss04/www/Jan_Gacnik/xhtml-/section3.html** last seen at May, 2008.

Wang X., Guo X., and Yin X., **'NRCE: A New Random Cluster Election Algorithm'**, Journal of Communication and Computer, Vol. 3, No. 3, pp. 40-45, 2006.

Wang Y., Chen H., Yang X., and Zhang D., **'Cluster Based Location-Aware Routing Protocol for Large Scale Heterogeneous MANET'**, Proceedings of 2nd International Multisymposium on Computer and Computational Sciences, IEEE, 2007.

Wu J., Dai F., Gao M., and Stojmenovic I., **'On Calculating Power-aware Connected Dominating Sets for Efficient Routing in Ad Hoc Wireless Networks'**, IEEE/KICS Journal of Communication Networks, Vol. 4, No. 1, pp. 59-70, 2002.

Wu Z., Dong X., and Cui L., **'A Grid-based Energy Aware Node-Disjoint Multipath Routing Algorithm for MANETs'**, Proceedings of 3rd International Conference on Natural Computation, IEEE, 2007.

Yu J., and Chong P., **'A Survey of Clustering Schemes for Mobile Ad Hoc Networks'**, IEEE Communications Survey & Tutorial, First Quarter, Vol. 7, No. 1, 2005.

Appendix A: GloMoSim

This part is taken from (Gerla *et al*., 1999) by paraphrasing and aggregation.

Global Mobile Information System Simulator (GloMoSim) is a scalable simulation environment for large wireless and wired communication networks. GloMoSim uses a parallel discrete-event simulation capability provided by Parsec.

GloMoSim is designed in a layered approach with standard APIs used between the different simulation layers. The protocol stack includes models for the channel, radio, MAC, network, transport and higher layers. The GloMoSim kernel APIs are in the form of function calls, while for the other layers, the API is in the form of message exchanges required to interact with the layers.

GloMoSim simulates networks with up to thousand nodes linked by a heterogeneous communications capability that includes multicast, asymmetric communications using direct satellite broadcasts, multi-hop wireless communications using ad-hoc networking, and traditional Internet protocols. The following table lists the GloMoSim models currently available at each of the major layers:

Table A.1: Available Models in GloMoSim

Layer	Models
Physical (Radio Propagation)	Free space, Two-Ray
Data Link (MAC)	CSMA, MACA, TSMA, 802.11
Network (Routing)	Bellman-Ford, FSR, OSPF, DSR, WRP, LAR, AODV
Transport	TCP, UDP
Application	Telnet, FTP

The node aggregation technique is introduced into GloMoSim to give significant benefits to the simulation performance. Initializing each node as a separate entity inherently limits the scalability because the memory requirements increase dramatically for a model with large number of nodes. With node aggregation, a single entity can simulate several network nodes in the system.

Node aggregation technique implies that the number of nodes in the system can be increased while maintaining the same number of entities in the simulation. In GloMoSim, each entity represents a geographical area of the simulation. Hence the network nodes which a particular entity represents are determined by the physical position of the nodes.

GloMoSim has a Visualization Tool that is platform independent because it is coded in Java. To initialize the Visualization Tool, we must execute from the *java_gui* directory the following: *java GlomoMain*. This tool allows to debug and verify models and scenarios; stop, resume and step execution; show packet transmissions, show mobility groups in different colors and show statistics.

The radio layer is displayed in the Visualization Tool as follows: When a node transmits a packet, a yellow link is drawn from this node to all nodes within its power range. As each node receives the packet, the link is erased and a green line is drawn for successful reception and a red line is drawn for unsuccessful reception. No distinction is made between different packet types (*i.e. control packets vs. regular packets, etc*).

The main configuration parameters for setting up a scenario are defined in the CONFIG.IN file. Table A.2 shows these parameters:

Table A.2: GloMoSim Parameters

Parameter	Description
SIMULATION-TIME	Maximum simulation time. The number portion can be followed by optional letters to modify the simulation time.
SEED	Random number used to initialize part of the seed of various randomly generated numbers in the simulation.
TERRAIN-DIMENSIONS	Terrain Area simulated in meters.
NUMBER-OF-NODES	Number of nodes being simulated.
NODE-PLACEMENT	Represents the node placement strategy.
MOBILITY	Represents the mobility model.
PROPAGATION-LIMIT	Signals below this parameter (in dBm) are not delivered.
PROPAGATION-PATHLOSS	Specifies the path loss model.
TEMPERATURE	Temperature of the environment (in K).
RADIO-TYPE	Radio model to transmit and receive packets.
RADIO-FREQUENCY	Frequency in Hertz.
RADIO-BANDWIDTH	Bandwidth in bits per second.
RADIO-RX-TYPE	Specifies the packet reception model.
RADIO-TX-POWER	Radio transmission power (in dBm).
RADIO-ANTENNA-GAIN	Antenna Gain (in dB).
RADIO-RX-SENSITIVITY	Sensitivity of the radio (in dBm).
RADIO-RX-THRESHOLD	Minimum power for received packet (in dBm).
MAC-PROTOCOL	Definition of Medium Access Protocol.
PROMISCUOUS-MODE	It is set to YES if nodes want to overhear packets destined to the neighboring node.
NETWORK-PROTOCOL	Definition of the Network Protocol.
ROUTING-PROTOCOL	Definition of the Routing Protocol.
APP-CONFIG-FILE	Specifies the file that sets up applications such as FTP, CBR and Telnet.

Applications such as FTP and Telnet are configured in APP.CONF file, the traffic generators currently available are FTP, FTP/GENERIC, TELNET, CBR, and HTTP.

Because of the way radio transmissions are affected by the environment in such a complex way, it is quite difficult to predict the comportment of a system and to define a radio transmission range of a node. The radio range is the average maximum distance in usual operating conditions between two nodes.

The only available mobility model in GloMoSim v2.03 is the Random Waypoint Mobility Model (RWPM). In this model a node randomly selects a destination from the physical terrain, and moves in the direction of that destination in a speed uniformly chosen between MOBILITY-WP-MIN-SPEED and MOBILITY-WP-MAX-SPEED parameters (*defined in meter/sec*). After it reaches its destination, the node stays there for a MOBILITY-WP-PAUSE time period.

If we want to use mobility patterns other than RWPM, then we must specify the parameter MOBILITY TRACE in order to indicate GloMoSim that individual movements for nodes will be taken from a file specified by MOBILITY-TRACE-FILE.

Appendix B: Results Summary

Table B.1: The Number of Cluster Heads by MACHM and WCA under Different Scenarios

Mobility	BPT1	Simulation Area	Number of Nodes	MACHM	WCA	MACHM Improvement over WCA
Off	30	1500 X 1500	10	2	4	50%
Off	30	1500 X 1500	15	5	5	0%
Off	30	1500 X 1500	20	6	7	15%
Off	30	1500 X 1500	25	7	9	23%
Off	40	1500 X 1500	10	3	4	25%
Off	40	1500 X 1500	15	4	5	20%
Off	40	1500 X 1500	20	6	7	15%
Off	40	1500 X 1500	25	7	9	23%
Off	50	1500 X 1500	10	2	4	50%
Off	50	1500 X 1500	15	3	5	40%
Off	50	1500 X 1500	20	4	7	43%
Off	50	1500 X 1500	25	6	9	34%
Off	30	2000 X 2000	15	6	8	25%
Off	30	2000 X 2000	20	10	11	10%
Off	30	2000 X 2000	25	11	15	27%
Off	30	2000 X 2000	30	13	17	24%
Off	40	2000 X 2000	15	5	8	38%
Off	40	2000 X 2000	20	8	11	28%
Off	40	2000 X 2000	25	10	15	34%
Off	40	2000 X 2000	30	12	17	30%
Off	50	2000 X 2000	15	4	8	50%
Off	50	2000 X 2000	20	7	11	37%
Off	50	2000 X 2000	25	9	15	40%
Off	50	2000 X 2000	30	12	17	30%
On	40	1000 X 500	10	2	3	34%
On	40	1000 X 500	15	4	6	34%
On	40	1000 X 500	20	5	8	38%
On	40	1000 X 500	25	7	9	23%
On	40	1500 X 1500	10	3	7	58%
On	40	1500 X 1500	15	5	8	38%
On	40	1500 X 1500	20	6	9	34%
On	40	1500 X 1500	25	10	11	10%
On	40	2500 X 2500	10	4	9	56%
On	40	2500 X 2500	15	6	11	46%
On	40	2500 X 2500	20	10	14	29%
On	40	2500 X 2500	25	13	16	19%

Table B.2: The Number of Invoke Requests by MACHM and WCA under Different Scenarios

Mobility	BPT1	Simulation Area	Number of Nodes	MACHM	WCA	MACHM Improvement over WCA
On	40	1000 X 500	10	4	11	64%
On	40	1000 X 500	15	5	30	84%
On	40	1000 X 500	20	6	34	83%
On	40	1000 X 500	25	9	62	86%
On	40	1500 X 1500	10	6	42	84%
On	40	1500 X 1500	15	9	73	88%
On	40	1500 X 1500	20	10	102	91%
On	40	1500 X 1500	25	13	206	94%
On	40	2500 X 2500	10	5	10	50%
On	40	2500 X 2500	15	8	30	74%
On	40	2500 X 2500	20	9	32	72%
On	40	2500 X 2500	25	12	96	88%

Table B.3: The Percentage of Cluster Heads Consumption by MACHM and WCA under Different Scenarios

Mobility	BPT1	Simulation Area	Number of Nodes	MACHM	WCA	MACHM Improvement over WCA
Off	30	1500 X 1500	10	40.07%	64.47%	38%
Off	30	1500 X 1500	15	50.00%	63.80%	22%
Off	30	1500 X 1500	20	43.82%	73.97%	41%
Off	30	1500 X 1500	25	43.25%	64.56%	34%
Off	40	1500 X 1500	10	54.90%	64.47%	15%
Off	40	1500 X 1500	15	40.07%	63.80%	38%
Off	40	1500 X 1500	20	49.35%	73.97%	34%
Off	40	1500 X 1500	25	51.87%	64.56%	20%
Off	50	1500 X 1500	10	15.62%	64.47%	76%
Off	50	1500 X 1500	15	13.51%	63.80%	79%
Off	50	1500 X 1500	20	21.96%	73.97%	71%
Off	50	1500 X 1500	25	35.62%	64.56%	45%
Off	30	2000 X 2000	15	54.80%	67.50%	19%
Off	30	2000 X 2000	20	40.52%	66.52%	40%
Off	30	2000 X 2000	25	44.78%	65.47%	32%
Off	30	2000 X 2000	30	52.17%	71.90%	28%
Off	40	2000 X 2000	15	42.89%	67.50%	37%
Off	40	2000 X 2000	20	50.50%	66.52%	25%
Off	40	2000 X 2000	25	50.50%	65.47%	23%
Off	40	2000 X 2000	30	50.71%	71.90%	30%
Off	50	2000 X 2000	15	9.37%	67.50%	87%
Off	50	2000 X 2000	20	42.60%	66.52%	36%
Off	50	2000 X 2000	25	42.60%	65.47%	35%
Off	50	2000 X 2000	30	33.69%	71.90%	54%
On	40	1000 X 500	10	31.28%	85.07%	64%
On	40	1000 X 500	15	7.66%	83.98%	91%
On	40	1000 X 500	20	22.22%	90.13%	76%
On	40	1000 X 500	25	33.46%	94.60%	65%
On	40	1500 X 1500	10	47.15%	98.40%	53%
On	40	1500 X 1500	15	25.00%	63.80%	61%
On	40	1500 X 1500	20	49.76%	88.85%	44%
On	40	1500 X 1500	25	33.24%	84.32%	61%
On	40	2500 X 2500	10	54.10%	72.08%	25%
On	40	2500 X 2500	15	54.10%	75.44%	29%
On	40	2500 X 2500	20	47.78%	79.78%	41%
On	40	2500 X 2500	25	58.50%	83.50%	30%

Table B.4: The Percentage of All Nodes Consumption by MACHM and WCA under Different Scenarios

Mobility	BPT1	Simulation Area	Number of Nodes	MACHM	WCA	MACHM Improvement over WCA
Off	30	1500 X 1500	10	43.64%	33.90%	-23%
Off	30	1500 X 1500	15	42.64%	33.58%	-22%
Off	30	1500 X 1500	20	55.44%	32.07%	-43%
Off	30	1500 X 1500	25	49.74%	40.43%	-19%
Off	40	1500 X 1500	10	39.40%	33.90%	-14%
Off	40	1500 X 1500	15	44.66%	33.58%	-25%
Off	40	1500 X 1500	20	52.72%	32.07%	-40%
Off	40	1500 X 1500	25	50.96%	40.43%	-21%
Off	50	1500 X 1500	10	41.91%	33.90%	-20%
Off	50	1500 X 1500	15	40.51%	33.58%	-18%
Off	50	1500 X 1500	20	44.37%	32.07%	-28%
Off	50	1500 X 1500	25	45.94%	40.43%	-12%
Off	30	2000 X 2000	15	43.92%	45.30%	4%
Off	30	2000 X 2000	20	50.43%	49.47%	-2%
Off	30	2000 X 2000	25	52.83%	53.04%	1%
Off	30	2000 X 2000	30	57.63%	35.29%	-39%
Off	40	2000 X 2000	15	44.24%	45.30%	3%
Off	40	2000 X 2000	20	54.56%	49.47%	-10%
Off	40	2000 X 2000	25	54.33%	53.04%	-3%
Off	40	2000 X 2000	30	54.96%	35.29%	-36%
Off	50	2000 X 2000	15	39.44%	45.30%	13%
Off	50	2000 X 2000	20	49.38%	49.47%	1%
Off	50	2000 X 2000	25	48.60%	53.04%	9%
Off	50	2000 X 2000	30	50.49%	35.29%	-31%
On	40	1000 X 500	10	37.20%	73.62%	50%
On	40	1000 X 500	15	27.71%	56.39%	51%
On	40	1000 X 500	20	33.30%	66.78%	51%
On	40	1000 X 500	25	34.62%	89.39%	62%
On	40	1500 X 1500	10	48.35%	98.58%	51%
On	40	1500 X 1500	15	42.75%	93.39%	55%
On	40	1500 X 1500	20	54.39%	94.37%	43%
On	40	1500 X 1500	25	53.04%	95.26%	45%
On	40	2500 X 2500	10	49.92%	75.19%	34%
On	40	2500 X 2500	15	48.93%	71.10%	32%
On	40	2500 X 2500	20	50.04%	90.86%	45%
On	40	2500 X 2500	25	54.26%	83.44%	35%

Table B.5: The Throughput of Data Messages by MACHM and WCA under Different Scenarios

Mobility	BPT1	Simulation Area	Number of Nodes	MACHM	WCA	MACHM Improvement over WCA
Off	30	1500 X 1500	10	7 bps	7 bps	0%
Off	30	1500 X 1500	15	19 bps	20 bps	-5%
Off	30	1500 X 1500	20	20 bps	20 bps	0%
Off	30	1500 X 1500	25	29 bps	33 bps	-13%
Off	40	1500 X 1500	10	7 bps	7 bps	0%
Off	40	1500 X 1500	15	20 bps	20 bps	0%
Off	40	1500 X 1500	20	20 bps	20 bps	0%
Off	40	1500 X 1500	25	29 bps	33 bps	-13%
Off	50	1500 X 1500	10	7 bps	7 bps	0%
Off	50	1500 X 1500	15	20 bps	20 bps	0%
Off	50	1500 X 1500	20	22 bps	20 bps	10%
Off	50	1500 X 1500	25	32 bps	33 bps	-4%
Off	30	2000 X 2000	15	12 bps	17 bps	-30%
Off	30	2000 X 2000	20	15 bps	18 bps	-17%
Off	30	2000 X 2000	25	22 bps	19 bps	14%
Off	30	2000 X 2000	30	24 bps	24 bps	0%
Off	40	2000 X 2000	15	15 bps	17 bps	-12%
Off	40	2000 X 2000	20	19 bps	18 bps	6%
Off	40	2000 X 2000	25	19 bps	19 bps	0%
Off	40	2000 X 2000	30	20 bps	24 bps	-17%
Off	50	2000 X 2000	15	15 bps	17 bps	-12%
Off	50	2000 X 2000	20	17 bps	18 bps	-6%
Off	50	2000 X 2000	25	19 bps	19 bps	0%
Off	50	2000 X 2000	30	20 bps	24 bps	-17%
On	40	1000 X 500	10	13 bps	11 bps	16%
On	40	1000 X 500	15	14 bps	12 bps	15%
On	40	1000 X 500	20	16 bps	23 bps	-31%
On	40	1000 X 500	25	19 bps	32 bps	-41%
On	40	1500 X 1500	10	20 bps	14 bps	30%
On	40	1500 X 1500	15	21 bps	15 bps	29%
On	40	1500 X 1500	20	21 bps	21 bps	0%
On	40	1500 X 1500	25	23 bps	25 bps	-8%
On	40	2500 X 2500	10	10 bps	11 bps	-10%
On	40	2500 X 2500	15	12 bps	14 bps	-15%
On	40	2500 X 2500	20	23 bps	15 bps	35%
On	40	2500 X 2500	25	25 bps	17 bps	32%

تقليل إستهلاك الطاقة في شبكات الحاسوب العشوائية المنعقدة

إعداد

راني عبدالله المحارمة

المشرف

الدكتور عماد صلاح

المشرف المشارك

الأستاذ الدكتور نديم عبيد

ملخص

تعتبر الموارد الخاصة بالجهاز محدودة وقيمة في الشبكات العشوائية النقالة، وأكثرها أهمية طاقة البطارية الخاصة بالجهاز, وهذا المصدر محدود نظرا للحجم الصغير نسبيا للأجهزة النقالة, كما تعتبر إدارة هذا المصدر المحدود من التحديات الرئيسة في بيئة الشبكات العشوائية النقالة.

لغايات تحقيق قابلية القياس, تم تبني حلول طبقية مثل المفهوم المبني على العناقيد. تعتبر عملية إختيار رؤوس العناقيد وإدامتها خلال عمل الشبكات العشوائية وظائف حيوية ومهمة.

تم وضع العديد من المقترحات التي تتضمن عوامل مختلفة بما فيها عامل الطاقة من أجل إختيار أفضل رؤوس عناقيد متوفرة. قمنا بإعتماد طريقة متعددة الحساسية والإهتمامات تسمى صيانة الرأس العنقودي متعددة الحساسية لإختيار رؤوس عناقيد إعتمادا على معادلة موزونة, وهذه المعادلة تتضمن عوامل تتعلق بتوزيع الحمل والجهد وإعتبارات جغرافية وحركية وإعتبارات تتعلق بالطاقة.

تهدف طريقة صيانة الرأس العنقودي متعددة الحساسية إلى تقليل كمية الطاقة المستهلكة من قبل الأجهزة الموجودة في الشبكة، وخاصة الرؤوس العنقودية التي تعتبر أكثر حساسية لفقدان الطاقة وذلك بسبب دورها المهم في الشبكة.

كما قمنا بإقتراح آلية من أجل إدامة رؤوس العناقيد التي تم إختيارها سابقا إعتمادا على مستوى الطاقة الخاص بها، وذلك بالبحث عن رأس عنقودي بديل من مجموعة الجيران المباشرين للرأس العنقودي السابق وذلك بهدف تخفيف التأثير الناتج عن عملية تغيير الرأس العنقودي. نتائج المحاكاة التي قمنا بها أظهرت أن طريقتنا تفوقت على خوارزمية العنقدة الموزونة بما يتعلق بعوامل عدد الرؤوس العنقودية وعدد طلبات الإستدعاء ونسبة الطاقة المستهلكة للرؤوس العنقودية وكل الأجهزة وكمية رسائل البيانات المستلمة.