uOttawa

L'Université canadienne
Canada's university

FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTOCTORALES

uOttawa

L'Université canadienne
Canada's university

FACULTY OF GRADUATE AND POSDOCTORAL STUDIES

Mohammed Saeed Al-Kahtani

AUTEUR DE LA THÈSE / AUTHOR OF THESIS

Ph.D. (Electrical Engineering)

GRADE / DEGREE

School of Information Technology and Engineering

FACULTÉ, ÉCOLE, DÉPARTEMENT / FACULTY, SCHOOL, DEPARTMENT

QOS Management Framework for Reliable Mobile ad hoc Networks

TITRE DE LA THÈSE / TITLE OF THESIS

Hussein Mouftah

DIRECTEUR (DIRECTRICE) DE LA THÈSE / THESIS SUPERVISOR

CO-DIRECTEUR (CO-DIRECTRICE) DE LA THÈSE / THESIS CO-SUPERVISOR

EXAMINATEURS (EXAMINATRICES) DE LA THÈSE / THESIS EXAMINERS

Nirwan Ansari

Azzedine Boukerche

Roshdy Hafez

Ramiro Liscano

Gary W. Slater

Le Doyen de la Faculté des études supérieures et postdoctorales / Dean of the Faculty of Graduate and Postdoctoral Studies

QOS MANAGEMENT FRAMEWORK FOR RELIABLE MOBILE AD HOC NETWORKS

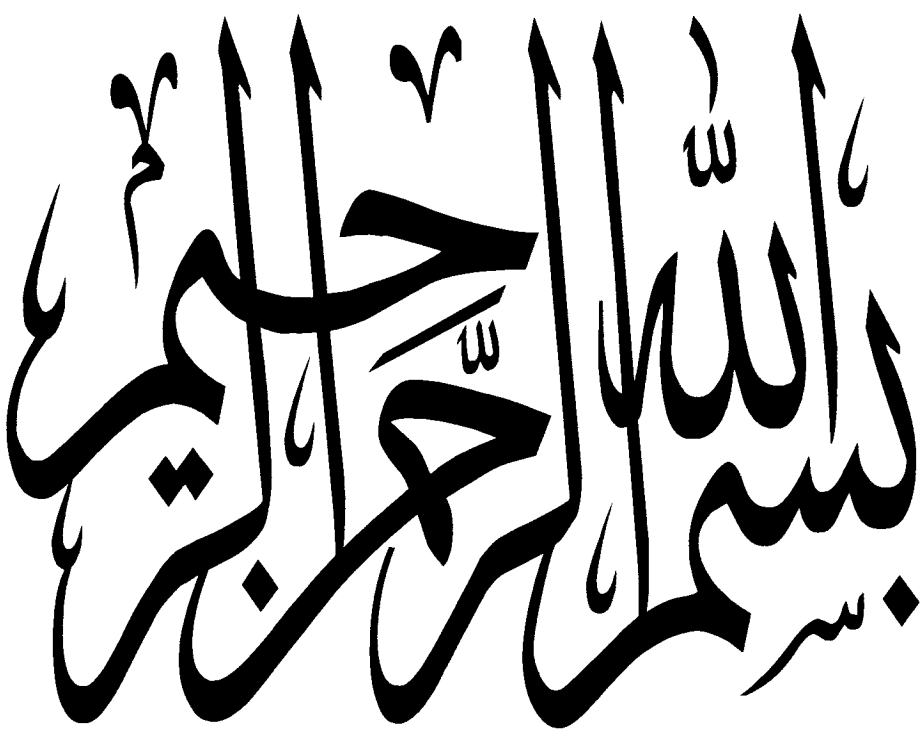
By

Mohammed Saeed Al-kahtani

Thesis submitted to the Faculty of Graduate and Post-Doctoral Studies
in partial fulfillment of the requirements for the
Degree of Doctor of Philosophy in
Electrical and Computer Engineering

Ottawa-Carleton Institute for Electrical and Computer Engineering
School of Information Technology and Engineering
University of Ottawa
Ottawa, Ontario, Canada

Your file Votre référence
ISBN: 0-494-10941-6
Our file Notre référence
ISBN: 0-494-10941-6


بسم الله الرحمن الرحيم

ABSTRACT

This thesis is primarily concerned with the design of a Quality-of-Service (QoS) management framework to enhance the stability and support reliability in the cluster-based mobile ad hoc networks (MANETs). This framework is based on providing a secondary clusterhead (SCH) for each clusterhead, which we call here primary clusterhead (PCH). This SCH, which is a regular member node, is identified and assigned by its PCH to be the future leader of the cluster. The SCH will be triggered to be the PCH when the former PCH can no longer be a clusterhead.

To enhance the cluster stability, this framework introduces a new protocol to reform the cluster, namely the Smooth and Efficient Re-Clustering (SERC) protocol. In SERC, since the future clusterhead is known by the cluster members, the cluster leadership will be transferred smoothly and the cluster will be reformed immediately with no need to invoke the clustering algorithm. Also, since the member nodes are associated to the cluster with its subsequent clusterheads, the cluster looks stable to the other clusters. Hence, the smooth clusterhead transfer from a node to another aims at increasing the cluster residence time, which will sustain the stability of the network, decrease the clustering communication overhead, and minimize the time spent by each node to join or to reform a cluster.

To support the route reliability, this framework provides a new multipath routing protocol, which is named a Localized Cluster-based Rerouting and Resource Reservation Protocol (LC3R). In LC3R, the main route can be established

through the PCH chains, while the backup route can be established through the SCH chains. This protocol intends to improve the performance of MANETs by increasing the route lifetime to enhance the network stability and by developing a reliable cluster-based routing protocol to support the QoS requirements.

In this study, we describe the general strategies and structures of SERC and LC3R and demonstrate their effectiveness via analysis and simulation. This study includes a comprehensive set of performance evaluations of a MANET environment. The results show how these approaches can achieve a high packet delivery rate and a low packet delivery time with no more rerouting and reclustering communication overhead. To the best of our knowledge, this is the first work that proposes two clusterheads for each cluster, which is the main contribution of this work.

ACKNOWLEDGEMENTS

In the name of Allah, The Most Gracious The Most Merciful. I am praising Allah (Subhanallahu Wa Taala) for His Bounty in the ability to finish this thesis, which without His Willing and Permission nothing of this work would have been existed and useful.

This dissertation would not have been possible without the help of many people. I would like to take this opportunity to express my deep appreciation to all those who helped me in this arduous but extremely rewarding process.

First, it is a pleasure to express my sincere and deepest heartfelt gratitude to my supervisor Prof. Hussein T. Mouftah. He went beyond the limits of his duties to provide guidance, support, and encouragement at every stage of this research endeavor. His guidance, both in terms of technical advice on my research and in terms of professional advice, was invaluable. He has influenced me the most as a researcher, and I hope I am able to live up to his high standards. I am very proud of being one of his students.

Working at School of Information Technology and Engineering (SITE) for the majority of my graduate life was a good learning experience. I would like to thank everyone at the SITE for making my stay a pleasant one.

Research funding plays an extremely important role in a graduate students life, and I was no different. I am indebted to the GOTEVOT, Kingdom of Saudi Arabia for their financial support. There is no doubt that this research could not have been completed in such a timely fashion without their support. Also, I wish

to express my thankfulness and gratitude to the Saudi Arabian Cultural Bureau in Canada for their support and guidance during my stay in Canada.

I am extremely grateful to my wife Hanan Al-Hemali for her love, encouragement, support and comfortable atmosphere during the long hard work of this thesis, to my sons Khalid, Ziyad, and Reyadh for their lovely smiles that can relieve any kinds of tiredness. I am also extremely grateful to my father (Rahemahu Allah), my mother, all brothers and sisters, and all relatives and friends for their prayers and kind thoughts on my behalf. I am grateful to Allah for blessing me with such wonderful family and friends; without their support and encouragement, more than anything else, I would have never reached this stage in my life. The expression of my gratitude for them is beyond words!

To my wife Hanan

To my sons Khalid, Ziyad, and Reyadh

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

GLOSSARY

ABR	Associativity-Based Routing
ALP	Link-State Protocol
ALP	Adaptive Link-State Routing
AODV	Ad hoc On-demand Distance Vector
AODV-BR	Ad hoc On-demand Distance Vector Backup Routing
AODVM	AODV-Multipath
AP	Access Point
BMT	Border Mobile Terminal
BPL	Battery Power Level
CBR	Constant Bit Rate
CEDAR	Core-Extracted Distributed Ad hoc Routing
CGSR	Clusterhead Gateway Source Routing
CH	Clusterhead
CSMA/CA	Carrier Sense Multiple Access/Collision Avoidance
DCA	Distributed Clustering Algorithm
DiffServ	Differentiated Service
DSDV	Destination Sequenced Distance Vector

DSR	Dynamic Source Routing
EL	Energy Level
FQMM	Flexible QoS Model for MANET
FSR	Fisheye State Routing
GPS	Global Positioning System
GSR	Global State Routing
HEL	Highest Energy Level
HMRSVP	Hierarchical Mobile RSVP
IBSS	Independent Basic Service Set
INIR	Intermediate Node Initiated Rerouting
IntServ	Integrated Service
ISSLL	Integrated Services over Specific Link Layer
LANMAR	Landmark Ad Hoc Routing
LC3R	Localized Cluster-based Rerouting and Resource Reservation Protocol
LCA	Link Cluster Algorithm
LCC	Least Clusterhead Change
LCC-LID	Least Cluster Change - Lowest ID
LMR	Lightweight Mobile Routing
LMR	Lightweight Mobile Routing
LRR	Least Resistance Routing
MAC	Medium Access Control

MANET	Mobile Ad hoc Network
MAPLE	Mobility-Aware Proactive Low Energy
MDSR	Multipath Dynamic Source Routing
MPLS	Multi-Protocol Label switching
MRSVP	Mobile RSVP
MT	Mobile Terminal
NIC	Network Interface Card
OLSR	Optimized Link State Routing
OSPF	Open Shortest Path First
PA-VBS	Power-Aware Virtual Base Station
PCH	Primary Clusterhead
QoS	Quality of Service
RABR	Route-Lifetime Assessment Based Routing
REER	Route Error
ROAM	Routing On-demand Acyclic Multipath
RREP	Route Reply
RREQ	Route Request
RSRV	Resource ReSerVation Protocol
SCH	Secondary clusterhead
SERC	Smooth and Efficient Re-Clustering
SIRR	Source Initiated ReRouting

SMR	Split Multipath routing
SSA	Signal Stability-Based Adaptive
STAR	Source Tree Adaptive Routing
TDR	trigger-based distributed Routing
TORA	Temporarily Ordered Routing
VBS	Virtual Base Station
VBS-O	Virtual Base Station On-demand
VPN	Virtual Private Network
WCA	Weighted Clustering Algorithm
WEAC	Warning Energy Aware Clusterhead
WRP	Wireless Routing Protocol
ZRP	Zone Routing Protocol

Chapter 1

INTRODUCTION

1.1 Background

Since the Internet was introduced to the general public at the beginning of the 1990s, its traffic has been rapidly increasing with an explosive growth of its users. As we already know, the current Internet was implemented to offer what is called "Best Effort" service, in which all data are treated equally without any reservations or priority levels. With the growth of the Internet and its emergence with the multimedia applications, it has become necessary to change this approach. Since real time applications are delay-sensitive and need more bandwidth, they should be treated in a different fashion. Increasing the network resources is not the solution; the solution is to use it in a more efficient manner. Unfortunately, most of the present Internet services are still unable to provide the required communication quality. Therefore, the next item on the Internet's agenda is to add the features and characteristics that were missing from its original design such as the provision of the Quality of Service (QoS) [ARM00, WAN01].

In the last ten years, wireless technology, as a new trend in the communication industry, has reached most locations on the face of the earth. This convergence between the wireless networks and the Internet makes new challenges for the Internet evolution, as well as the wireless networks. As it is already known, the wireless network is not able to achieve performance similar to that of the wired network

because of the bandwidth, quality, and power limitations, which face that technology. QoS parameters for typical applications are bandwidth, packet delay, packet loss rate, and jitter. Bandwidth is the most obvious difference between the wired and the wireless domains, the wireless being much slower than the wired one. The wireless network typically experiences longer delays than the wired. The wireless link is easily affected by fading and interference; so it is considered to be poor and unreliable, and its quality is changing with time. Wireless communication is also affected by the fact that the mobile units run on batteries. Therefore, delivering hard QoS guarantees in the wireless networks is very difficult [EVA02, WIS02].

In a region where there is no fixed network infrastructure, or where the network infrastructure is costly and time consuming to build up, mobile ad-hoc networks (MANETs) can provide network connectivity [TOH01, LLY02]. MANETs consist of a collection of mobile wireless nodes that dynamically create a network among themselves without using infrastructure or administrative support. The necessary control and administration functions on such networks are only accomplished by the interactions among their constituent nodes. Due to these features, MANETs are often described using other expressions like wireless multihop networks, self-organizing networks and mesh networks. In such networks, mobile terminals could develop the role of an end terminal that is the source or destination of data traffic flows and a mobile router that provides traffic relaying functionalities. The routing provision in MANETs can be difficult due to the error-prone nature of the wireless links, the unpredictability of the network topology, and the limited power autonomy of each node. Therefore, due to wireless link characteristics, user mobility, and battery power limitations in the MANET, end-to end connections are frequently interrupted and they need to be rerouted in the network while preserving their QoS requirements. Route discoveries due to frequent link changes increase the amount of control messages throughout the network. Route interruptions cause

slow route reconfigurations and packet loss retransmissions. This is very inefficient in an environment where resources such as radio bandwidth and battery power are limited. Therefore, the main challenges facing the MANET are the route failures and topology changes.

Therefore, among the many challenges for MANET designers, finding and maintaining stable routes are critical issues. Almost all the routing approaches in literatures can be classified into: *Proactive routing* and *Reactive routing*. Proactive routing is table-driven based and requires an overhead for maintaining and exchanging information about the state of the network to build and update those tables. On the contrary, since reactive routing tries to find paths on-demand, it causes more delay to do so. Therefore, both proactive and reactive approaches are not scalable. Clearly, some sort of a compromise needs to be achieved between those strategies. *Clustering* strategies tend to do just that. Most of the *cluster-based routing algorithms* tend to use proactive approaches within the cluster and reactive approaches for inter-cluster routing.

The concept of dividing the geographical region into small zones has been presented as clustering in the literature. The idea of clustering in ad hoc networks is not new. Clustering basically transforms a physical network into a virtual network of interconnected clusters or groups of nodes. Those clusters are dominated by clusterheads and connected by gateways or border mobile terminals. Any node can be a clusterhead if it has the necessary functionality, such as processing and transmission power. The nodes that register with the nearest clusterhead become members of that cluster. Clustering in MANET has been used in order to facilitate management, to improve routing efficiency, to support QoS, and to save power consumption. The main goal is to find the clusterhead nodes and partition the network into clusters. These nodes take on a special role in managing routing information. Therefore, stability starts from partitioning the network into clusters.

Once the clusters are established, the route stability between a source-destination pair is based on the clusterheads involved in that route. However, the frequent changes of the clusterheads affect the performance of all the other scheduling, routing, and signaling protocols. Due to the dynamic nature of the mobile nodes, their association and disassociation to and from clusters perturb the stability of the network and the problem becomes worse if these nodes are clusterheads. Eventually, the clustering stability in MANET would be significantly affected.

1.2 Motivation and Objectives

This work is motivated by the lack of cluster stability and route reliability in multihop cluster-based mobile ad hoc networks. Our objectives are to keep the cluster residence time as long as possible in order to sustain the network stability, and to keep the routes as long as possible in order to support the network reliability. In most clustering algorithms, a clusterhead is elected based on an agreed upon rule. Most of these algorithms are interested in how the clusterhead is elected and which weighting factors are used. The cluster residence time for such a node in such a cluster has not been taken into account. We believe this metric can give an indication to how stable and efficient the clustering algorithm is. The cluster stability will reduce the clustering overhead as well as the routing overhead caused by frequent full routing updates.

The traditional design of MANET QoS routing mechanisms has focused on providing a single feasible path to route data between a source-destination pair. A feasible path is a path that has the required resources and can satisfy the QoS constrains. This can be accepted when the network, wired or wireless, is stable and can provide acceptable performance. However, it cannot be expected to perform effectively in a dynamic environment where route failures occur frequently.

In cluster-based infrastructure protocols, the network reliability may also be affected due to single points of failure in these clusterhead critical nodes. Therefore, loosing packets, increasing delay, and increasing overhead caused by route failures and rerouting must be explicitly considered when designing routing protocols for MANETs. Hence, in MANETs, it is required to maintain redundant route information and establish an alternate route promptly in the failure of the initial route. The path redundancy scheme, which establishes multiple routes for the same connection and each packet is sent over each route based on its priority, is suggested to increase the probability of successfully delivering the data packets as well as to solve the problems occurred by those frequent interruptions. However, this scheme wastes the resources of the already resource-scarce wireless environment. In addition to route redundancy, it is important to localize the effect of the failure to reduce the control traffic overhead and the route reconfiguration time.

1.3 Thesis Contributions

The objectives above have been addressed throughout the thesis. Our primary contributions and accomplishments include the following:

- A new clustering algorithm for MANETs, namely the Smooth and Efficient Re-Clustering (SERC) protocol, has been introduced to support the cluster stability and to reduce the clustering overhead. The basic idea of this protocol is the election of a primary clusterhead (PCH) and a secondary clusterhead (SCH) for each cluster. The SCH works as a backup for the PCH and the future leader for the cluster. In SERC, the cluster member that was supervised by the PCH and was being heard by the SCH will stay associated with the cluster. This member will be associated with the SCH immediately when its PCH can no longer be a clusterhead. Once the SCH is triggered to be a PCH, a new SCH is assigned by the new PCH. The

clusterhead smooth transfer from the PCH to the SCH, when needed, will give the cluster more lifetime and will save the re-clustering communication overhead. The SCH will reform the cluster very fast to save the overhead caused by the re-computation of the clusterheads and the frequent messages that are exchanged between the participating nodes.

- A Localized Cluster-based Rerouting and Resource Reservation (LC3R) protocol has been proposed to enhance the reliability of the network with no waste of bandwidth and in a reasonable time. LC3R utilizes the SERC infrastructure protocol with packet redundancy over its PCH and SCH chains. The SCH works as a backup for the PCH and is the future leader for the cluster. The main route can be established through the PCHs, while the pack-up route can be established through the SCHs. The route and its resource reservations would be recovered locally by the SCH once it becomes a PCH. By using this scheme, the route will be maintained locally and the resources will be reserved locally within the affected cluster, and there is no packet loss during the route failure. Hence, localization routing and localization resource reservation is expected to improve the performance.

- The performance of the SERC and LC3R protocols has been evaluated via mathematical analysis and computer simulations.

In the network with large population, the performance evaluation results show that the SERC protocol improves the cluster residence time for each node. This metric is so important because it has a direct correlation with the cluster stability as well as the algorithm scalability. Also, the average-cluster-merge-operation latency and the number of messages needed to reform the cluster are reduced. Therefore, the saved time will be spent in sending or forwarding packets instead of just building or joining clusters. This reduction comes as a result of deploying

SCH for each PCH and using the *Hello* messages to transfer the leadership from PCH to SCH with no need for more clustering communication overhead.

In this study, we show how the LC3R approach can increase the route life time, reduce the routing overhead, decrease the packet delivery time, and achieve a high packet delivery rate with no extra overheads. The delay caused by the rerouting and the resource reservations will be reduced in the presence of route failures because the route life time is increased. Using a virtual secondary path will increase the packet delivery ratio and decrease the packet loss ratio. Consequently, it will improve the network reliability with no waste of bandwidth.

The infrastructure formation in the SERC/LC3R framework demonstrates quick response to topological changes in MANETs. Quick route reconfiguration and packet redundancy can improve performance measures such as packet delivery fraction and average end-to-end packet delay. Additionally, the framework is scalable to networks with large populations of mobile nodes. It outperforms current infrastructure creation protocols in stability and reliability. The Warning Energy Aware Clusterhead (WEAC) infrastructure creation protocol [SHE03] and the Virtual Base Station On-demand (VBS-O) protocol [SHE02a] are used as a background of this framework. These protocols have been chosen because of their features in regard to the load balancing and the energy saving. SERC/LC3R adopt the basic concept of WEAC, but several modifications are made to take advantage of the secondary clusterhead, which is the main contribution of this work.

1.4 Thesis Outline

The rest of this dissertation is organized as follows. In the next chapter, we present a review of the significant work done in relation to these research problems. In Chapter 3, we describe in details the SERC and LC3R protocols and discuss their features and how they work. The network model, the mathematical analysis,

and the protocol properties are derived in Chapter 4. The simulation and the models description, along with results demonstrating the effectiveness of these new approaches, are shown in Chapter 5. Finally, in Chapter 6, we close this study with the conclusions reached in this work and the suggestions we propose for future research.

Chapter 2

PREVIOUS WORK ON RELIABLE QOS IN MANETS

2.1 Introduction

This chapter presents a brief coverage of a wide range of concepts and systems that are related to this research problem. An overview of the MANETs routing protocols will be presented and discussed in Section 2.1. In Section 2.2, we discuss why the IP QoS Models cannot be applied directly on MANETs. Therefore, the QoS support in MANETs will be presented in Section 2.3, highlighting the models, the signalling models and the QoS routing protocols. Finally, a summary is presented in Section 2.4.

2.2 MANETs Routing Protocols

QoS communications in MANETs is highly dependent on routing protocols and medium access control (MAC) protocols. In addition, complying with QoS guarantees imposes the use of a MAC method, which in its turn guarantees the successful transmission of packets under high mobility and/or heavy load circumstances, and hence meets the QoS constraints dictated by the communications application. Thus, routing protocols are responsible for maintaining and reconstructing the routes in a timely manner, as well as establishing the durable routes. Due to the mobility and the dynamic status of the wireless links, the routers must

implement adaptive algorithms that are responsive to the changes in the network topology, without over-utilizing the network resources. Thus, routing control packets must be utilized efficiently to deliver data packets and should be generated only when necessary. Reducing the control overhead can make the routing protocol efficient in bandwidth and energy consumption. The routing protocols that have been designed for MANETs can be classified in four categories according to the way routes are created and maintained:

- Table-driven Routing Protocols
- On-demand Routing Protocols
- Cluster-based Routing Protocols
- Constraint Routing Protocols

2.2.1 Table-Driven Routing Protocols

Table-driven routing protocols are characterized by their attempt to maintain a current picture of the network topology autonomously. They lack distinct multiple modes of operation such as discover phases and teardown phases. They are continually in a maintenance mode of operation and require memory proportional to hold all destinations of interests. They are proactive protocols which try to match the conventional routing algorithm (the link state and distance vector) ideas to the wireless environment by taking the above limitations into account and trying to reduce or eliminate them. Protocols of this type include the Wireless Routing Protocol (WRP) [MUR96], the Destination Sequence Distance Vector (DSDV) routing protocol [PER94], the Least Resistance Routing (LRR) [PUR93], and the protocol by Lin and Liu [LIN99]. Some protocols that are based on link state algorithms [MCQ80, MOY98] such as Global State Routing (GSR) [CHE98], Fisheye State Routing (FSR) [PEI00a], Adaptive Link-State Protocol (ALP) [PEI00a], Source

Tree Adaptive Routing (STAR) [ACE99], Optimized Link State Routing (OLSR) protocol [CLA00], and Landmark Ad Hoc Routing (LANMAR) [PEI00b] fall into this category.

In table-driven protocols, there are two disadvantages. First, topology update is taken place even it is not needed, which is considered overhead. Second, topology changes are propagated throughout the network, thus affecting the network globally. However, optimizations and tradeoffs are available to minimize these shortcomings. For example, updates that are triggered due to some changes can be moderated so that in effect the node will "listen, before speaking". This will let the node filter the updates it hears so that only the best updates are propagated. Also advertising only routes that are used can help to alleviate the routing overhead. It is plausible that not every destination will need to receive traffic. Routes that are needed and unknown can be discovered through some sort of a query process. Requiring all routes to be discovered on demand is the basis of the second category of ad-hoc routing protocols.

2.2.2 On-Demand Routing Protocols

On-demand routing protocols or reactive routing protocols create routes only when desired by the source node so it is called "On-demand". This type of routing discovers paths on needed basis. They are characterized by a route discovery process and usually also have a route error detection and a route maintenance process. This style of routing can incur large initial delays, and minimize route changes to a local scope. This approach requires some sort of explicit communicational handshaking between the application and the routing algorithm. The application layer needs to communicate to the routing layer about finding destinations, and the routing algorithm needs to return error or confirmation messages. The network layer may also have to provide buffering while routes are found and some flow control is performed to prevent the buffers from overflowing.

This type of routing has two major components: route discovery and route maintenance. The route discovery function requires a source to use some form of flooding. The transit nodes, upon receiving a query, "learn" the path to the source and enter the route in their forwarding tables. The destination node responds to the source using the path traversed by the query. A route can then be established between source and destination. Route maintenance is responsible for reacting to topological changes in the network and its implementation differs from one algorithm to the other. On-demand routing protocols [MAL99] have been proposed only for ad hoc networks. Numerous protocols of this type have been proposed. Lightweight Mobile Routing (LMR) [COR95], Dynamic Source Routing (DSR) [JOH96, DAS00, ROY99a], Ad-Hoc On Demand Distance Vector (AODV) routing [PER99, PER00a, ROY99a, ROY99b], Temporarily Ordered Routing Algorithm (TORA) [PAR01, PAR97], Associatively-Based Routing (ABR) [TOH01], Signal Stability-Based Adaptive (SSA) routing [DUB97], Routing On-demand Acyclic Multipath (ROAM) algorithm [RAJ99], Multipath Dynamic Source Routing (MDSR) [NAS01], Relative Distance Micro-discovery Ad Hoc Routing (RD-MAR) protocol [NAS01], and Route-Lifetime Assessment Based Routing (RABR) protocol [NAS01] are typical on-demand routing protocols. A good review, description, and comparison between the existing table-driven and on-demand Ad Hoc routing protocols is presented in [ROY99a] which it is recommended if more information is needed.

2.2.3 Cluster-Based Routing Protocols

With the increase in size of the networks, flat routing schemes do not scale well in terms of performance. Aggregated routing information is the key to Internet scalability. In multihop mobile wireless networks, clustering is the method, that aggregates nodes into groups (clusters) to provide a suitable framework for the

development of important features (i.e. routing bandwidth allocation, mobility, security and topology management), in order to reduce routing messaging overhead. The concept of dividing the geographical region into small zones has been presented as clustering in the literature. Clustering basically transforms a physical network into a virtual network of interconnected clusters or groups of nodes. Any node can be a clusterhead if it has the necessary functionality, such as processing and transmission power. Nodes register with the nearest clusterhead and become members of that cluster. Clustering in MANET has been used in the past in order to facilitate management, to improve routing efficiency, to support QoS and to save power consumption [CHI97, HAS01, JIA99, SAF01a]. Recently there has been interest in hierarchical architectures for MANET. In hierarchical MANET architectures, conventional cellular architectures are mapped into MANETs to form the strong ground for solving the various communication problems. One such problem is routing.

Most clustering algorithms are heuristic in nature and their aim is to generate the minimum number of clusters. In [GER95], two clustering algorithms were proposed: the *Highest-Degree* which takes into consideration the degree of a node, and the *Lowest-ID* which assigns a unique ID to each node and chooses the node with the minimum ID as an CH. In the *Linked Cluster Algorithm* (LCA) [BAR81], a node becomes the CH if it has the highest identity among all nodes. To generate a smaller number of clusters, LCA was improved by LCA2 algorithm. However, generating a minimum number of clusters might not ensure minimum energy usage. The *Max-Min D-hop* clustering strategy proposed in [AMI00] generates fewer clusters than the LCA and LCA2 algorithms.

The *Distributed Clustering Algorithm* (DCA) uses weights associated with nodes to elect CHs [BAS99]. These weights are generic and can be defined based on the application. The *Weighted Clustering Algorithm* [CHA02] (WCA) tries to

capitalize on the advantages of the other algorithms. It uses the degree of a node, its transmission power, its mobility and its battery power to calculate the combined weight. The algorithm aims at computing the minimum number of CHs in order to maximize the resource allocation. Since its election procedure is very expensive, it is invoked as rarely as possible. It can be noted that the existing heuristics are all special cases of WCA.

McDonald and Znati [MCD99] designed a (α,t)-clustering algorithm that adaptively changes its clustering criteria based on the current node mobility. This algorithm determines cluster membership according to a cluster's internal path availability between all cluster members over time. Another clustering algorithm based on a mobility metric is proposed in [BAS01]. The distributed clustering approach, named $(p,\ t,\ d)$-clustering [SIV04], is based on intelligent mobility prediction that enables each mobile terminal to anticipate the availability of its neighbors. This approach presents a scalable way to predict the mobility, and thus the availability, of mobile terminals, achieved with the introduction of geographically-oriented virtual clusters. This approach of stability improvement depends on the accuracy of its mobility prediction.

The authors in [PAL04] proposed a framework for mobility-aware pro-active low energy (MAPLE) clustering in ad hoc mobile wireless networks. The MAPLE approach addresses the problem of clustering in a medium-access control framework and enables pro-active and energy-efficient clustering exploiting the node mobility information. In particular, for pro-active clustering, a method to exploit the link level information to estimate the mobility pattern of the wireless nodes and the cost of using a wireless link in terms of required transmission power has been introduced. A channel reservation technique is used to reduce the number of contentions among the nodes while accessing the channel during cluster formation. Simulation results show that the proposed framework results in superior clustering performance in

terms of control overhead, average number of link failures and load distribution compared to other clustering approaches proposed in the literature such as the LCC-LID (Least Cluster Change - Lowest ID)-based clustering.

Most of those algorithms are interested on how the CH is elected and which weighting factors are used. The communication overhead for the formation and maintenance of the cluster has not been taken into account. There has been little published work that assesses analytically the communication overhead incurred in hierarchical routing. The authors in [YOU04] showed how this cost can be minimized by electing the CH randomly with no exchanging messages. However, the nodes still need to send messages to join the CH. In [SUC02], a study provides a theoretical upper bound on the communication overhead incurred by a particular clustering algorithm (the ALCA) for hierarchical routing. It is demonstrated that the average clustering overhead generated per node per second is polylogarithmic in the node count. Hence, the clustering overhead is proportionally increased with the number of nodes. Therefore, the aim of this work is to reduce this communication overhead that caused by cluster formation and maintenance.

The aggregation of nodes into clusters under clusterhead control provides a convenient framework for the development of efficient protocols both at the MAC layer (e.g., code separation among clusters channel access), bandwidth allocation, and at the network layer (e.g., hierarchical routing [GER95]). Clustering provides an effective way to allocate wireless channels among different clusters. Some examples of the cluster-based routing protocols in the literature are CGSR [ROY99a, CHI97], and VBS [HAS01].

2.2.4 Constraint Routing Protocols

In wireless mobile ad hoc networks, the infrastructure function is very important. It lays the ground for any routing protocol to maximize its throughput. In

the cluster-based routing protocols, the clusterhead is used as a base station as in cellular network. It is used for routing as well as for network management and for resource allocation among nodes in its cluster. In most of the published clustering based approaches, the selection of a clusterhead is on based different factors like cluster ID or degree of connectivity. If traditional clustering approaches like lowest/highest ID or highest connectivity are applied, the same node will be elected as a clusterhead every time, resulting in this node to drain its power very fast. As it is already known, battery power is a strategic resource for mobile devices. Therefore, conservation of power and power-aware routing must be taken into consideration in order to realize a dynamic and adaptive networking concept for future wireless communication systems. Hence, rotation of the role of the clusterhead among other members in the cluster based on the battery power is suggested to balance the load and then to save the power and increase the network lifetime.

The Warning Energy Aware Clusterhead (WEAC) [SHE03] and the Power Aware Virtual Base Station (PA-VBS) [SAF01b] are two novel infrastructure-creation protocols for MANETs which use power constraint at the clusterhead election. WEAC and PA-VBS do not drain all the battery power of the clusterheads since clusterheads are allowed to become saturated, and hence do not accept any more merge requests, on the contrary to the other clustering protocols. WEAC and PA-VBS use the concept of battery power level but with different mechanisms. However, unlike PA-VBS, clusterheads in WEAC use a warning threshold, which gives their mobile terminals (MTs) an indication to look for another clusterheads.

In this work, we have adapted the Warning Energy Aware Clusterhead (WAEC) infrastructure creation protocol [SHE03] and the Virtual Base Station On-demand (VBS-O) [SHE02a] routing protocol. The justification of the use of the WEAC protocol in this study are provided in Section 3.2.1. In Appendix A, description of the WEAC infrastructure protocol and the VBS-O routing protocol are presented.

2.3 QoS Models in IP Networks

Quality of Service (QoS) [ARM00, WAN01, EVA02, WIS02], with respect to computer networks, is a set of service requirements provided to a certain traffic by the network to meet the satisfaction of the traffic user. QoS requires a capability in the network to be able to treat some selected traffic with better or different service than what other traffic receives. Any QoS mechanism does not create an additional bandwidth for the selected network; instead, it uses the available bandwidth in a way that the network can provide the maximum requested QoS with the maximum bandwidth available for that traffic during the session. The network traffic is selected based on classification of packets so that some packet flows can be treated better than the others.

In the Internet networks, the only service model traditionally provided was the best-effort service. This was acceptable when the connectivity and data delivery were the main targets. With the continuous growth of Internet services, including the convergence of real time application like voice over IP (VoIP) and multimedia, QoS is required to guarantee their operation. To meet these demands, the network must be enhanced with new technologies that can offer the capabilities to provide or control its quality of services. The IETF has defined different IP QoS mechanisms or architectures [ARM00]. The most prominent among them are the integrated services (IntServ) [BRA94] and the differentiated services (DiffServ) [BLA99]. These services differ in their QoS strictness which describes how tightly the service can be bound by specific bandwidth, delay, jitter and loss characteristics. This section briefly examines these mechanisms with focus on the wireless and the mobile applications.

2.3.1 Integrated Services

The philosophy behind IntServ [ARM00, BRA94] is that routers must be able to reserve resources for individual flows to provide QoS guarantees to end-users. The IntServ QoS control framework supports two additional classes of services besides "best effort": Guaranteed Service and Controlled Load Service. Both services must ensure that adequate bandwidth and packet processing resources are available to satisfy the level of service requested. This must be accomplished through active admission control. In the IntServ, the most needed component is the signaling protocol that is used to set up and tear down reservations. Resource ReSerVation Protocol (RSVP) [BRA97] is used by the IntServ as a signaling protocol. To achieve its goals, IntServ requires per-flow state to be maintained throughout the network. Unfortunately, the biggest known problem with the per-flow approach is its poor scalability because it incurs huge storage and processing overhead at routers in the presence of thousands of flows. Also, it is well known that reservations need to be regularly refreshed, thus consuming valuable resources, especially in bandwidth scarce environments such as the wireless. Other problems include the fact that the call admission procedures and the routing scheduling schemes are complex and rely heavily on the per-flow state.

2.3.2 Differentiated Services

DiffServ [ARM00, BLA99], on the other hand, aggregate multiple flows with similar traffic characteristics and performance requirements into a few classes. This approach pushes complex decision making to the edge routers or Ingress routers, resulting in less processing load on core routers and thus faster operation. According to this mechanism, QoS information is carried in-band within the packet in the Type of Service (TOS) field in IPv4 header or Differentiated Service (DS) field in IPv6. Classification, rate shaping, and policing are done at the edge routers

and packets are mapped onto service levels through static service level agreements. Per-hop queuing and scheduling behaviors, or simply per-hop behaviors (PHBs), are defined through that number of edge-to-edge services that might be built. Two service models have been proposed: Assured Service and Premium Service [ARM00, BLA99].

DiffServ appears to be a good solution to a part of the QoS problem as it removes the per-flow state and scheduling that lead to scalability problems in the IntServ architecture. Therefore, DiffServ is simpler than the IntServ and does not require end-to-end signaling. However, it provides only static QoS configuration, typically through service level agreements, as there is no signaling of the negotiation of QoS. Also, it cannot guarantee the end-to-end QoS.

2.3.3 IntServ Over DiffServ

It would be very interesting to study this framework for IntServ operation in the future over DiffServ networks [BER00]. This solution tries to conjugate benefits of both the IntServ and the DiffServ approaches to QoS provisioning. In fact, it achieves scalability due to the DiffServ aggregation in the core network, while keeping the advantages of end-to-end signaling by employing the IntServ reservation at the edges of the network. The reference architecture includes DiffServ region in the middle of two IntServ regions. The advantage of this solution is that it allows hop-by-hop call admission and per-flow reservation at the edge of the network where low densities make this the most practical way to achieve good-quality guarantees. In the core of the network, the scalable solution of DiffServ scheduling can be used where hard guarantees in QoS can be made on the basis of more probabilistic judgments. This approach has been proposed by the Integrated Services over Specific Link Layer (ISSLL) Working group at the IETF. The basic philosophy of the architecture is to remove as many computation intensive functions as possible from the backbone routers and push these functions towards the

edge routers. That way, the core routers would be free to do high-speed forwarding of the packets and they would remain simple to manage.

2.3.4 MPLS

In the conventional IP routing, the packets are forwarded independently in each router. Its forwarding algorithm is quite slow because of the processing time of the IP header. Also its route selection is based on the shortest path algorithm, and this restricts the capabilities to manage the resources of the network.

Multi-Protocol Label Switching (MPLS) [ROS01] as a new technology tries to overcome the problems of traditional IP routing. MPLS was originally presented as a way of improving the forwarding speed of routers, but it has capabilities that enable an operator to provide service differentiation. MPLS is a combination of switching and routing. It combines some of the features of ATM with IP. The basic idea is that routers perform flow classification, tunneling and QoS marking at the edge of the domain, while switches do high-speed forwarding in the core. MPLS introduces the concept of connection-oriented into the Internet. The MPLS nodes will route based on topology not traffic and they will set up flows on an end-to-end basis rather than hop-by-hop, which is done by the IP protocol. The key idea is that the control plan is separated from the forwarding plane and forwarding is based on label swapping, which replaces the incoming label with a new outgoing label on each packet. MPLS is inherently more scalable than VC-based technologies and because label swapping is so general, forwarding can take place over circuits, LANs and non-broadcast multiple access networks like ATM. Indeed, MPLS is becoming widely used as a network management or traffic engineering mechanism.

In an MPLS domain, a label is assigned to each packet at the ingress node of the domain. Then the packet is forwarded according to the label until it is received at the egress node of the domain. That means, the full IP header analysis

is done only once at the ingress node, which simplifies and accelerates the routing process. Each label can be associated with a Forwarding Equivalence Class (FEC), which means a group of packets that can be treated in the same manner. The binding between a label and an FEC is distributed by Label Distribution Protocol (LPD) [AND01]. IETF defines three Label Distribution Protocols, LDP, CR-LDP [JAM02], and RSVP-TE [AWD01]. MPLS supports two types of route selection: hop-by-hop and explicit routing. The strength of MPLS is in provisioning Traffic Engineering [AWD02, AWD99]. The use of explicit routes as an example, gives the ability to manage the network resources and support new services.

2.4 QoS Support in MANETs

The major challenge facing the QoS support in MANETs is the fact that the quality of the networks varies with time. Therefore, providing the traditional QoS models, IntServ and DiffServ, is insufficient. They require accurate link state and topology information, which is so difficult to maintain in this time varying and low capacity resource environments. However, QoS of MANET should benefit from the concepts and features of the existing models in order to come up with a model that can satisfy such networks. In this section, we review the current research on QoS support in MANETs. This includes QoS models, resource reservations signaling, QoS routing and some other related works.

2.4.1 QoS Models in MANETs

A Flexible QoS Model for MANET (FQMM) [XIA00] is the first QoS model designed especially for MANET. It takes the characteristics of MANETs into account and combines the advantages of both the IntServ and DiffServ. FQMM defines three types of nodes: an ingress node which sends data; an egress node which is a destination; and an interior node which forwards data to other nodes. Obviously, each node may have multiple roles. The provisioning in FQMM to allocate

and determine the resources at the mobile nodes, is a hybrid IntServ (per-flow) and DiffServ (per-class) scheme. Therefore, the traffic of the highest priority is given per-flow treatment, while other traffic is given per-class differentiation. The traffic conditioner, to mark, discard, and shape the packets based on the traffic profile, is placed on the ingress node. FQMM assumes that the smaller proportion of the traffic belongs to the highest priority class. Therefore, the scalability problem of IntServ is not an issue in FQMM. However, the DiffServ problem still exists because making a dynamically negotiated traffic profile is very difficult.

2.4.2 QoS Signaling

QoS signaling is used to set up, tear down, and renegotiate flows and reserve and release resources in the networks [WIS02]. It is useful for coordinating traffic handling techniques like shaping, policing, and marking. Signaling is important to configure uniform successful end-to-end QoS service handling across the network. True end-to-end QoS requires that all the network elements like switches, routers, and firewalls appropriately support QoS. The coordination of these components is done by QoS signaling. Therefore, QoS signaling information must be carried, interpreted, and processed by all the networks elements. Using the scarce resources in an efficient way while maintaining strong service guarantees, assumes that some reservation signaling will be required for real-time applications.

The signaling can be classified into two categories [WIS02]: in-band signaling, when it is carried with the data, and out-band signaling, when it is separated from the data in explicit control packets. The in-band signaling ensures that the information is always carried to each router that the data may visit; thus is useful when routes change frequently, as in mobile networks. However, the important point here is the fact that it requires an overhead which in some applications may reach 10%. Out-band signaling, as used in the telephone networks, is more easily

transported to devices not directly involved in data transmission, as in admission control servers.

The signaling may be soft in which case the reservation needs to be refreshed frequently. This makes it resilient to node failures. Conversely, a hard-state protocol can minimize the amount of signaling. However, care needs to be taken to remove the reservation that is no longer required.

2.4.2.1 RSVP

The Resource ReSerVation Protocol (RSVP) [ARM00, WIS02, BRA97] is a mechanism for signaling QoS across the network. It is a key element of both IntServ and ISSLL approaches which were described above. RSVP is an out-band signaling system operated in a soft-state mode. Also, it is possible to operate in a near-hard-state mode across any section of a network. Two types of messages, PATH and RESV, are used in RSVP to set up resource reservation states on the nodes along the path between the sender and the receiver. The PATH messages are generated by the sender and propagated through the network to the receiver, gathering information about the network. Each node that processes the message records the flow identifier and the address of the previous RSVP-enabled IP router. Once this message reaches the intended receiver, the receiver responds with a RESV message, which actually chooses the required QoS and establishes the reservation. Therefore, RSVP is known as a receiver-initiated resource reservation. This message is propagated back along the path that is stored in the selected PATH message. If the sufficient resources are available, a soft Reservation State is established in the router. Otherwise, an Error message is generated and sent back to the receiver.

In RSVP, maintaining a reservation when the MN moves between regions is a challenge. A scheme is needed to define how smooth this transition should be

since it affects the QoS reservation. The disruption of service significantly impacts the QoS. In the presence of micro-mobility, the situation will get worse. Efforts are underway to enhance the RSVP protocol in the mobile networks for transporting real-time applications. According to the original RSVP signaling protocol, it can not be applied directly in the MANETs because it is not aware of mobility. When the MN moves and then the topology changes, the prior reservations are no longer available. The impact of the mobility in the last-hop mobile IP networks have been investigated and resolved by RSVP Tunnel [TER99], MRSVP [BAD01] and Hierarchical MRSVP [TSE01]. However, these techniques are not suitable for the MANETs due to different concepts.

2.4.2.2 INSIGNIA

The first signaling protocol designed especially to deliver real-time application in the MANETs is INSGNIA [LEE00a, AHN99]. It is a lightweight protocol in term of bandwidth consumption because it is in-band signaling, which supports fast flow reservation and restoration. The signaling control information is carried in the IP option of every packet, which is called the INSIGNIA option. Since it is a per-flow based protocol, like RSVP, each flow state is managed over an end-to-end session in response to the topology and end-to-end QoS condition and in the soft-state method. The wireless flow management model at the mobile node includes a packet forwarding module, a routing module, an INSIGNIA module, an admission control module, a packet scheduling module and a medium access controller module (as shown in Figure 2.1). In this figure the position and the function of INSIGNIA in this model is illustrated as well. Coordinating with the admission control module, INSIGNIA allocates bandwidth to the flow if the resource requirements can be satisfied. Otherwise, if the required resources are not available, the flow will be degraded to the best-effort service. To achieve a fast responding to the topology

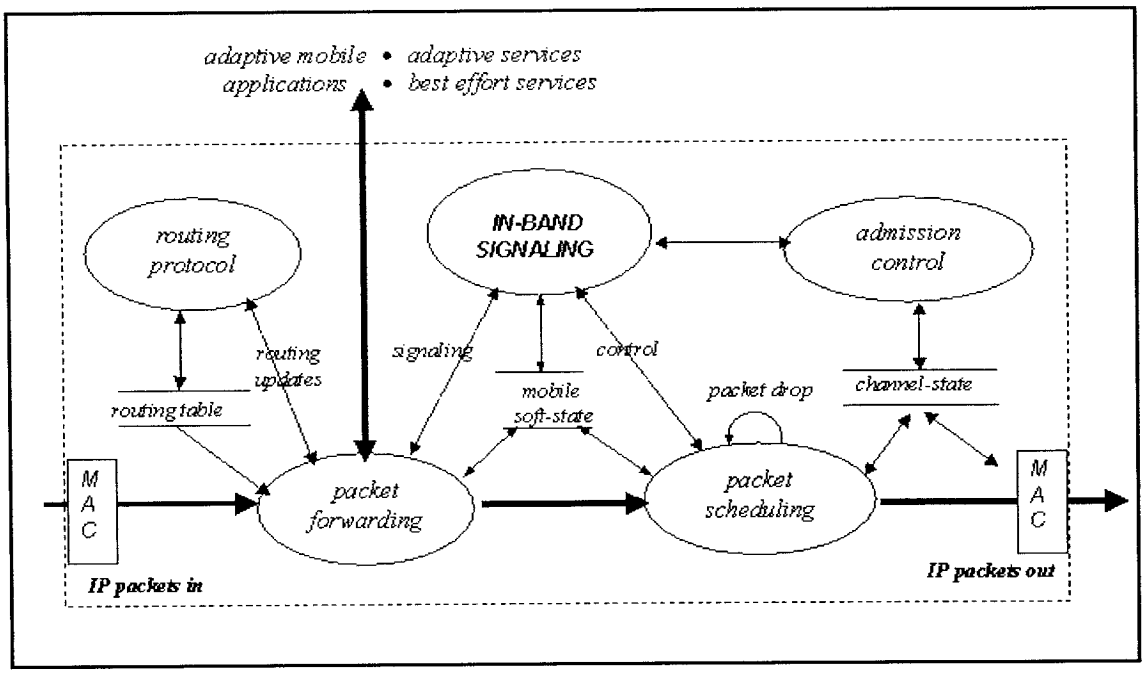


Figure 2.1: Flow Management Model in the INSIGNIA Protocol

changes and the end-to-end QoS conditions, INSIGNIA uses QoS reports to inform the sources node of the status of the real-time flows. The destination node actively monitors the received flows and calculates QoS statistical results such as loss rate, delay and throughput. The reports are sent to the source node periodically. By using this kind of feedback information, the source node can take corresponding actions to adapt the flows to observed network conditions.

Coordinating with other network components, INSIGNIA is an effective signaling protocol and can efficiently deliver real-time applications in MANETs. However, since the flow state information should be kept in the mobile nodes, the scalability problem may hinder its deployment in the future, similar to RSVP. Also, it is hard in some cases, to implement some functionality with in-band signaling. For example, when the established data flow is unidirectional from the source to the destination, sending the feedback control messages back to the source with in-band signalling with the session of the flow is impossible.

2.4.3 Multipath Routing Protocols in MANETs

The MANETs routing protocols usually employ single path routing. Multipath routing protocols employ a set of paths from the source to the destination so that the total volume of traffic may be divided and communicated via selected multiple paths. Therefore, there are mainly two approaches to use the multiple paths in MANETs. In the first approach, the data packets are transmitted along one path while the others are reserved as backup paths in case the used one is broken. When all possible paths are broken, a new multipath discovery procedure is initiated again. In the second approach, in order to balance the network load, the multipaths are used simultaneously as in dispersity routing [MAX93], which disperses the data traffic along different paths in order to achieve smaller average delay and delay variance. The dispersity routing protocols are classified as redundant and non-redundant routing protocols. In redundant dispersity routing, only one part of the multiple paths are used to transfer data and the remaining paths are used to transfer redundant information such as error correcting codes. In contrast, in non-redundant dispersity routing, all multiple paths are used to transmit data simultaneously.

Packet replication along predetermined multiple paths to the destination is used to seek communication reliability. Alternatively, data traffic is distributed along disjoint or meshed multiple paths - called selective forwarding. In [DE03], the authors studied the resource usage in two schemes: packet replication and selective forwarding. They have shown that packet-by-packet throughput with packet replication is higher than the throughput in selective forwarding. This is because sending a packet along multiple error-prone paths, rather than along a chosen route, surely increases the chance of successful arrival of at least a copy of the packet. However, to successfully route a message, packet replication has substantially high network resource requirements such as channel bandwidth and

battery power. Therefore, they observed that meshed multiple path routing with selective forwarding has an overall superior performance.

The MANET on-demand routing protocols, DSR [BOR98] and TORA [BOR98], have built-in capability to compute multiple paths. But either of them suffers from a different set of performance problems. DSR uses source routing, which can detect loops easily and can gather a lot of routing information per route discovery. However, the aggressive use of route caching and the lack of route expiration policy cause problems such as stale caches and reply storms. Hence, extending DSR to support multiple routes without addressing the caching problem may also multiply the problem, which limits the performance benefits of caching multiple paths and hurt performance in many cases [HU00, MAR01]. TORA, on the other hand, builds and maintains multiple loop free paths without the use of source routing. Also, it can detect network partitions. TORA uses an idea based on link reversals to recover from link failures. One major drawback of TORA is the maintenance overhead because of the requirement of reliable in-order delivery of routing control messages [BOR98].

AODV is a popular routing protocol that creates distance vector routing tables on-demand and requires a lower overhead as compared to DSR. The authors in [LEE00b] proposed an extension to AODV, called AODV-BR (AODV with backup routes). AODV-BR establishes the mesh and multipaths without transmitting any extra control message. The technique behind AODV-BR is to allow AODV to maintain backup routes at the neighboring nodes of a primary route. This can be done with no additional overheads. The idea is to avoid dropping packets on flight when the primary route fails. However, every node still has at most one route per destination just as in AODV. Also, the scheme does not perform well under heavy traffic networks.

A framework for reliable routing in MANETs and an AODV-multipath (AODVM) protocol have been proposed in [YE03]. The AODV has been modified to enable the discovery of multiple node-disjoint paths from a source to a destination. Instead of discarding the duplicate RREQ packets, intermediate nodes are required to record the information contained in these packets in a table which we refer to as the RREQ table. This reliable framework is based on reliable nodes that are physically more sophisticated in terms of being capable of combating fading, possessing better batteries and being physically more secure. These reliable nodes could then be allowed to participate in routing along multiple routes between the same source-destination pair. The revised objective is then to construct a sequence of reliable segments between the source and the destination. Nodes that join two segments have to be reliable nodes. A concatenation of reliable segments is called a reliable path. In this study, based on the randomized min-cut algorithm, the authors proposed a deployment strategy that determined the positions and the trajectories of these reliable nodes.

Split Multipath Routing (SMR) [LEE01] is an on-demand routing protocol that constructs maximally disjoint routes. SMR is based on DSR, but it introduces a different packet forwarding approach. While DSR drops duplicate Route Request packets, SMR allows intermediate nodes to forward the duplicate RREQs in order to find more disjoint routes. The destination node selects multiple disjoint routes and sends Route Reply packets back to the source via the chosen routes. Data traffic is split into multiple routes that are found during the route setup procedure. The SMR approach provides robust packet delivery and is beneficial in network load sharing, but causes out-of-order-delivery and considerable route setup overhead.

In [NAS99], the authors proposed an on-demand routing protocol that provides multiple redundant routes on the primary route which is taken by the first

query reaching the destination. Each intermediate node in the primary route can have a disjoint route to the destination. To accomplish this, a reply to the route request is targeted not to the source, but to the intermediate node. In the failure of the primary route, the upstream intermediate node nearest to the failed link forwards in-transit data packets on the alternate route if it has one. The protocol acts as a DSR if the primary route does not have any alternate route. This protocol can reduce the frequency of floods and route reconfiguration time by providing redundant routes. However, exploring redundant routes is confined to the primary route and this may limit the provision of enough redundancy and lead to excessive delay due to long alternate routes.

2.4.4 MANETs QoS Routing Protocols

As mentioned above, many authors have addressed the routing issues in MANETS from the best-effort service point of view. Although these protocols attempt to minimize the control and database maintenance overhead, they do not meet real-time QoS requirements. We believe that the routing protocols have to be adopted to tackle the delay and bandwidth constraints of real-time applications in MANET. At the same time the buffer and the signaling overhead should be managed in a proper way to optimize the resource utilization. In general, the notion of QoS has been proposed to capture the qualitatively or quantitatively defined performance contract between the service provider and the user applications. The QoS requirement of a connection is given as a set of constraints. The basic function of QoS routing is to find a network path which satisfies the given constraints [GRA98, MIR00]. In addition, most QoS routing algorithms consider the optimization of resource utilization. The problem of QoS routing is difficult because multiple constraints often make the routing problem intractable and the dynamic of the network state makes it difficult to gather up-to-date information in a large

network. Moreover, the QoS routing protocols should work together with resource management to establish paths that meet end-to-end QoS requirements through the network. Therefore, QoS routing is an essential part of the QoS architecture. Before any connection is established or any resources are reserved, a feasible path between a source-destination pair should be discovered. QoS routing is a routing mechanism under which paths for flows are determined on the basis of some knowledge of resources availability in the network, as well as the QoS requirements of the flows themselves. QoS routing is difficult in MANETs because of the following reasons [PER02, CHA99]:

- Because the QoS routing overhead is too high, in this already limited bandwidth environment, the MN should have a mechanism to store and update the link state information. Therefore, balancing the benefit of QoS routing against the bandwidth consumption is needed.

- Maintaining the precise link state information is very difficult because of the dynamic nature of the MANETs.

- Establishing a feasible path does not mean the required QoS should be ensured. The reserved resources may not be guaranteed because of the mobility induced route disconnection and power reduction of the MN. Therefore, QoS routing should rapidly find a feasible new route to recover the service.

Although, QoS routing in MANETs is relatively scarce due to the above reasons, some promising works have been done and shown a good performance. Below we introduce CEDAR, ticket-based probing, QoS over AODV, bandwidth routing and trigger-based distributed routing as examples to illustrate how to deal with the above difficulties.

2.4.4.1 CEDAR

The Core-Extracted Distributed Ad hoc Routing (CEDAR) [SIN99] protocol has been proposed as a QoS routing protocol for networks consisting of tens to hundreds of nodes. CEDAR dynamically establishes a core of the network, and then incrementally propagates link states of stable high bandwidth links to the nodes of the core. Route computation is on-demand and is performed by core hosts using local state only. CEDAR does not compute optimal routes because of the minimalist approach to state management, but the trade-off of robustness and adaptation for optimality is believed to be well justified in ad-hoc networks. The following is a brief description of the three key components of CEDAR:

- Core Extraction: A set of hosts is distributedly and dynamically elected to form the core of the network by approximating a minimum dominating set of the ad hoc network using only local computation and local state. Each core host maintains the local topology of the hosts in its domain, and also performs route computation on behalf of these hosts.

- Link State Propagation: While it is possible to execute ad-hoc routing algorithms using only local topology information at the core nodes, QoS routing in CEDAR is achieved by propagating the bandwidth availability information of stable links in the core. The basic idea is that the information about stable high-bandwidth links can be made known to nodes far away in the network, while information about dynamic links or low bandwidth links should remain local. The propagation of link-state is achieved through slow-moving "increase" waves (which indicate increase of bandwidth) and fast-moving "decrease" waves (which indicate decrease of bandwidth). The key questions to answer in link state propagation are: When should an increase/decrease wave be initiated? How far should a wave propagate? How fast should a wave propagate?

- Route Computation: Route computation first establishes a core path from the domain of the source to the domain of the destination. This initial phase involves probing on the core and the resultant core path is cached for future use. The core path provides the directionality of the route from the source to the destination. Using this directional information, CEDAR iteratively tries to find a partial route from the source to the domain of the furthest possible node in the core path (which then becomes the source for the next iteration) which can satisfy the requested bandwidth, using only local information. Effectively, the computed route is a concatenation of the shortest-widest paths found locally using the core path as the guideline; otherwise, a failure is reported.

2.4.4.2 Ticket-Based Probing

The authors in [CHE99] developed a generic distributed QoS routing framework based on selective probing. Using this framework, they proposed a distributed multi-path QoS routing scheme for MANET, called ticket-based probing, which is designed to work with imprecise state information. It allows the dynamic trade-off between routing performance and overhead. For ad-hoc networks, they devised a source-initiated routing algorithm and a destination-initiated routing algorithm for the bandwidth-constrained routing problem. These algorithms do not require global state or topology information and use only the local state maintained at every node. Imprecise state information can be tolerated and multiple paths are searched simultaneously to find the most feasible and the best path.

The basic idea of this scheme is that the tickets are utilized to limit the number of the candidate paths during the route discovery. A ticket is the permission to search for a single path. When a source needs a path to reach the destination, it first issues a routing message, called a probe, to the destination. A probe is required

to carry at least one ticket, but it may consist of more. Therefore, the maximum number of the searched paths is bounded by the tickets issued by the source. At the intermediate node, a probe with more than one ticket is allowed to split into multiple tickets. Then, based on its available state information, it decides whether the received probe should split and to which neighbours the probe(s) should be forwarded. Once the destination node receives the probe message, a possible path is found.

In the ticket-based probing scheme, a three-level path redundancy is utilized in case of route failures. The path redundancy scheme establishes multiple routes for the same connection. For the highest level of redundancy, resources are reserved along multiple paths and every packet is routed along each path. In the second level, resources are reserved along multiple paths. However, only one is used as the primary path while the rest serve as backup. In the third level, resources are only reserved on the primary path even multiple paths are selected.

2.4.4.3 QoS Over AODV

The Ad hoc On-demand Distance Vector (AODV) [PER00a] routing protocol has been introduced for best-effort routing in MANET. AODV builds routes using a route request / route reply query cycle. When the source needs to find a path to reach a new destination, it broadcasts a RREQ packet across the network. Nodes receiving this packet update their information for the source node and set up backward-pointers to the source node in the route tables. In addition to the source node's IP address, current sequence number and broadcast ID, the RREQ contains the most recent sequence number for the destination of which the source node is aware. A node receiving the RREQ may send a route reply (RREP) if it is either the destination or if it has a route to the destination with the corresponding sequence number greater than or equal to that contained in the RREQ. If this

is the case, it sends a RREP back to the source. Otherwise, it rebroadcasts the RREQ. Nodes keep track of the RREQ's source IP address and broadcast ID. If they receive a RREQ that they have already processed, they discard the RREQ and do not forward it.

As the RREP propagates back to the source, nodes set up forward pointers to the destination. Once the source node receives the RREP, it may begin to forward data packets to the destination. If the source later receives a RREP containing a greater sequence number or contains the same sequence number with a smaller hop-count, it may update its routing information for that destination and begin using the better route.

To provide QoS support, a minimal set of QoS extensions has been specified for the RREQ and RREP messages [CHE99]. Specifically, a mobile node may specify one of two services: a maximum delay and minimum bandwidth. Before broadcasting RREQ or unicasting RREP, a node must be capable to meet the QoS constraints. When a node detects that the required QoS is no longer available, it must send an ICMP QoS-LOST message back to the source. This protocol, which is called E-AODV, addresses the bandwidth and delay guarantee requirements. However, its reactive nature does not help minimize the service disruptions due to nodal mobility.

2.4.4.4 Bandwidth Protocol

In [PER00b], the authors proposed an algorithm for bandwidth calculation and reservation based on the destination sequenced distance vector (DSDV) routing algorithm for multimedia support in multihop wireless networks. The proposed bandwidth routing scheme depends on the use of a CDMA over TMDA medium access scheme. The transmission scale is organized in frames, each containing a fixed number of time slots, and a global clock or time synchronized mechanism is

used. The path bandwidth between a source and a destination is defined as the number of free time slots between them. Bandwidth calculation needs information about the available bandwidth on each link along the path as well as resolving the scheduling of free slots. Since this problem is an NP-problem, a heuristic approach is needed, as detailed in [LIN99]. The purpose of the proposed QoS routing protocol is to derive a route to satisfy the bandwidth requirement for the QoS constraint. At a source node, the bandwidth information can be used to decide whether to accept a new call or to delay it. In addition, the authors introduced the "standby" routing method to support fast rerouting when a path is broken. When the primary path fails, the secondary route becomes the primary and another secondary is discovered.

2.4.4.5 Trigger-Based Distributed Routing

In [DE02], the authors presented an on-demand and yet proactive routing algorithm called trigger-based distributed routing (TDR) to deal with the link failures due to nodal mobility in MANETs. They try to provide real-time QoS support while keeping the network overhead low. Thus, the flow state for each session is maintained in a distribution fashion at active nodes. In case of imminent link failure in the active route, an alternate route searching overhead is kept limited by localizing the reroute queries within certain neighbors of the node along the source-destination active route. For cost efficiency, rerouting is attempted from the location of an imminent link failure which is referred to as an intermediate node initiated rerouting (INIR). If INIR fails to keep the flow state disruption to a minimum, rerouting is attempted from the source node, in which case it is referred to as source initiated rerouting (SIRR). The TDR scheme keeps the size of the nodal database small by maintaining only the local neighboring information. In addition, it makes use of selective forwarding of routing requests based on GPS information.

From the network point of view, the TDR scheme is an on-demand algorithm, as the rerouting routine is triggered as an active node based on the level and trend of variation of its receive power from the downstream active node. Hence, it is called "trigger-based" routing. On the other hand, from the user application point of view, it is a proactive algorithm as, ideally, the traffic experiences no break in the logical route during the session, thus making it suitable for dealing with real-time traffic. TDR also is "distributed" in the sense that any active node participating in a session can make its own routing decision.

2.5 Summary

In this chapter, we have presented a brief coverage of a wide range of concepts and systems that are related to this research problem. We believe that all of these topics will have some degree of influence on the outcome of the research that will be presented in this dissertation. We have provided a classification of the MANETs routing protocols according to the routing strategy, i.e., table-driven, on-demand, cluster-based, and constrains routing protocols. We have presented descriptions and comparisons of several routing schemes, highlighting their characteristics, features, and drawbacks. Also, we have reviewed the current research on QoS support in MANETs. This includes QoS models, resource reservations signaling, QoS routing protocols, and some other related works. Additional details on any of theses topics is available in the references listed in the endnotes of this thesis.

Chapter 3

A FRAMEWORK FOR PROVIDING RELIABLE QOS IN MANETS

3.1 Introduction

In the mobile ad-hoc network (MANET), two nodes can communicate if the distance between them is less than the minimum of their two broadcast ranges. Each node can function as a host, when it is a source or a destination, or a router, when it is an intermediate node between the source and the destination providing store-and-forward services to neighboring nodes. Due to the limited transmission range of wireless interfaces, multiple hops may be needed to exchange data between nodes in the network, so sometimes the term multi-hop is used to describe the ad-hoc network and the multi-hop routing mechanism is used to maintain the network connectivity. Those networks provide mobile users with ubiquitous communication capability and information access regardless of location. However, in the MANETs, not only the wireless link characteristics cause the disruption of the QoS, the host mobility and the battery power consumption may do so as well. The main problem that faces MANETs is the lack of stability in the network topology and reliability of the routes that carry the data between the source-destination pair. Therefore, supporting the stability and the reliability of the data route is the aim of the SERC/LC3R framework. This will help extend the QoS techniques to the mobile ad hoc networks and support its requirements.

The SERC/LC3R framework has three main goals that work toward achieving network stability and reliability. These goals fall into three main areas: clustering, routing, and QoS. Obviously, each of these areas will affect the other areas. Reforming the cluster quickly will improve the routing process, which will increase the route lifetime and provide more QoS support, which in turn will make the network more reliable which is the main aim of this work. In this chapter, we describe the SERC infrastructure protocol and the LC3R protocol in details with some illustrated examples.

3.2 The SERC Protocol Description

The SERC protocol aims to achieve the stability and reliability in the cluster-based MANETs. These objectives can be achieved by utilizing two clusterheads for each cluster. The basic idea here is the election of a primary clusterhead (PCH) and a secondary clusterhead (SCH) for each cluster and both of them can gather the reservation signaling messages and the data packets that are sent by the downstream nodes. The SCH is elected and assigned by its PCH which deals with it at the same time as a member node. Also, the SCH stores the forwarding packets based on an expiry timer. Based on its battery power level, if the PCH can no longer be a clusterhead, it will inform its SCH to be the PCH and it will change its status to be a regular member node. On the other hand, by using the location/mobility prediction [SU01], the PCH can be predicted if it is about to move out of the transmission range of its members. Thus, the SCH could be triggered to be involved as a PCH before the cluster actually collapses. Since the new CH is already assigned and is just triggered to be involved in the route, the clusterhead computation and the communication overhead, due to the frequent information exchange among the participating nodes, will be reduced. Moreover, the cluster residence time for the member node will be extended because the node

is associated with the cluster not to the clusterhead. Therefore, the SCH works as a backup for the PCH and the route going through it works as a backup route for the main one that is going through the PCH. Hence, a smooth clusterhead transfer from the PCH to the SCH will give the cluster more lifetime and stability and give the route that is going through it more lifetime and reliability. The flowchart shown in Figure 3.1 gives the whole picture of the SERC protocol.

The WEAC infrastructure creation protocol and the virtual base station on-demand (VBS-O) protocol [SHE03] are taken as a background for this approach. These protocols have been chosen because of their positive features regarding load balancing and energy saving. To accommodate our approach by WEAC, several modifications are made to adopt and take advantage of the secondary clusterhead, which is the main contribution of this work. Although we modified the WEAC protocol to accommodate our protocol, we believe that it can be applied with any other clustering algorithm.

3.2.1 WEAC Protocol Qualifications

We have chosen the WEAC protocol for our SERC protocol modifications, comparisons and investigations. This is because in a series of comparative studies [AMI00, HAS01, SHE02b, SHE03, SAF01b] with different clustering algorithms and protocols such as Min-Max [AMI00], CGSR [ROY99a, CHI97], VBS [HAS01], PA-VBS [SAF01b] in different aspects such as stability and load balancing, the WEAC protocol has proved to be a good candidate for this study. The simulation results in [SHE03] showed that VBS-O on top of the WEAC protocol outperforms both PA-VBS and VBS in load balancing, power saving, packet delivery time, duration of clusterheads, and the life time of the network. Also, the results in [SHE02b, SAF01b] showed that PA-VBS protocol outperforms both VBS and CGSR in power saving and packet delivery time. Moreover, in [HAS01], VBS

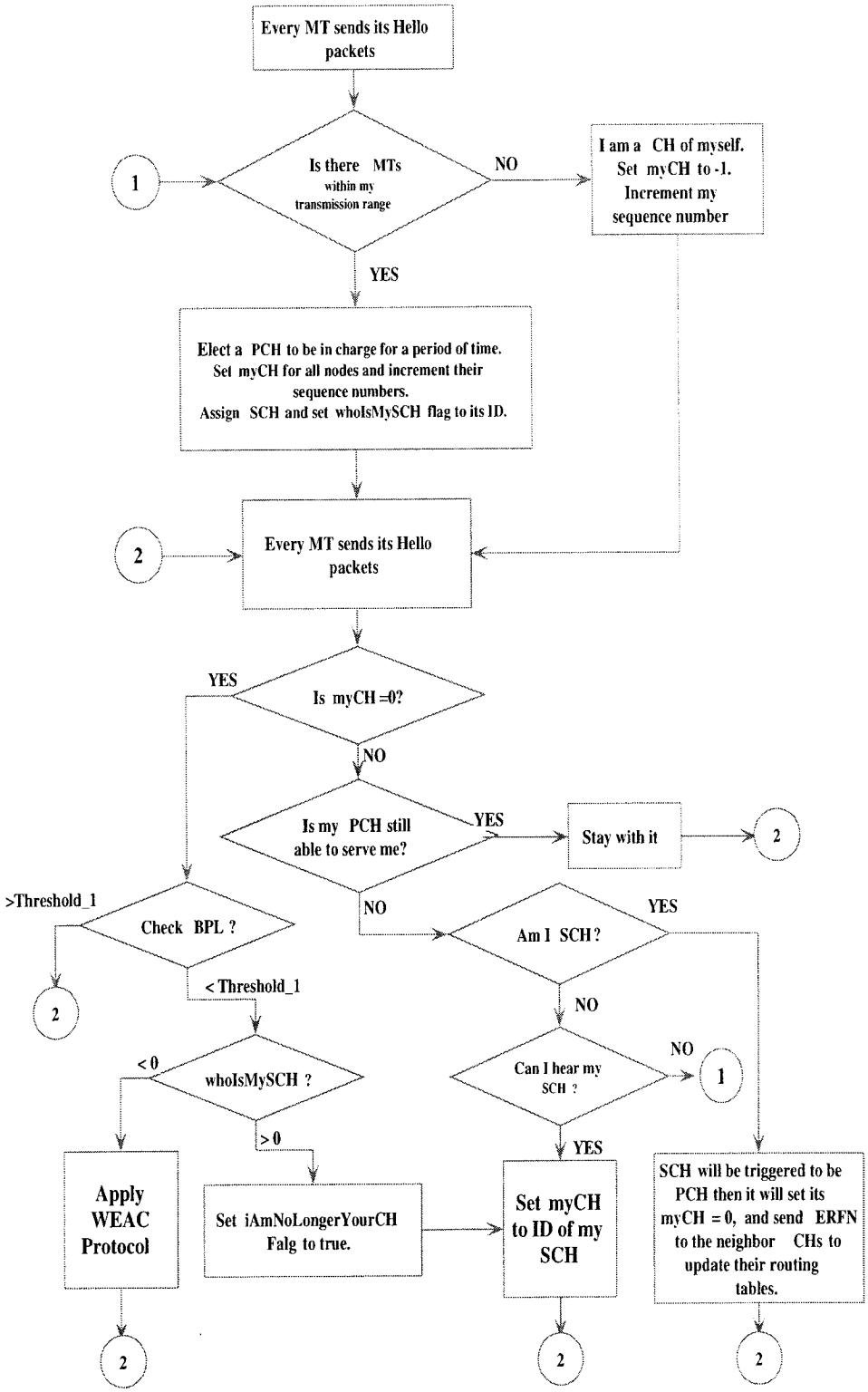


Figure 3.1: The SERC Protocol Flowchart

surpasses max-min in terms of the average VBS duration, and the zone MT's membership duration. This indicates that VBS is more stable than the Max-Min algorithm. In [AMI00], the simulation experiments demonstrate that the Max-Min heuristic is better than the two earlier heuristics, namely the LCA and LCA2 solutions. Consequently, since Max-Min outperforms LCA and LCA2, VBS outperforms Max-Min, PA-VBS outperforms VBS and CGSR, and WEAC outperforms VBS and PA-VBS, then the WEAC protocol outperforms all the LCA, LCA2, Max-Min, CGSR, VBS, and PS-VBS protocols. Let us now summarize these comparative studies.

Max-Min compared to LCA and LCA2: Compared to the LCA heuristic, the Max-Min heuristic produces fewer clusterheads, much larger clusters, and longer clusterhead duration on the average. While the degree based heuristic does have slightly larger cluster sizes than the Max-Min, it suffers greatly in other categories such as clusterhead duration, and cluster member duration. The LCA2 heuristic produces clusterheads that are comparable in number to those of Max-Min. However, Max-Min has clusterhead durations that are approximately 100% larger than that of LCA2 for dense networks. Furthermore, the Max-Min clusterhead duration continues to increase with increased network density, while the LCA2 heuristic clusterhead duration decreases with increased network density. Based on the simulation results in [AMI00], the Max-Min heuristic provides the best all around clusterhead leader election characteristics as compared to LCA and LCA2.

VBS Compared to Max-Min: The average clusterhead duration is an important measure of the stability of any clusterhead-formation protocol. The results in [HAS01] clearly show that VBS outperforms max-min in terms of stability. The clusterhead duration in the case of Max-Min is less than VBS. This is a direct result of the fact that Max-Min forms d-hop clusters. Since the set of all mobile stations that are within d hops from a clusterhead is more than or equal to the

number of mobile stations that are a single-hop away, it is more likely that a d-hop cluster-formation protocol, such as Max-Min, will undergo disorder, unlike a single-hop infrastructure-creation protocol, such as VBS. Therefore, the larger the number of MTs a clusterhead is in charge of, the more often the protocol must be rerun to maintain a valid infrastructure. This implies that using VBS makes it less likely to interrupt ongoing communications due to VBS handovers. Moreover, the results in [HAS01] show that the zone MTs' membership duration in the case of VBS is much larger than Max-Min. This is due to the fact that the average clusterhead duration for Max-Min is less than the average clusterhead duration for VBS. Since Max-Min suffers from frequent handovers, the cluster members will switch from one clusterhead to another more often, thus resulting in a small zone-MT membership duration. As a result, VBS provides more stability than Max-Min.

PA-VBS Compared to VBS and CGSR: PA-VBS surpasses VBS in balancing the load amongst the nodes of the wireless mobile ad hoc network. Unlike other clustering protocols, PA-VBS allows the mobile nodes to use their valuable battery power fairly. Contrary to VBS, PA-VBS does not drain all the battery capacity of the clusterheads since clusterheads do not accept anymore merge requests when the energy left is below a certain threshold. Moreover, PA-VBS introduces the concept of service denial. In [SAF01b], the simulation results showed that PA-VBS always elects 100% of the wireless nodes as clusterheads; hence, it attains fair clustering. All the nodes will serve as clusterheads, at least once, during their lifetime. Most importantly, PA-VBS keeps the total consumed energy by the MTs almost constant. This guarantees fair energy consumption amongst the wireless mobile nodes. PA-VBS balances the load amongst the wireless nodes regardless of the carried routing load. Consequently, PA-VBS can be utilized as a basis for routing in ad hoc networks. Moreover, PA-VBS can form the cornerstone for the wise distribution of the network load amongst all the viable paths between a source

and destination pair. A comparative study of the PA-VBS and CGSR in [SHE02b] shows that PA-VBS outperforms CGSR in terms of stability. This is because the PA-VBS allows clusterheads to be within the transmission range of each other; however, CGSR does not allow that. This point led to a significant difference in the average lifetime of both the clusterheads and the terminals.

WEAC Compared to VBS and PA-VBS: The simulation results in [SHE03] show that WEAC and PA-VBS protocols produce almost the same number of clusterheads. The VBS protocol produces the smallest number of clusterheads. This is because WEAC and PA-VBS distribute the load amongst the MTs under certain power level constraints; however, the VBS protocol, which uses the smallest ID algorithm, elects the nodes with small IDs all the time. In this study, the average duration of clusterheads of the VBS protocol has the longer duration. However, this implies that clusterheads consume more power than other MTs and their batteries drain faster than other MTs. Therefore, the VBS protocol is most likely to experience a reduction in connectivity. In the case of WEAC, the average duration of clusterhead is more than that of PA-VBS, and the network in the case of WEAC is more stable than PA-VBS. However, as mentioned above, we should look for the next metric before we jump to any conclusion. The number of election in the case of the VBS protocol is much higher than that of WEAC and PA-VBS. Therefore, even though the duration is much longer, the system is not stable because re-elections are taking place very often. WEAC has a smaller number of elections than PA-VBS. Hence, we can conclude that WEAC is more stable than VBS and PA-VBS protocols. WEAC and PA-VBS attain load balancing between MTs in the network. Every MT is elected as a clusterhead in the simulation experiments. One the other hand, the VBS protocol elects a small number of MTs (smallest ID) as clusterheads, which means that the VBS protocol does not assure fairness between MTs as with the case of WEAC and PA-VBS. The study

in [SHE03] shows that WEAC is the fairest scheme in balancing the load amongst MTs. The VBS protocol has the worst performance in this metric. Also, this study shows that VBS-O/WEAC outperforms the PA-VBS and the VBS protocols in successfully transmitted packets and in packet delivery time. This is because WEAC guarantees more robust routes than the case of both PA-VBS and VBS.

3.2.2 Clustering Process

The concept of dividing the geographical region into small zones has been presented as clustering in the literature. The idea of clustering in ad hoc networks is not new. Clustering basically transforms a physical network into a virtual network of interconnected clusters or groups of nodes. Those clusters are dominated by clusterheads and are connected by gateways or border mobile terminals. Any node can be a clusterhead if it has the necessary functionality, such as processing and transmission power. The nodes registered with the nearest clusterhead become members of that cluster. Clustering in MANET has been used in order to facilitate management, to improve routing efficiency, to support QoS and to save power consumption. The main goal is to find the clusterhead nodes and partition the network into clusters. These nodes take on a special role in managing routing information. Therefore, stability starts from partitioning the network into clusters. Once the clusters are established, the route stability between a source-destination pair is based on the clusterheads involved in that route. However, due to the dynamic nature of the mobile nodes, their association and dissociation to and from clusters perturb the stability of the network and the problem becomes worse if these nodes are clusterheads. The frequent changes of the clusterheads affects the performance of other protocols such as scheduling, routing and signaling that rely on it. Also, re-computations of clusterheads and cluster memberships, and frequent information exchange among the participating nodes will cause more

overhead. Cluster maintenance overhead has been seen as a serious disadvantage for the most clustering protocols. Therefore, in the SERC protocol, since the clusterhead tries to identify and assign one node as the future leader of the cluster and it is known by the cluster members, the communication overhead to reform the cluster will be reduced. The leadership of the cluster will be handled smoothly from node to another with no need to invoke the re-clustering messages (*merge request/accept/disjoint* messages). To accommodate our approach by WEAC, we add to the *Hello* message a new flag called *WhoIsMySCH* flag. This flag is used to inform the SCH about its new responsibility and to inform the cluster members about their future leader. The details of this scheme in the clustering phase are as follows:

- The PCH is elected based on the WEAC protocol.
- The PCH assigns periodically its SCH based on the distance-power algorithm shown in Algorithm 6 (see Section 3.5.2), and then sets its *WhoIsMySCH* flag to the *ID* of the SCH.
- Setting of the *warning* flag or the *iAmNoLongerYourCH* flag by the PCH can be used by the SCH once it receives its *Hello* message to trigger itself to be the PCH and to assign a new SCH.
- By using location/mobility prediction [SU01], if the PCH can no longer be a clusterhead for its members, the SCH will be triggered to be the PCH and a new SCH will be assigned.
- If the *Hello* packets of the PCH are lost at the SCH, the SCH will check its topology information (GPS is used) and then if it is not moved away, it will trigger itself to be the PCH and will assign a new SCH. However, if it is moved away, it will change its status to be a free node.

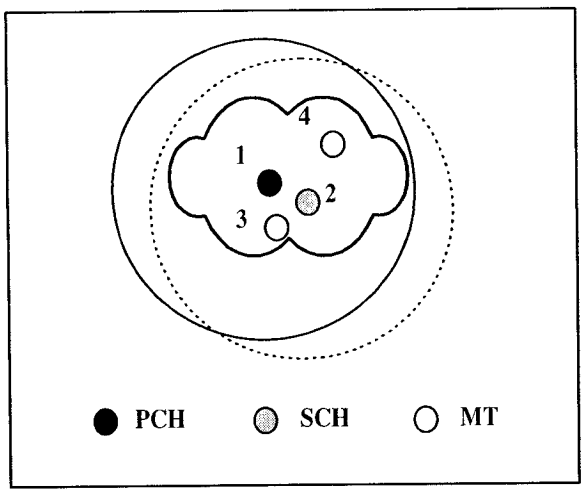


Figure 3.2: PCH & SCH First Election

- If the *Hello* packets of the SCH are lost at the PCH, based on its topology information, PCH will make its decision. If PCH is not moved away, it will assign a new SCH. However, if it is moved away, it will change its status to be a free node.

- Once the PCH can no longer be a CH, and the cluster member node has realized that, the node, based on *WhoIsMySCH* flag, will set its myCH value to the *ID* of the SCH if its neighbors list contains it (if they are in the transmission range of each other); if not, it will set its status to be a free node.

3.2.3 Illustrated Example

In this section, an illustrated example will be provided to explain how our protocol works. In Figure 3.2 MT 1 is elected in the beginning to be the PCH by using WEAC protocol. MTs 2, 3, and 4 are its members. MT 2 is assigned to be the SCH based on power and location qualifications. Now, the cloud of MTs 1, 2, 3, and 4 forms the first cluster. In Figure 3.3, when MT 1 can no longer be a clusterhead, MT 2 which is the SCH will be triggered to be the PCH. Therefore,

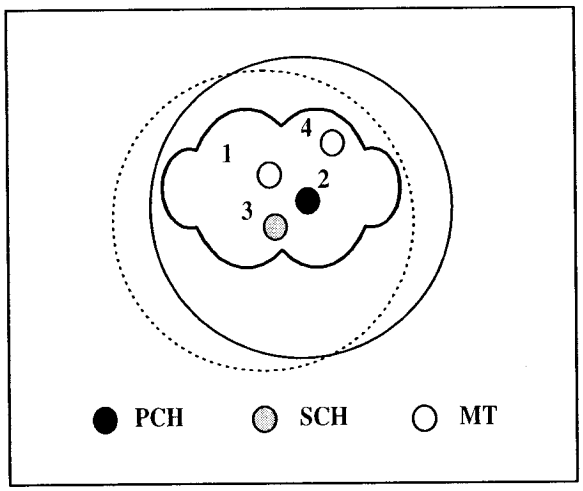


Figure 3.3: First Successful Cluster Leadership Transfer

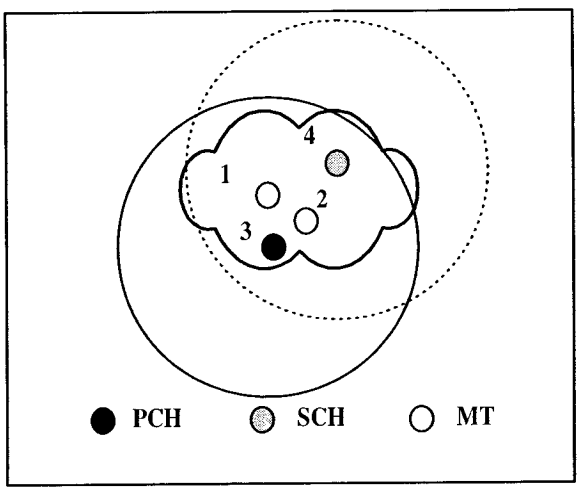


Figure 3.4: Second Successful Cluster Leadership Transfer

the cluster that was formed in Figure 3.2 is still surviving and its clusterhead is the only change that has been done. MT 2 in its turn will assign its SCH which will be MT 3. The cluster leadership will be transferred from MT 2 when it can no longer be a CH to MT 3 (Figure 3.4) and MT 4 will then be assigned to be the SCH. These processes will be repeated until there is no qualified node to be SCH. Since the cluster is still surviving and its head is smoothly transferred, the cluster looks stable in front of the other clusterheads. Also, cluster residence time for all MTs 1, 2, 3, and 4 will be longer, which is desired. The overhead for replacing of the PCH by the SCH is very low as compared with the clustering reformation overhead.

3.3 The LC3R Protocol Description

The LC3R protocol aims to achieve the routing reliability in the cluster-based MANETs. This objective can be achieved by utilizing the SERC protocol described above. The basic idea of SERC is the election of a primary clusterhead (PCH) and a secondary clusterhead (SCH) for each cluster. The SCH will be elected and assigned by its PCH which deals with it at the same time as a member node. In LC3R, the SCH as well as its PCH could gather the reservation signaling messages and the data packets that are sent by the downstream nodes. The SCH will store the forwarding packets based on an expiry timer. Based on its battery power level, if the PCH can no longer be a clusterhead, it will inform its SCH to be the PCH and it will change its status to be a regular member node. In a mobile environment, by using location/mobility prediction [SU01], the PCH can be predicted if it is about to move out of the transmission range of its members. Thus, the SCH could be triggered to be involved as a PCH before the cluster actually collapses. Moreover, the non-expired packets at the SCH will be forwarded by the SCH once it is triggered as a PCH. Therefore, the SCH works as a backup for the PCH and

the route going through it works as a backup route for the main one that is going through the PCH. Hence, a smooth clusterhead transfer from the PCH to the SCH will give the cluster more lifetime and stability, and give the route that is going through it more lifetime and reliability. To the best of our knowledge, this is the first work that proposes virtual multipath routing and locally rerouting in presence of two clusterheads for each cluster.

3.3.1 Routing Process

Maintaining and minimizing the route interruptions are two ways of keeping the stability of the network. We argue that the lifetime of a route in a MANET is crucial because of the delay involved in any route discovery protocol. We claim that the lifetime of a particular route is dependent on the movements and the battery power level of all the nodes involved in the route. By maintaining the route lifetime, we most likely keep the network in stable condition as long as possible. The localized cluster-based re-routing is designed to increase the route lifetime. Also, it is designed to save routing time and reduce routing overhead that is caused by failure of the PCH in the route recovery process or in the routing update tables. Moreover, since the new PCH has the knowledge of the route and has the recently sent packets when it was SCH, the clusterhead changing will not cause service interruption. The new SCH will gather the route information and the sent packets, and then play the old SCH role to keep the clusterhead re-election running smoothly. Notice that the power of the SCH will not be consumed to forward any packets; it just receives and stores them temporarily until it is triggered to be PCH and then forwards them. The routing process details of the LC3R protocol are as follows:

- The PCH is elected and the SCH is assigned periodically for each cluster by using the SERC protocol.

- Based on its power level or by using location/mobility prediction [SU01], if the PCH can no longer be a clusterhead for its members, the SCH will be triggered to be the PCH and a new SCH will be assigned.

- The new PCH sends an explicit routing failure notification (ERFN) to each downstream clusterhead of each current flow that it has.

- Once the ERFN has been received by the downstream PCHs and SCHs, they will update their routing tables.

3.3.2 QoS Support

Packets may be dropped in the middle, if the lifetime of a route between nodes is shorter than the time required to transmit them as a result of its failure. The unreliable end-to-end connections make it so difficult to provide QoS requirements. The route recovery/maintenance process in most routing protocols is suggested to ensure the successful packets transmission after a route interruption. However, the routes need to be recovered in a timely and efficient manner such that the packet delays and losses are minimized. In our protocol, because the dropped packets at the PCH are stored at the SCH and they will be sent immediately once the SCH involved in the route as a PCH, the packet delay/jitter and packet loss ratio will be improved. Therefore, the protocol increases the level of reliability of packet transmissions as requested by the connection. In order to limit the impact of the mechanism on the battery power consumption, the SCH is not involved in any packet transmission. It just receives the transmitted packets to its primary clusterhead. Since the mobile node transceiver is constantly listening to the channel for the detection of incoming packets, power consumed in the listen state is close to the receive power. Therefore, the power consumption is very low compared with the other multipath protocols.

Besides improving the packet delay/jitter and the packet loss ratio, the presence of the QoS or the resource allocation parameters at the SCH will help to provide more reliability to support the QoS requirements. The details of LC3R protocol to support the route QoS requirements are as follows:

- For each connection, the PCH's chain works as the main route, while the SCH's chain works as the back-up route.
- The resources should be reserved at the PCH as well as at the SCH.
- The received packets at the PCH should be received as well at the SCH through the downstream node.
- The PCH is the only one that forwards the packets, and the packets at the SCH are just stored temporarily to be forwarded when it is triggered to be a PCH.
- Those packets will be dropped based on timeout expiration if there is no change to the SCH status.
- RSVP refresh messages, if they are used, can do the reservation at the new SCH.

By doing this, the resources can be reserved locally within the affected cluster, and no packet losses during the route failure because of the ongoing sent packets are buffered at the SCH.

3.3.3 Illustrated Example

In this section, an illustrated example will be provided to explain how the LC3R protocol works. When the sender has data packet to send to the destination, the main route can be established through the PCHs chain (Fig. 3.5), while the

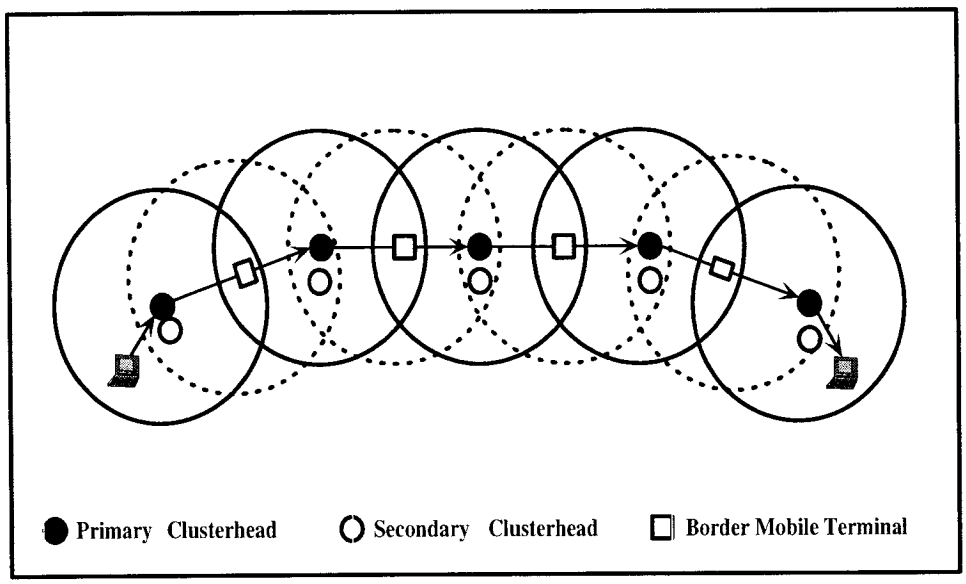


Figure 3.5: Primary Route is established from the sender to the destination.

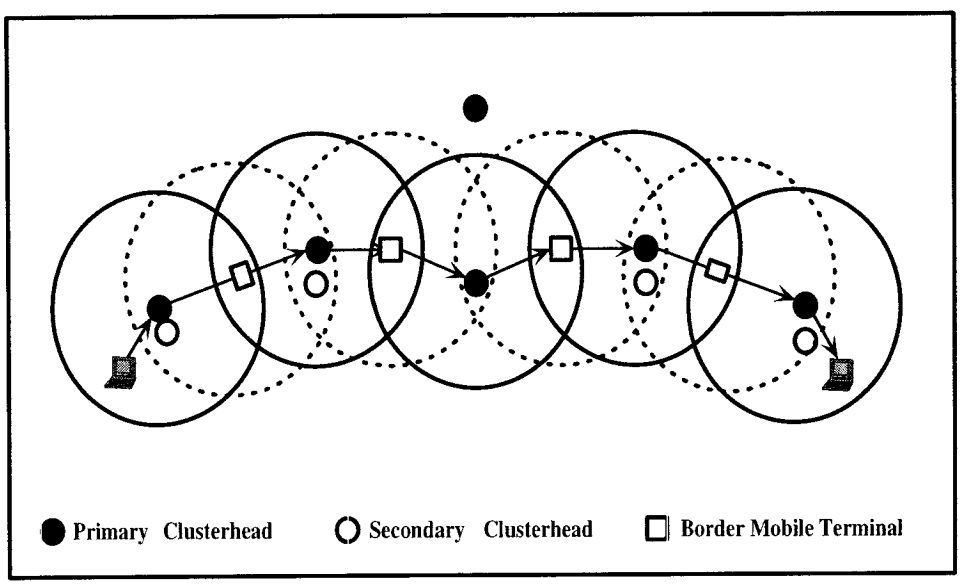


Figure 3.6: The PCH is leaving its cluster and SCH is being triggered to be a new PCH.

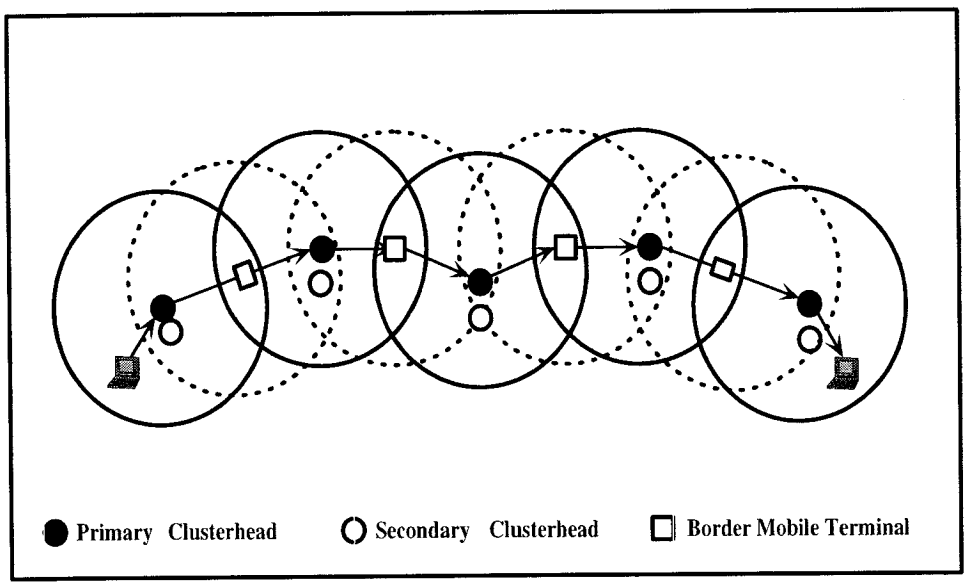


Figure 3.7: New Secondary is electing

back-up route can be established through the SCHs chain. Once the PCH moves away or can no longer be a clusterhead due to its power consumption, the SCH can be triggered to be the PCH (Fig. 3.6) and a new SCH would be assigned (Fig. 3.7). Now, the new PCH has the QoS information, which can be used immediately without re-reservation. Furthermore, because the ongoing sent data packets are stored at the new PCH, they will be forwarded immediately to their next hop destinations. By using this scheme, the route can be maintained locally within the affected cluster, the resources have been reserved locally as well, and no packets are lost during the route failure. Moreover, since the new PCH was already assigned and it is just triggered to be involved in the route, the clusterhead computation and the communication overhead, due to the frequent information exchange among the participating nodes, will be reduced. Also, no transmissions are needed over the secondary route because the SCHs can hear the transmissions over the primary routes and then gather and store, temporarily based on a timeout timer, the ongoing sent packets that are being forwarded by the downstream nodes. When the SCH is triggered to be the PCH, the non-expired stored packets will be

forwarded to avoid packet losses. Therefore, the bandwidth of the network is not wasted by the redundant transmissions.

3.4 SERC/LC3R Framework Overview

As in all cluster-based protocols, in our framework, some of the MTs, based on their current battery power level, are elected to be in charge of all the MTs within their transmission range, or a subset of them. This can be achieved by electing one to be the primary clusterhead (PCH) and another to be the secondary clusterhead (SCH). Every MT acknowledges its location and its battery power level (*BPL*) via what is called *Hello* packets, sometimes called *beacon* packets, for a period of time. MTs, in our protocol, are classified as follows:

- Primary Clusterhead (PCH): is the leader of the cluster.
- Secondary Clusterhead (SCH): is the future leader of the cluster.
- ZoneMT: is an MT supervised by a PCH.
- FreeMT: is an MT that is neither a CH nor zoneMT, (i.e., it is not associated with a cluster).
- Border Mobile Terminal (BMT): is an MT that lies between more than one CH or FreeMT; it can be a CH or a zoneMT or a FreeMT.

Every MT has a *myCH* variable which is used to store the *ID* number of the PCH in charge of that MT. An MT sets its *myCH* variable to the *ID* number of its PCH; however, if that MT itself is a PCH, then the *myCH* variable will be set to 0, otherwise it will be set to -1, indicating that it is a PCH of itself or a free node. Once the PCH has been elected in such cluster, it will assign its SCH based on *distance-power-level algorithm*, which will be described later. If there is

an eligible ZoneMT that can be a SCH, the PCH will set its *whoIsMySCH* flag to the *ID* of this MT. The value of this flag is *NULL* if this PCH has no eligible SCH. This flag is used to inform the SCH about its new responsibility and to inform the ZoneMTs of their future leader. Therefore, based on their PCH *whoIsMySCH* flag, all the ZoneMTs will set their flags as well. A PCH collects complete information about all other PCHs and their lists of MTs, and broadcasts this information in its periodic hello messages. This information can be collected also by the SCH, and it can be used when the SCH is triggered to be a PCH. When the SCH is triggered to be a PCH, the ZoneMTs of the former PCH will switch to their new PCH and set their myCH variable to their *whoIsMySCH* variable. To announce that the cluster leadership has been transferred successfully and the cluster has only one PCH, the new PCH will set its *iAmYourPCH* flag to true. Also, the new PCH in its turn will assign a new SCH and send an explicit routing failure notification (ERFN) message to the neighbor PCHs to update their routing tables.

When the ZoneMTs find the *iAmYourPCH* flag to be true, they will start sending their packets to their new PCH. ZoneMTs accumulate information about the network from their neighbors between *Hello* messages, and they broadcast their neighbor list to their neighbors in their *Hello* packets. Each MT announces its *ID* number, location and *BPL* with its periodic *hello* message. The method of electing the PCH from a set of nominees is based on its *BPL*. Therefore, an MT sends a *merge-request* message to another MT, if the latter has a higher BPL and it should be more than or equal to $THRESHOLD_1$. The receiver of the *merge-request* message responds with *accept-merge* message, and sets its myCH variable to zero. When an MT receives the *accept-merge* message, it sets its *myCH* variable to the *ID* number of its PCH. The *BPL* of each and every MT is categorized under one of the following four categories (Fig. 3.8):

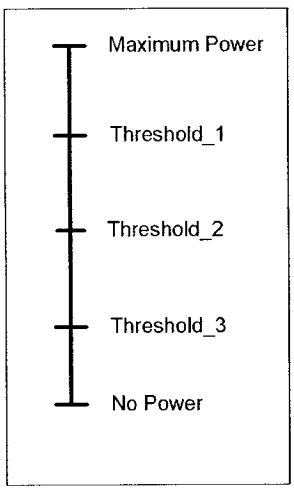


Figure 3.8: Four power levels of each MT

- MT $BPL \geq THRESHOLD_1$: An MT is eligible to be a PCH and willing to accept other MTs to be under its supervision if these MTs have a lower *BPL*. If the MTs with the same *BPL*, which it is almost impossible, then the one with more number of neighbors wins. Then, the PCH will assign its SCH if it has a qualified node and will set its *whoIsMySCH* variable.

- $THRESHOLD_1 >$ MT $BPL \geq THRESHOLD_2$: An MT will ignore any *merge request* messages that are sent to it by other MTs. If the MT is serving as a clusterhead, it will remain a clusterhead, but it will add no more nodes under its supervision.

- $THRESHOLD_2 >$ MT $BPL \geq THRESHOLD_3$: If an MT is serving as a PCH (its $myCH$=0), and it has a qualified SCH, it will set *iAmNoLongerYourCH* flag to true to trigger the SCH to be the new PCH. Then, setting this flag to true will inform all the MTs that were under the former PCH supervision to set their *myCH* variable to its *whoIsMySCH* variable with no need to invoke the clustering messages. However, if it does not have a qualified SCH and the *whoIsMySCH* variable is *NULL*, as in the WEAC protocol, it will set

its *warningThreshold* flag to true informing its ZoneMT to look for another PCH, nevertheless, they can remain with it until its *BPL* drains down to $THRESHOLD_3$.

- MT $BPL < THRESHOLD_3$: An MT will ignore any *merge request* messages and will set *iAmNoLongerYourCH* flag to true to inform all the nodes under its supervision, if it was serving other nodes, that it can no longer be a PCH. This flag will trigger the SCH, if there is one, to be a PCH. Then, as in the last point, setting this flag to true will inform all the MTs that were under the former PCH supervision to set their *myCH* variable to its *whoIsMySCH* variable. However, if there is no SCH and the *whoIsMySCH* variable is *NULL*, the regular WEAC protocol will be invoked to build a new cluster.

In all the above cases, if $myCH > 0$, no *merge request* will be sent by MTs. However, if $myCH = -1$, then it will send a *merge request* to an MT whose *BPL* is greater than or equal to $THRESHOLD_1$.

3.5 Detailed Description of the SERC/LC3R Framework

This section contains the pseudo code for the algorithms executed by the MTs running the SERC/LC3R framework. The *hello* routine, see Algorithm 1, is called by an MT periodically when it needs to send its *hello* message. Each MT broadcasts its *ID*, sequence number, *myCH* variable , neighbors list, *BPL* and location (GPS information is used) as part of its *hello* message. The *hello* message contains other useful pieces of information, such as the Border Mobile Terminal (*BMT*) flag, *whoIsMySCH* flag, *iAmYourPCH* flag, *warningThreshold* flag, and the *iAmNoLongerYourCH* flag. If the *BMT* flag is set to true, then it means that this node is a gateway, which is the only node eligible to acquire full knowledge of the network. Other flags will be shown in details in this section.

Algorithm 1: Hello Routine

```
1 Send: my ID,
        my Sequence Number,
        myCH,
        my Location X and Y,
        my List of Neighbors,
        my BPL,
        my BMT flag,
        my whoIsMySCH flag,
        my iAmYourPCH flag,
        my warningThreshold flag,
        my iAmNoLongerYourCH flag
2 if myCH > 0 then
3     DeleteAllListsEcxeptMyCH'sLists;
4 end
```

3.5.1 Merge Request, Accept and Disjoint Algorithms

After receiving a *hello* message, an MT sends a *merge request* message to the issuer of the *hello* message, if at least one of the following two conditions is satisfied:

i. The MT is a FreeMT, with the following constraints:

   (a) Its *BPL* is less than the *BPL* of the issuer of the *hello* message.

   (b) The *BPL* of the sender is above $THRESHOLD_1$.

ii. The MT is under the supervision of PCH and its *whoIsMySCH* variable is *NULL*, with the following constraints:

   (a) The *warningthreshold* of its PCH is true.

   (b) Its *BPL* is less than the *BPL* of the *hello* message issuer.

   (c) The *BPL* of the sender is above $THRESHOLD_1$.

Algorithm 2: Merge Request Decision Process

```
1 if whoIsMySCH is NULL then
2     if ((recieved BPL ≥ Threshold_1)
3     && (recieved BPL ≥ myBPL))
4     && ((myCH = -1)
5     ||((myCH ≥ 0)
6     && (my warningThreshold = true))) then
7         SendMergRequestToHelloSender;
8     end
9 end
```

Algorithm 2 shows how the decision is made by an MT to determine whether to send a *merge request* to the issuer of the *hello* message, after receiving a *hello* message.

Upon receiving a *merge request* message, an MT starts the *merge accept* process (Algorithm 3) if it is not supervised by a PCH and its *BPL* is above $THRESHOLD_1$. If these conditions are satisfied, the receiver of the *merge request* sets its *myCH* variable to 0 if it is not yet a PCH, increments its list of MTs by 1, starts a timer for the sender (in order to activate the initiation of a timeout period), and sends a *merge accept* message to the sender. Sometimes the *BPL* of the receiver of the *merge request*, upon receiving it, is below $THRESHOLD_1$, or it might go out of the communication range of the sender before it receives the *merge request*. From the sender side, it assumes that its *merge request* was rejected if it does not receive a feedback within a certain time.

As soon as an MT receives a *merge accept* message, it executes the steps shown in Algorithm 4 if and only if it is neither PCH nor an MT served by a PCH. It sets *myCH* variable to the ID of the sender of the *merge accept* message, and starts a timer for its new clusterhead.

The *disjoin* message is sent by an MT when its current PCH sets its *warningThreshold* flag to true, i.e., the current clusterhead *BPL* is below $THERSHOLD_2$. Algorithm 5 shows the series of actions taken by a PCH after receiving

Algorithm 3: The Merge Request Receipt Routine

```
1 if my BPL > Threshold_1 then
2     if myCH ≤ 0 then
3         myCH = 0;
4         ListOfMyMTs++;
5         StartMTTimer();
6         SendMergeAccept(MergeRequestSender);
7     end
8 end
```

Algorithm 4: The Merge Accept Receipt Routine

```
1 if ( myCH = -1 || (myCH > 0 & my warningThreshold = true)) then
2     if myCH > 0 then
3         CancelMyCHTimer();
4         SendDisjointMessage(myCH);
5     end
6     myCH = MergeAcceptSenderID;
7     StartMyCHTimer();
8 end
```

a *disjoint* message. First, it decrements its list of MTs by one, then cancels the timer of the sender of the *disjoint* message. Once the list of MTs becomes empty, the PCH sets its *myCH* to -1.

Algorithm 5: The Disjoint Receipt Routine

```
1 if myCH = 0 then
2     ListOfMyMTs- -;
3     ResetMTTimer(DisjointSenderID);
4     if ListOfMyMTs = 0 then
5         myCH = -1 ;
6     end
7 end
```

All *merge-request*, *merge-accept*, and *disjoint* messages are only invoked in the beginning to build the cluster or when there is no qualified SCH. Therefore, the presence of the SCH will reduce the clustering communication overhead in addition to its advantages in routing and QoS support.

3.5.2 SCH Selection Algorithm

Once the PCH has been elected in such cluster, the SCH will be assigned by its PCH. Having been assigned as SCH, a node should be within accepted range from the PCH and should have the highest *BPL* of the eligible nodes. The SCH should be near its PCH to have the same range to its cluster members and border nodes. The SCH is selected by using the *distance-power-level-based algorithm* (Algorithm 6) which is taking advantage of GPS. This algorithm uses two parameters; the node distance D from the PCH and its battery power level *PBL*. Therefore, each node, within the distance D from the PCH and its power level is *BPL*, is assigned a pair $DBPL = (D,BPL)$. The *BPL* of the eligible SCH should be equal to or greater than $THRESHOLD_1$. Let $NODE_1$ has $DBPL_1 = (D_1,BPL_1)$ and $NODE_2$ has $DBPL_2=(D_2,BPL_2)$. Then $DBPL_1 > DBPL_2$, if $D_1 < D_2$ and BPL_1 & $BPL_2 > THRESHOLD_1$. That is, $NODE_1$ has priority over other eligible SCHs if it is the nearest zoneMT to the PCH. This algorithm is running by the PCH and is used to set its *whoIsMySCH* flag to the ID of the eligible MT to be a SCH. Missing of the hello message from the assigned SCH, will lead to re-run this algorithm again. When the targeted MT acknowledges to be SCH, it will prepare itself to be SCH.

In the case of using the power saving by the SCH and switching back and forth to and from the sleeping mode, the SCH may stay in its power saving mode until at least one of the two following conditions is satisfied:

- The BPL of the PCH becomes less than $THRESHOLD_1$.
- The distance between the SCH and its PCH becomes greater than half of the transmission range if the SCH is moving.

Algorithm 6: Distance-Power Based Algorithm

```
1  if (myCH = 0) then
2      for (i = 0 ; i < allMyClusterMembers; i++) do
3          if (BPL_i ≥ Threshold_1) then
4              EligibleSCHList++;
5              Add to The Eligible List;
6          end
7      end
8      D = Tx;
9      for (i = 0 ; i < EligibleSCHList; i++) do
10         if (D_i < D) then
11             D = D_i;
12             whoIsMySCH = i;
13         end
14     end
15     Clear The Eligible List;
16 end
```

3.5.3 The whoIsMySCH and the iAmYourPCH Flags

The *whoIsMySCH* flag is used by the PCH to inform the assigned SCH that it has been elected to be SCH based on the above *distance-power-level algorithm* (Algorithm 6). Also, it is used to inform the zoneMTs about its SCH as a future leader in case of the PCH absence. At receiving a *hello* message, an MT matches its *ID* with the *whoIsMySCH* variable (Algorithm 7). If it is the assigned one, it will prepare itself to be the SCH. This flag gives an authorization to the SCH to receive and store any message addressed to its PCH. Therefore, once the assigned SCH has been acknowledged, it will get the ability to gather any message sending to its PCH to be stored until the proper time. Also, it will collect the routing information from its PCH. Each stored packet will be given a timeout timer. When this timer expired before the SCH triggering, it will be deleted. However, once the SCH becomes PCH, those packets will be forwarded to the right path instead of forwarding its original at its former PCH. Based on missing the *hello* message from the PCH or finding the *iAmNoLongerYourCH* flag is true, the SCH becomes

PCH and then sets its *iAmYourPCH* flag to true. The new PCH in its turn will then assign its SCH based on the above *distance-power-level algorithm* too. When an MT receives this flag, it will set its *myCH* variable to the *ID* of the sender (Algorithm 8).

Algorithm 7: Using *whoIsMySCH* and *iAmYourPCH* Flages by SCH

```
1  if myCH > 0 & myCH = SenderID then
2      if myID = whoIsMySCH then
3          if iAmNoLongerYourCH = true then
4              CancelMyCHTimer();
5              myCH = 0 ;
6              iAmYourPCH = true;
7              StartMyCHTimer();
8              SendERFNToUpdateRouting();
9          end
10         else
11             SetMySelfToBeSCH();
12             StartMyCHTimer();
13         end
14     end
15 end
```

Algorithm 8: Using *iAmYourPCH* Flag by MTs

```
1  if myCH > 0 & iAmYourPCH = true then
2      CancelMyCHTimer();
3      myCH = senderID;
4      StartMyCHTimer();
5  end
6  else
7      StartMyCHTimer();
8  end
```

3.5.4 The WarningThreshold Flag

The *warningThreshold* flag is used by an PCH to notify its SCH and its ZoneMTs, whether they can look for another PCH or not (Algorithm 9). If it is

set to false, its list of MTs know that they are not allowed to look for another PCH for at least one *hello* period. However, if the flag is set to true, its list of MTs can stay with their PCH, but they should look for another PCH or elect a new PCH (Algorithm 10). Notice, as we mentioned earlier, if $whoIsMySCH > 0$, instead of setting this flag, *iAmNoLongerYourCH* flag will be set to true to move the responsibility to the SCH and to trigger it to be the new PCH (Algorithm 11). Then, all zoneMTs that are within the transmission range of this new PCH will set their *myCH* variable to *whoIsMySCH* variable

Algorithm 9: Using *warningThreshold* & *whoIsMySCH* Flags by PCH

```
1  if myCH = 0 then
2      if my BPL < Threshold2 & my BPL ≥ Threshold3 then
3          if whoIsMySCH > 0 then
4              ListOfMyMTs = 0;
5              iAmNoLongerYourCH = true;
6              myCH = woIsMySCH;
7          end
8          else
9              warningThreshold = true;
10         end
11     end
12     if my BPL ≥ Threshold2 then
13         warningThreshold = false;
14     end
15 end
```

3.5.5 The IAmNoLongerYourCH Flag

The *iAmNoLongerYourCH* flag is used by the PCH to tell its SCH and its ZoneMT, whether it can support them for another hello period or not. Algorithm 11 and Algorithm 13 show how this flag is used by the PCH and the ZoneMT.

If *iAmNoLongerYourCH* flag is set to false, its ZoneMTs learn that they will stay with their current PCH for at least one *hello* period. However, if it is set to true for any reason, the SCH will be triggered to be the new PCH (Algorithm 12).

Algorithm 10: Using *warningThreshold* Flag by MT

```
1  if myCH > 0 & myCH = snederID then
2      if warningThreshold = true then
3          for AllMyNeighborList do
4              if neighborBPL > (Threshold_1 & my BPL) then
5                  SendMergeRequest(NeighborID);
6                  if Merge Accpet Recieved then
7                      SendDisjointMessage(myCH);
8                      CancelMyCHTimer();
9                      myCH = NeighborID;
10                     StartMyCHTimer()
11                 end
12             end
13             else
14                 ElectANEWCH() or StayWithmyCH();
15                 RestartMyCHTimer();
16                 KeepSearchingForANewCH;
17             end
18         end
19     end
20 end
```

The ZoneMTs would have to look for its SCH or to elect a new clusterhead by exchanging the *merge-request* and *merge-accept* messages according to the mechanisms described in the above. Each node maintains a neighborList, which contains its neighbors, and broadcasts it in its hello message, in order to be used in routing decisions.

Algorithm 11: Using *iAmNoLongerYourCH* Flag by PCH

```
1  if myCH = 0 then
2      if myBPL ≤ Threshold_3 then
3          ListOfMyMTs = 0;
4          iAmNoLongerYourCH = true;
5          myCH = -1;
6      end
7      else
8          iAmNoLongerYourCH = false;
9      end
10 end
```

Algorithm 12: Using *iAmNoLongerYourCH* Flag by SCH

```
1  if myCH = senderID & myID = whoIsMySCH then
2      if iAMNoLongerYourCH = true then
3          CancelMyCHTimer();
4          iAmYourPCH = true;
5          myCH = 0;
6          SendERFNToUpdateRouting();
7      end
8      else
9          StartMyCHTimer()
10     end
11 end
```

3.6 Summary

In the SERC/LC3R framework, we provide new approaches in the reclustering as well as in the multipath routing protocol. We make modifications to the cluster-based routing protocol to provide for each cluster two clusterheads (PCH and

Algorithm 13: Using *iAmNoLongerYourCH* by ZoneMT

```
1  if myCH = senderID then
2      if iAmNoLongerYourCH = true then
3          if whoIsMySCH > 0 then
4              deleteMyCHTimer ;
5              myCH = whoIsMySCH;
6          end
7          else
8              deleteMyCHTimer ;
9              myCH = -1;
10     end
11     else
12         startMyCHTimer;
13 end
```

SCH). The SCH works as a backup for the PCH and is the future leader for the cluster. In the SERC protocol, the cluster member that was supervised by the PCH and was being heard by the SCH will stay associated to the cluster. This member will be associated with the SCH immediately when its PCH can no longer be a clusterhead. Once the SCH is triggered to be a PCH, a new SCH is assigned by the new PCH. The smooth clusterhead transfer, when needed, from the PCH to the SCH will give the cluster more lifetime and will save the re-clustering communication overhead.

Moreover, the smooth transfer of the cluster leadership from the PCH to the SCH will localize the effect of the PCH failure. In the LC3R, since the SCH might fall into the transmission range of the upstream and downstream nodes of the routes that are going through its PCH, it can provide a new approach to the multipath routing protocol with no waste of bandwidth. The SCH could gather the reservation signaling messages and the data packets that were sent to its PCH. Simply, the main route can be established through the primary clusterheads, while the pack-up route can be established through the secondary clusterheads. In this

chapter we describe the SERC and the LC3R protocols and how they can provide solutions in clustering, routing and QoS areas in the MANETs.

Chapter 4

THE FRAMEWORK MODELLING

4.1 Introduction

In Chapter 3, we have provided the description of the SERC/LC3R Framework in details. We have seen that the basic idea of this framework is the existence and the clusterhead transfer from the PCH to the SCH. In this chapter, we will provide a simple network model that can help to define the conditions to select the SCH, and to analysis the clustering communication overhead and the cluster residence time to validate the framework features. Also, in this chapter, we will provide a model that can represent the PCH to the SCH transition. This model will help understand how the framework is working in the cluster level and how/when it can provide a better communication performance.

4.2 Network Model

We represent a mobile ad hoc network as a dynamic undirected graph. It can be modelled as a graph $G=(V,E)$, where V is the set of N nodes in the network and $E= \{(u,v):u,v \in V\}$ is the edge set of G.

Assume there is a cluster G_{PCH} such that node PCH is the present primary clusterhead of this cluster, and nodes in the set $X= \{x_1,x_2, \ldots ,x_m \mid x_i \in V(G), 1 \leq i \leq m\}$ are the set of node members of the cluster. Now we have $(x_i,PCH) \in E(G)$, based on the definition of cluster. Let $D_i(x_i,PCH)$ be the line of sight distance

between the pair (x_i,PCH) and R be the maximum transmission range of each node. Then, $D_i(x_i,PCH) \leq R$.

The conditions that are used by PCH to select a secondary clusterhead, denoted by SCH, among nodes in X are as follows:

- $D_{SCH}(SCH,PCH) = min\ \{\ D_i(x_i,\ PCH),\ \forall x_i \in X\}$, and
- SCH meets the energy requirement of being selected as a clusterhead. That means the battery power level of SCH should be greater than $THRESHOLD_1$.

Therefore, having been assigned as a SCH, a node should be the nearest member node to the PCH and should meet the energy requirement of being selected as a clusterhead. The SCH should be close to its PCH to have the same range to its cluster members and border nodes. The SCH is selected by using the *distance-power-level-based algorithm* which takes advantage of GPS. This algorithm uses two parameters: the node distance D from the PCH and its battery power level PBL. Therefore, each node, within the distance D from the PCH and its power level is BPL, is assigned a pair $DBPL = (D,BPL)$. The BPL of the eligible SCH should be equal to or greater than the $THRESHOLD_1$. Let x_1 has $DBPL_1 = (D_1,BPL_1)$ and x_2 has $DBPL_2$=(D_2,BPL_2). Then $DBPL_1 > DBPL_2$, if $D_1 < D_2$ and BPL_1 & $BPL_2 > THRESHOLD_1$. That is, x_1 has a priority over the other of the eligible SCHs if it is the nearest zoneMT to the PCH. This algorithm is run by the PCH and is used to set its *whoIsMySCH* flag to the *ID* of the eligible MT to be a SCH. Missing of the *Hello* message from the assigned SCH will lead to re-run this algorithm again. When the targeted MT acknowledges to be an SCH, it will prepare itself to be an SCH.

4.2.1 Clustering Overhead Analysis

Our goal in this section is to develop an analytical model that can be used as a tool in the provisioning and evaluation of clustering overhead in the WEAC and the SERC clustering algorithms. Mainly, we consider the clustering overhead (CO)that occurred in the setup phase to build the cluster and in the maintenance phase to maintain the cluster. Also, we consider the re-clustering overhead (RCO) that occurred when the PCH can no longer be a clusterhead. We assume that the cluster is only affected by two main reasons: mobility and PCH battery power consumption. In general, the clustering overhead CO is caused by the cluster setup and the cluster maintenance messages. The setup messages are the first exchange messages to build the cluster by sending *merge request* message and receiving *merge accept* message. The maintenance messages, which we consider as re-clustering overhead RCO, are used to maintain the cluster.

By using the WEAC protocol to maintain the cluster, the cluster member, due to its mobility or its CH mobility, should start negotiation by sending *merge request* message and receiving *merge accept* message to join or build a new cluster. Besides those messages, when the CH can no longer be a clusterhead due to its battery power reaching a certain threshold, the member needs to send *disjoint* message to its CH before starting another setup phase for a new cluster. Therefore, in WEAC, to maintain a cluster G_{PCH} with m members, the re-clustering overhead can be calculated by

$$\text{WEAC } RCO = \begin{cases} 2m & \text{due to mobility only} \\ m & \text{due to battery failure only} \end{cases} \tag{4.2-1}$$

Now, to find out the total clustering overhead CO, we first assume the probability that the CH will leave its cluster due to its power reaching $THRESHOLD_2$ is P_p. Second, during the cluster lifetime, in presence of mobility, we assume that some of the cluster members or the clusterhead itself will leave or move out the

cluster. Therefore, we assume the probability of changes in the cluster due to mobility is P_m. If the CH moves out the cluster, P_m will be equal to one. This is because all the members will start clustering negotiation. Then the total CO due to setup and maintenance messages will be

$$\text{WEAC } CO = 2m + mP_p + 2mP_m \quad \textit{where} \quad 0 \leq P_m, P_p \leq 1$$

If there is no mobility, P_m will be equal to zero, and its value will increase with the increase of the mobility rate which can be measured by the mobility speed. On the other hand, when there is no mobility, reaching the BPL of the PCH to $THRESHOLD_2$ is the only reason behind the cluster failure. Then P_p is equal to 1 when P_m is equal to 0. Therefore, if the cluster does not collapse by the CH or the members movements, it will collapse by the CH battery power failure. Then the P_p is the complement of P_m and P_p = 1 - P_m.

$$\begin{aligned} \text{WEAC } CO &= 2m + m(1 - P_m) + 2mP_m \\ &= m(3 + P_m) \quad \textit{where} \quad 0 \leq P_m \leq 1 \end{aligned} \tag{4.2-2}$$

The total WEAC CO that is generated in the network during the whole network run time is equal to the above quantity in (4.2-2) multiplied by the total number of generated clusters during the lifetime of the network, N_{gc}. Therefore the Total WEAC CO is

$$\text{Total WEAC } CO = N_{gc}(m(3 + P_m)) \tag{4.2-3}$$

Where m is the average node density per cluster. The WEAC CO per node will be

$$\text{WEAC } CO = \frac{N_{gc}}{N} m(3 + P_m) \tag{4.2-4}$$

Where N is the total number of nodes in the network. As a main WEAC protocol characteristic [SHE03], each node will take its turn to be a clusterhead at least once. However, in the presence of mobility the node might be elected more than once to be a clusterhead. Therefore $N_{gc} \geq N$, and then the $\frac{N_{gc}}{N}$ is equal to or greater than one.

In SERC, the *Hello* message has an ability to reform the cluster in the presence of the SCH which saves most of the clustering exchange messages. Suppose that the assigned SCH can be heard by the members denoted by the set $Z = \{z_1, z_2, \ldots, z_n \mid z_i \in X), 1 \leq i \leq n\}$. Therefore, the re-clustering overhead (RCO) due to mobility is given by

$$\text{SERC } RCO_m = \begin{cases} 2m & \text{if } n = 0, \quad \textit{no qualified SCH} \\ 2(m-n) & \text{if } 1 \leq n < m \\ 0 & \text{if } n = m \end{cases} \tag{4.2-5}$$

On the other hand, when the cluster failure occurs due to its PCH battery failure and a *disjoint* message has taken place, the RCO is given by

$$\text{SERC } RCO_p = \begin{cases} m & \text{if } n = 0, \quad \textit{no qualified SCH} \\ m-n & \text{if } 1 \leq n < m, \\ 0 & \text{if } n = m \end{cases} \tag{4.2-6}$$

We assume that P_s is the average of the probability that the SCH can cover the members that are covered by its PCH, then $n = m \cdot P_s$ and the (4.2-5) and (4.2-6) will be

$$\text{SERC } RCO_m = 2(m - mP_s)$$

$$= 2m(1 - P_s) \qquad \textit{where} \quad 0 \leq P_s \leq 1 \tag{4.2-7}$$

$$\text{SERC } RCO_p = m - mP_s$$

$$= m(1 - P_s) \qquad \textit{where} \quad 0 \leq P_s \leq 1 \tag{4.2-8}$$

As shown above, the SERC protocol can save all the RCO to reform the cluster if its members can be reached and heard by its SCH.

When the SCH is triggered to be a PCH, the cluster will be entered a new setup phase. However, the cluster members that were reached by the SCH will not exchange the regular setup messages. Therefore, the average cluster setup messages will be reduced too by the factor (1 - P_s). Now the average clustering overhead CO per node of the SERC protocol can be given by

$$\text{SERC } CO = \frac{N_{gc}}{N}(2m(1-P_s) + m(1-P_s)(1-P_m) + (2m(1-P_s))P_m)$$

$$= \frac{N_{gc}}{N}m(3+P_m)(1-P_s) \tag{4.2-9}$$

In this equation, the CO of the SERC protocol increases as the members that can be reached by the SCH decrease. In the worst case, when there is no qualified SCH, the CO of the SERC protocol is equal to the CO of the WEAC protocol.

4.2.2 Cluster Residence Time Analysis

Cluster Residence Time is defined as the time a node remains associated with a given cluster. This is measured as the duration from the execution time of the *merge-accept* message until the execution time of the *disjoint* message for such a node. This metric will be used to assess the stability of the cluster topology. In this section we will explain how this time can be estimated.

Suppose that τ is the average clusterhead duration time. The average cluster residence time for each member node is at most equal to τ, not only in the WEAC protocol, but also in all the other clustering algorithms. However, the SERC protocol can provide more cluster stability by extending, in some cases, the cluster residence time of each member. In its first round of creating a cluster, if there is no mobility, the cluster residence time for each member node is τ. In the second round, when the SCH clusterhead is used, the cluster residence time of the nodes

(n) that are supervised by the SCH will be 2τ while the cluster residence time of the rest nodes (m-n) is equal to τ. Then, the average cluster residence time (CRT) for each member node is given by

$$CRT = \frac{(m-n)\tau + 2n\tau}{m}$$

$$CRT = \frac{m+n}{m}\tau \tag{4.2-10}$$

Therefore,

$$CRT = \begin{cases} \tau & if \quad n = 0, \\ \frac{m+n}{m}\tau & if \quad 1 \leq n < m, \\ 2\tau & if \quad n = m \end{cases} \tag{4.2-11}$$

Since $n = m \cdot P_s$, then

$$CRT = \begin{cases} \tau & if \quad P_s = 0, \\ \frac{m+mP_s}{m}\tau & if \quad 0 < P_s < 1, \\ 2\tau & if \quad P_s = 1 \end{cases}$$

$$CRT = \begin{cases} \tau & if \quad P_s = 0, \\ (1+P_s)\,\tau & if \quad 0 < P_s < 1, \\ 2\tau & if \quad P_s = 1 \end{cases} \tag{4.2-12}$$

As stated in the above equations, the CRT might be doubled for the cluster members that can be heard by its SCH. This is in the second round of cluster leadership transfer from the PCH to its SCH. In the third and the subsequent rounds, the average CRT is expected to increase.

Theorem 4.2.1 *In a cluster with m members, if the initial PCH has a distance to each member at most equal to* $\frac{R}{2}$ *and there is no mobility, then the cluster residence time for each member node is equal to* $m\tau$.

Proof: Since all the cluster members are within $\frac{R}{2}$ to its first PCH, the distance from edge to edge of the cluster is equal to R as shown in Fig. 4.1. Therefore, all the cluster members are within transmission range of each other and once anyone of

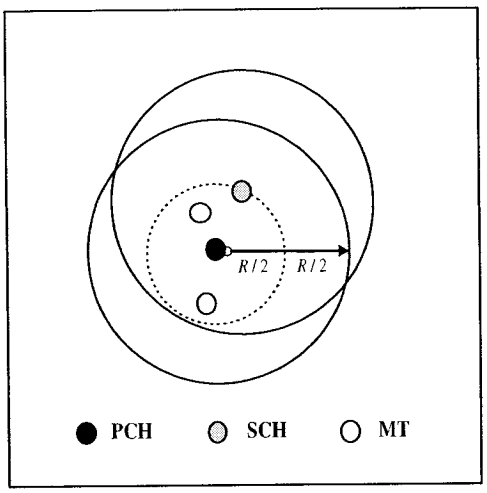


Figure 4.1: Proof of Theorem 4.2.1

them becomes a clusterhead, it will be heard by all the other m members. As shown in Fig. 4.1, the SCH transmission range can cover all the cluster members. Hence, the cluster residence time for each member will last for m rounds and extended to be $m\tau$. This is completely right in the absence of mobility. However, in the presence of mobility, some cluster members may move away from their clusterhead during its round. Therefore, the average cluster residence time will be affected and will be less than $m\tau$.

4.2.3 Protocol Properties

This section contains a description of the main properties of this protocol. The proof of each of these properties is also provided.

Lemma 4.2.1 *If (x, SCH) $\in$ E(G), $\forall$ x $\in$ X, to reform the cluster in case of the PCH failure, the* RCO *is equal to zero.*

Proof: When there is a link between the SCH and each member of its PCH, its *Hello* message would be heard by each member. Therefore, based on the *iAmYour-PCH* flag, the cluster will be reformed, its SCH will be the PCH and its members will be the former PCH's members. Each member will set its *myCH* flag to the

value of *whoIsMySCH* value. Hence, no need for negotiation messages to be generated by each node to reform the cluster or to join another cluster. Therefore, the re-clustering overhead in this case will be equal to zero.

Lemma 4.2.2 *If (x, SCH) $\in$ E(G), $\forall$ x $\in$ X and there is no mobility, the cluster residence time for all the cluster member will be doubled.*

Proof: When the leadership of the cluster has been transferred successfully from the PCH to the SCH, the cluster member that was supervised by the PCH and was being heard by the SCH will stay associated with the cluster. Hence, its cluster residence time will be extended. This time might be doubled if there is no mobility and the power consumption is the only reason behind the cluster collapse.

Lemma 4.2.3 *If (x, SCH) $\in$ E(G), $\forall$ x $\in$ X, to update the routing table in case of a PCH failure, instead of full dump update, an incremental update is used and its number of entry is equal to one.*

Proof: From Lemma 4.2.1 the cluster will be reformed completely and there is no change in its members list. The only change is that the SCH will become a PCH. Therefore, this change is the only update to be entered in the incremental update packet.

Lemma 4.2.4 *If (x, SCH) $\in$ E(G), $\forall$ x $\in$ X, all the routes that are going through the PCH will be recovered by the SCH with no need to invoke the route recovery process.*

Proof: The routes that are going through the PCH are coming from Border Mobile Terminals (BMTs) which are regular member nodes. Once those terminals are covered by the SCH and they are within its transmission range, the routes that are coming from them will be rerouted by the SCH. Therefore, no need to invoke the regular route recovery process of the applied routing protocol.

Lemma 4.2.5 *The cluster reformation time in case of a successful transfer from the PCH to the SCH will be within one Hello period.*

Proof: Since the hello message is the only message that is used by the SCH to reform the cluster, then any node within the transmission range of the SCH will hear it and set its *myCH* variable to the *whoIsMySCH* value immediately. This indicates that the SERC protocol has a very low convergence time. Therefore, each and every node will know its PCH within one *Hello* period.

4.2.4 Simulation Results

A simulation model is developed by using discrete event computer simulation to evaluate the clustering overhead of WEAC and our proposed protocols. This simulator is written using the Java programming language. We assume that every mobile node is aware of its location which can be obtained from GPS or some other location service systems. We also assume that all mobile nodes have the same transmission range. The simulated environment is 1000x1000 unit rectangular flat area. Initially, each mobile node is assigned a unique node ID, a random x-y position, a random mobility speed, and a random power level greater than 90% of its maximum battery power. The mobility model, the traffic model, and the power consumption model that are used in this simulation are described in details in Chapter 5. The node moves based on the random mobility model with random speed uniformly distributed between 10 and 20 units/second. The *Hello* messages are broadcasted every 1 second. Each simulation scenario is run for enough time to reach and collect the desired data at the steady state. Several runs of each simulation scenario are conducted (each run representing a set of random initial parameters) to obtain statistically confident averages.

These experiments are developed to evaluate the clustering overhead of the SERC and the WEAC protocols, and to provide simulation and analytical results of

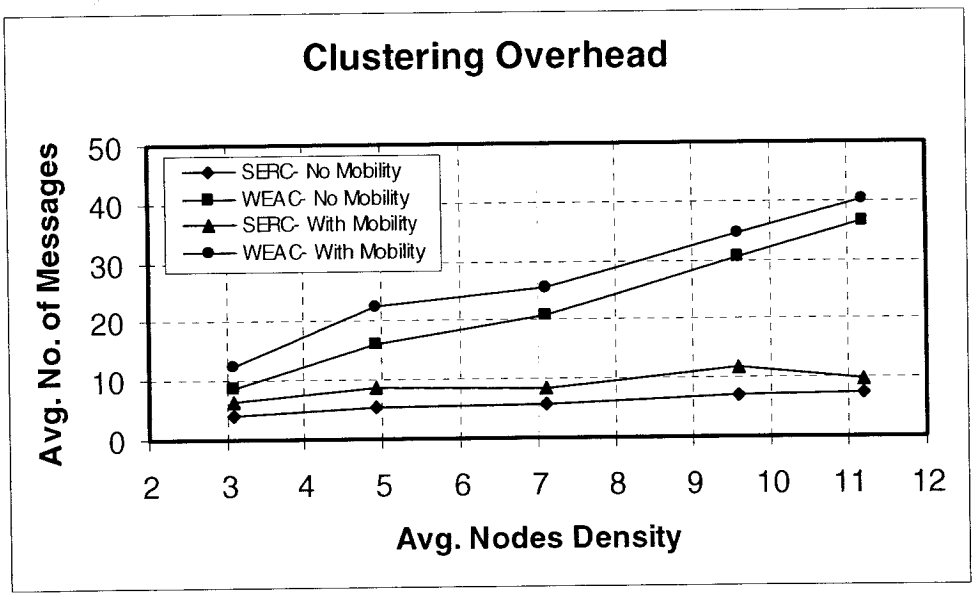


Figure 4.2: Average Clustering Overhead vs. Average Node Density in the Cluster

both of them in different mobility scenarios. The analytical results are obtained by incorporating (4.2-4) and (4.2-9) in the simulator, while the simulation results are taken directly without using them. In these experiments, the number of nodes in the network is set to 50 nodes. We set different node densities by fixing the topology size and varying the transmission range with the same constant number of nodes. The simulation experiments are conducted for different wireless transmission range varying from 150 to 350 units to have different node densities.

Before providing simulation results, we need to provide the following observations:

- The average node density m in a cluster is controlled by the transmission range. The longer the transmission range is the higher node density.
- P_s is affected by the transmission range for the same reason above. Therefore, by setting the transmission range, we will get different values for m and P_s.
- P_m is controlled by the average mobility speed of nodes. Also, it is affected by the node density. This is because the less the transmission range is the

more quickly the nodes would go out of their PCH covering area. When there is no mobility, P_m is equal to zero.

- N_{gc} is increased by the increase of the mobility speed and the decrease of the node transmission range. This is because the increase of the mobility speed and the decrease of the node transmission range would increase the probability that the cluster may collapse while its clusterhead still has enough battery power. Therefore, the clusterhead might be re-elected again. When there is no mobility, N_{gc} is expected to be equal to the number of nodes N and then $\frac{N_{gc}}{N}$ is expected to be equal to one.

In Fig. 4.2, in the WEAC protocol, the average number of cluster formation and maintenance messages generated by each node is proportionally increased with the increase of the node density of the cluster. This is obviously right because increasing the node density of the cluster will increase the power consumption of its PCH and then decrease the PCH duration time, as mentioned above. Therefore, the node will join more clusters during its lifetime, and then it needs to exchange more messages to do so. On the other hand, since the SERC protocol is mainly dependent on the *Hello* messages to reform the cluster, the average number of cluster formation and maintenance messages generated by each node is reduced. The improvement in saving this clustering overhead increases with increasing the nodes density. This is because the higher nodes density increases the probability that the candidate SCH is more close to its PCH location and then its transmission range can cover most of the cluster member nodes. Therefore, with higher density, it is very rare for the cluster member to invoke the re-clustering mechanism that is used by the WEAC protocol to build or join a new cluster. In the presence of mobility, it is obvious that the node might move out of its PCH transmission range, and then it will need to negotiate joining or building a new cluster. Therefore, the

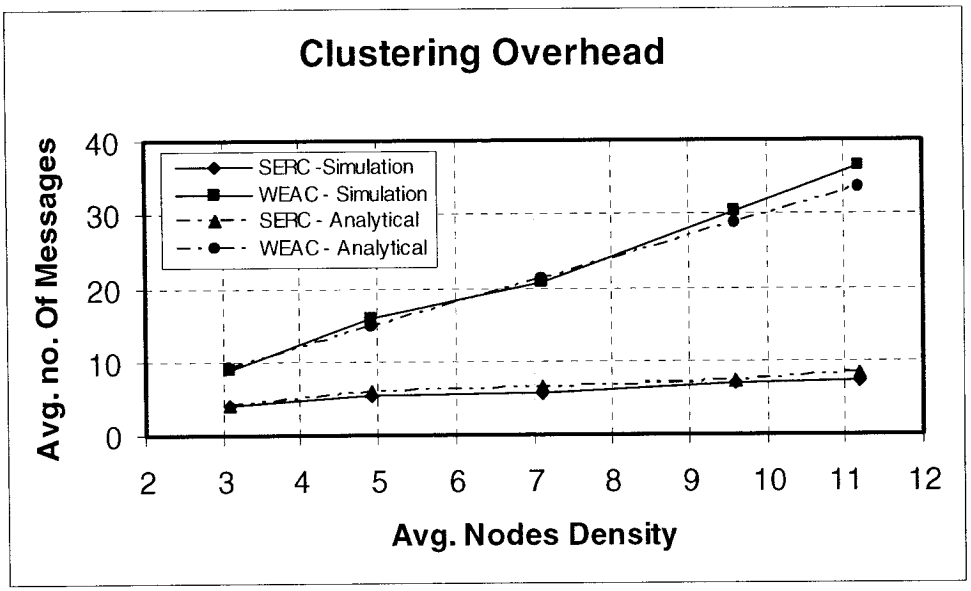


Figure 4.3: Average Clustering Overhead vs. Average Node Density in the Cluster with No Mobility

clustering overhead in the presence of mobility is more than the overhead if there is no mobility. However, the SERC clustering algorithm still outperforms the WEAC protocol.

Figs. 4.3 and 4.4 provide a comparison between the simulation and the analytical results with and without mobility for both protocols: SERC and WEAC. These results show how our analytical model is accurate and how the clustering overhead with different node densities can be estimated. Fig. 4.5 shows the effect of the value of P_s on the clustering overhead. To show the P_s effect only, we use different fixed values for the cluster node density, m, and we assume there is no mobility, which means P_m is set to zero. Also, Fig. 4.5 shows that with increasing P_s the clustering overhead is proportionally decreased. This result is obtained by using (4.2-4) and (4.2-9).

4.3 PCH to SCH Transition Modelling

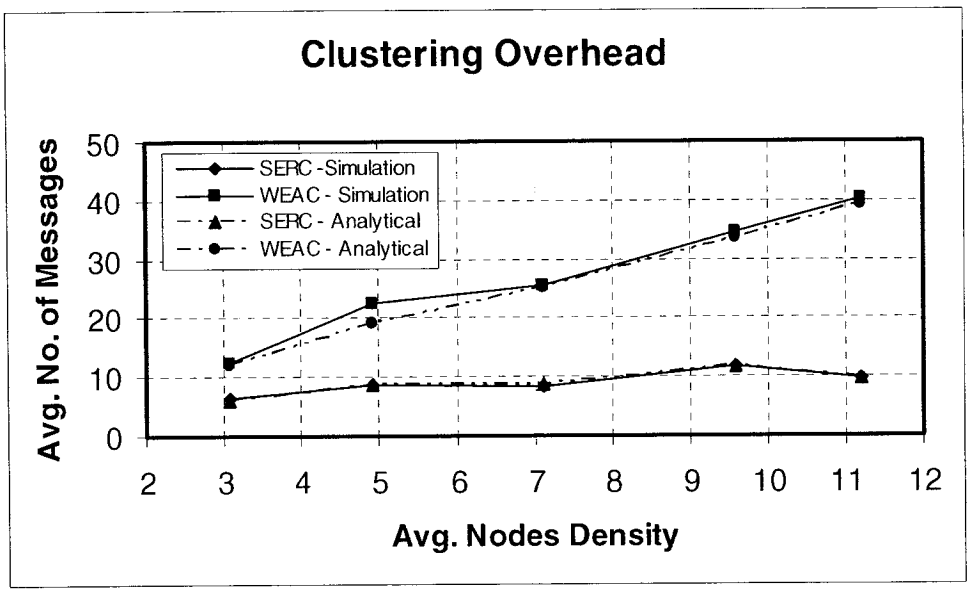


Figure 4.4: Average Clustering Overhead vs. Average Node Density in the Cluster with Mobility

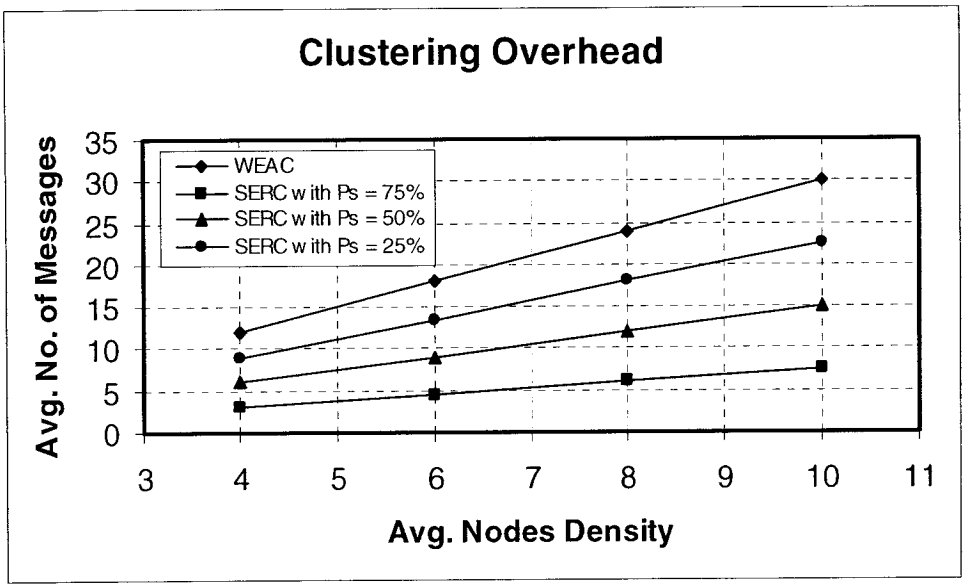


Figure 4.5: Average Clustering Overhead vs. Average Node Density in the Cluster with different Values of P_s

4.3.1 Markov Chain Model

Several of the most powerful analytic techniques for evaluating computer system performance (and many other systems) are based on the theory of Markov chains [GRI94, ISA95]. A Markov chain is a special case of a Markov process, which itself is a special case of a random or stochastic process. The Markov analysis is a method of analyzing repairable systems with constant failure and repair rates. Systems are described by state transition diagrams and may be used to model systems that exhibit strong dependencies. Markov states represent all possible "conditions" that the system can exist in. The system can only ever be in one state at a time. A single state must be set up as the initial starting state. Transition rates represent the rate at which the Markov diagram moves from one state to another. For example, the transition rate from a working state to a failed state is represented by the failure rate whereas the transition from a failed state to a working state is represented by the repair rate.

The Markov model is analyzed in order to determine such measures as the probability of being in a given state at a given point in time, the amount of time a system is expected to spend in a given state, as well as the expected number of transitions between states, for instance representing the number of failures and repairs.

Graphically, a Markov model is usually represented in the form of nodes and arrows, where the nodes represent the states that the system can be in, and the arrows represent the transitions between the states. The numerical parameters of the model, λ_{ij}, where i and j are used to denote the source and destination of the transitions, is referred to the transition rate, or transition speed, going from state i to state j.

4.3.2 Analysis of Markov Models

A Markov chain is a discrete-state random process in which the evolution of the state of the process beginning at a time t (continuous-time chain) or n (discrete-time chain) depends only on the current state X_t or X_n , and not how the chain has reached its current state or how long it has been in that state. To be more precise, let's begin with a random sequence $\{ X_t;\ t = 0, 1, 2, \ldots \}$, if for every time t and all possible states of the X_t,

$$P[X_{t+1} = j | X_0 = a, X_1 = b, ..., X_t = i] = P[X_{t+1} = j | X_t = i] \quad (4.3\text{-}13)$$

This is referred to as the Markov property. This Markovian property means that the probability of any future state given the entire past and present states, is independent of the past events and depends only on the present state of the process. The conditional probabilities P $[X_{t+1} = \text{j} — X_t = i]$ are called transition probabilities and are denoted by $\lambda_{ij}(t)$.

We can write a simple recursion equation for the state probabilities of a Markov chain. Suppose the probability that X_{t+1} takes on state j is $P_j(t+1)$. Now at time t the system may be in any state i with probability $P_i(t)$, $i = 1, 2, \ldots, N$. For each state i, there is a transition probability $\lambda_{ij}(t)$ that the system will make the transition from this state to state j. Hence

$$P_j(t+1) = \sum_{i=1}^{N} P_i(t) \cdot \lambda_{ij}(t) \quad (4.3\text{-}14)$$

or in matrix and vector representation

$$\mathbf{P}(t+1) = \mathbf{P}(t) \cdot \mathbf{\Delta}(t) \quad (4.3\text{-}15)$$

where $\mathbf{P}(t)$ is an N-dimensional row vector with elements $\mathbf{P}_j(t)$, $j = 1, 2, \ldots, N$, and $\mathbf{\Delta}$ is an $N\ x\ N$ matrix with elements $\lambda_{ij}(t)$ and is called Markov chain probability matrix at time t. if all $\lambda_{ij}(t)$ are independent of t, we have

$$\mathbf{P}(t+1) = \mathbf{P}(t) \cdot \mathbf{\Delta} \quad (4.3\text{-}16)$$

From this equation, we have

$$\mathbf{P}(t) = \mathbf{P}(0) \cdot \mathbf{\Delta}(0)\mathbf{\Delta}(1)\mathbf{\Delta}(2)...\mathbf{\Delta}(t-1) \tag{4.3-17}$$

if the process is nonstationary, and

$$\mathbf{P}(t) = \mathbf{P}(0) \cdot \mathbf{\Delta}^t \tag{4.3-18}$$

if the process is stationary. The later equation shows that the state probabilities of a Markovain chain are completely determined for all $t > 0$, if we know the transition probability matrix $\mathbf{\Delta}$ and the initial state probability vector $\mathbf{P}(0)$.

It is clear from the definition of conditional probabilities that the elements of the transition matrix $\mathbf{\Delta}$ must satisfy the following properties,

$$\lambda_{ij} \geq 0 \qquad \text{for all} \quad i,j \qquad \text{and} \qquad t = 0,1,2.. \tag{4.3-19}$$

and

$$\sum_{j=1}^{N} \lambda_{ij} = 1 \qquad \text{for all} \quad i \qquad \text{and} \qquad t = 0,1,2.. \tag{4.3-20}$$

Matrices satisfying those two conditions are called stochastic. Any stochastic matrix may serve as a transition probability matrix.

4.3.3 Problem Formulation

In the SERC protocol, the main idea is the existence and the smooth clusterhead transfer between the PCH and the SCH. It is designed to let these nodes communicate to each other by using the *Hello* message to take the decisions to do the transfer in the right place at the right time. These decisions take place based on the mobility and the battery power levels of these nodes, and they are executed by setting certain flags in their *Hello* messages. However, these *Hello* messages might be corrupted due to collisions as example, or they might not be executed

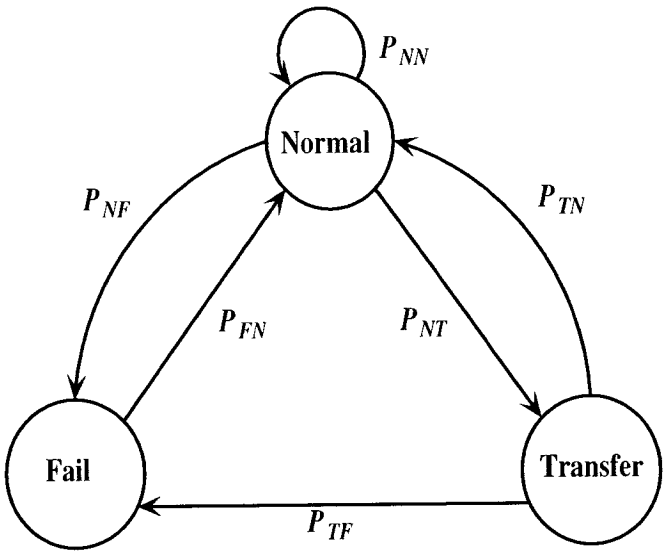


Figure 4.6: A transition diagram shows the three states and the probabilities of going from one state to another

in the right time. Therefore, these assumptions result in a simple model behavior for a cluster, such that we can apply a three state Markov chain describing the possible states for that cluster.

In this section, we formulate the problem and then derive the approximate mathematical model that can represent the SERC protocol. This model is based on Markov chain illustrated in Figure 4.6. The significance of the states of this Markov chain is the following:

- **Normal State:** This is the initial state when the cluster has elected its PCH and assigned its SCH.

- **Transfer State:** The state when the cluster leadership has been successfully transferred from the PCH to the SCH based on an agreement stated in the *Hello* messages such as *iAmNoLongerYourCH* and *iAmYourPCH* flags.

- **Fail state:** The state when the protocol fails to do the right clusterhead transfer from the PCH to the SCH. This state might exist in two possible

ways. The first way, from the normal state, if the *Hello* messages that are sent by the PCH get corrupted. Therefore, the SCH misses its PCH *Hello* message and triggers itself to be the PCH in presence of its PCH. That means the cluster might be partitioned into two clusters or the clusterhead transfers to the SCH while the PCH still has the ability to complete its job. The second way, from the transfer state, when the decision has taken to do the transfer but the SCH becomes not suitable to be a clusterhead. Therefore, the transfer comes in bad time and the SCH selection has not be updated.

Let the steady state probability of being in states Normal, Transfer, and Fail states represented by π_N, π_T, and π_F, respectively. We suppose that the *Normal* state is considered as an initial state ($t = 0$), so the initial matrix is stated as [1 0 0]. The analysis of a Markov model generally involves in the computation of the steady state probability that the system will be in a given state as a function of time. For instance, assuming that at $t = 0$ the system is in *Normal* state, computations are performed to determine the probabilities that the system will be in *Normal*, *Transfer* and *Fail* states at any given time t. At each point in time, the sum of probabilities of the states must add up to 1: if the probability that the system is in one state decreases by a certain amount x, that same amount x must be distributed over the other states in the system. Therefore, Fig. 4.6 can be represented in a mathematical way, and based on equation 4.3-18, by the following equation:

$$[\pi_N \quad \pi_T \quad \pi_F] = [1 \quad 0 \quad 0] \begin{bmatrix} P_{NN} & P_{NT} & P_{NF} \\ P_{TN} & P_{TT} & P_{TF} \\ P_{FN} & P_{FT} & P_{FF} \end{bmatrix} \quad (4.3\text{-}21)$$

The steady state probability of each state can be calculated based on the transition probabilities by using the Markov chain model stated above. However,

before calculating the transition probabilities between the above states, we will discuss some factors that might have some impact on those states.

4.3.3.1 Impact of Beacon Interval Length

The length of beacon intervals or arrival *Hello* messaging rate (λ_h) has an impact on the hosts sensitivity to environmental changes and power consumptions. Increasing this rate will increase the topology update speed which reduces the time spent in the *Transfer* state and then the reclustering will be converged very fast. However, this will lead to more power consumption. Although a larger interval will only contribute to a little more saving in power consumption [TSE03], tradeoff between the sensitivity to the environmental changes and the saving of power consumptions can be obtained by increasing the λ_h of the node only when it is either PCH or SCH.

4.3.3.2 Impact of Mobility

In general when the mobility rate, which is based on the mobile terminals average speed, is high, the probability that the mobile node moves out of its cluster within a period is increased. Therefore, the mobility rate has a major impact on the leaving of the PCH away from its cluster and on the moving of the SCH out of its PCH transmission range. The leaving of the PCH from its cluster leads to transfer the cluster leadership to the SCH which takes, in the Markov model above, the cluster from the *Normal* state to the *Transfer* state. While the moving of the SCH out of its PCH leads to assigning a new SCH. If the SCH election is not updated and the cluster is in the *Transfer* state, the cluster leadership transfer will fail and then the *Fail* state takes a place. Thus, the transition probabilities P_{NT}, P_{TN} and P_{TF} are affected by the mobility rate.

The probability that the PCH leaves its cluster can be calculated under the following assumptions:

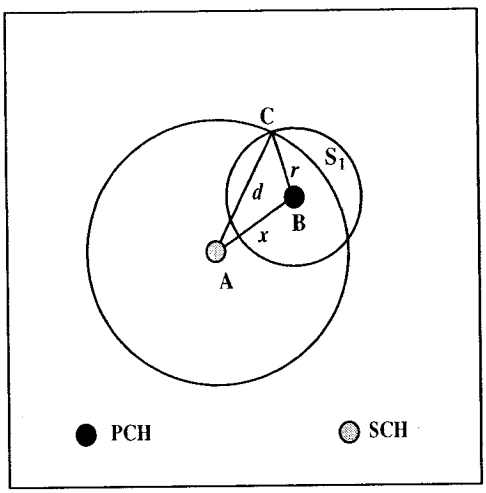


Figure 4.7: Calculation of the probability that the PCH moves into the shaded area, within a time interval t

- For a short period of length t, each node moves independently toward a random direction in [0,2π] with a constant speed v that is uniformly distributed in [0,v_{max}].
- The maximum transmission range of a node is d.
- All nodes are randomly distributed within an area S_0.

As shown in Figure 4.7, suppose the SCH is located in position A with its member PCH in position B. The maximum range of the SCH is AC=d, and the distance between the PCH and the SCH is x ($< d$). Let BC=r=$v_{max}.t$. The probability that the PCH leaves the cluster out of the maximum transmission range of the SCH within t is the probability that the PCH moves into the shaded area in Figure 4.7 in time t and it is denoted by P_L. We consider three cases [LI03]:

- Case I: $0 < r < d$

$$P_L = \int_{d-r}^{d} \frac{2\pi x}{S_0} \frac{S_1}{\pi r^2} dx$$

$$= \int_{d-r}^{d} \frac{2xS_1}{S_0 r^2} dx$$

where

$$S_1 = (\pi - \alpha_2)r^2 - (\alpha_1 d^2 - dx \sin \alpha_1),$$

$$\alpha_1 = \angle CAB = \arccos \frac{x^2 + d^2 - r^2}{2xd},$$

$$\alpha_2 = \angle CBA = \arccos \frac{x^2 + r^2 - d^2}{2xr}.$$

- Case II: $d \leq r < 2d$

$$P_L = \int_0^{r-d} \frac{2\pi x}{S_0} \frac{\pi(r^2 - d^2)}{\pi r^2} dx + \int_{r-d}^{d} \frac{2\pi x}{S_0} \frac{S_1}{\pi r^2} dx$$

$$= \int_0^{r-d} \frac{2\pi x}{S_0} \frac{\pi(r^2 - d^2)}{\pi r^2} dx + \int_{r-d}^{d} \frac{2xS_1}{S_0 r^2} dx$$

$$= \frac{\pi(r+d)}{S_0 r^2}(r-d)^3 + \int_{r-d}^{d} \frac{2xS_1}{S_0 r^2} dx$$

- case III: $r \geq 2d$.

$$P_L = \int_0^{d} \frac{2\pi x}{S_0} \frac{\pi(r^2 - d^2)}{\pi r^2} dx$$

$$= \frac{\pi(r^2 - d^2)d^2}{S_0 r^2}$$

P_L is used to determine the transition probability P_{NT}.

4.3.3.3 Impact of the Traffic Load

No doubt, a higher traffic load incurs higher power consumption, which is reasonable since hosts have less chance to sleep. A higher traffic load makes multiple packets may be transmitted in one beacon interval, which increases the power consumption rate. The increase of power consumption due to increase of traffic load will lead to the increase in the probability that the BPL of the PCH will be less than $THRESHOLD_2$, and then the probability of transferring from the PCH to the SCH. Therefore, the major impact that the power consumption rate, P_P, has is on the transition probability P_{NT} because it represents the transition from the PCH to its SCH. On the other hand, if the SCH selection is not updated, its power might be dropped to be less than $THRESHOLD_1$ and then it cannot be suitable to be a clusterhead. Therefore, it also has an impact on the transition probabilities P_{TF} as well as P_{TN}.

4.3.3.4 Impact of the SCH Selection Update

When the SCH has been elected and assigned by its PCH, the PCH should regularly monitor its SCH to check its suitability or its qualification to be a clusterhead in the near future. Therefore, the SCH selection should be updated. This update can happen on each *Hello* message interval. However, since the SCH has to conduct many functions and is involved in buffering the sent messages to its PCH, the update interval needs to be larger than the *Hello* message interval. In this section, we determine the maximum SCH selection update interval that can give the PCH an ability to overcome the changes that happened in the cluster and at the same time can give the SCH an ability to store the recent packets to its PCH. The failure of the clusterhead transfer, when it is needed by the cluster, happens when the probability of the update of the SCH selection is less than the probability of the SCH changes.

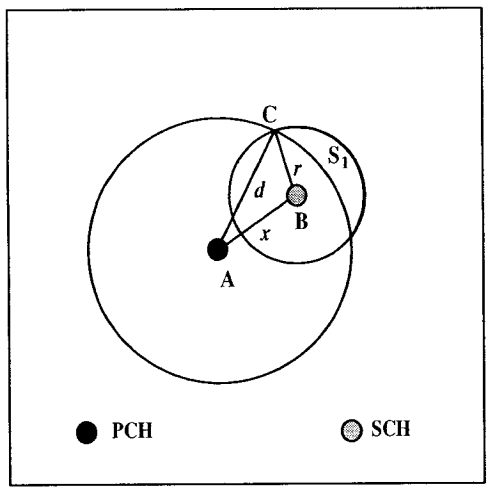


Figure 4.8: Calculation of the probability that the SCH moves into the shaded area, within a time interval t

To calculate the update interval, we need first to calculate the probability of the SCH changes. The probability that the SCH can no longer be qualified to be the future clusterhead can be calculated under the following assumptions:

- The probability that the BPL of the SCH may drop down $THRESHOLD_1$ is P_{P-SCH}.
- For a short period of length t, each node moves independently toward a random direction in [0,2π] with a constant speed v that is uniformly distributed in [0,v_{max}].
- The maximum transmission range of a node is d.
- All nodes are randomly distributed within an area S_0 and the average cluster members is m.

As shown in Figure 4.8, suppose the PCH is located in position A with its member SCH in position B. The maximum range of the PCH is AC=d, and the distance between the PCH and the SCH is x ($< d$). Let BC=r=$v_{max}.t$. The probability that the SCH moves out of the maximum transmission range of the

PCH within t is the probability that the SCH moves into the shaded area in Figure 4.8 in time t and it is denoted by P_M. We consider three cases:

- Case I: $0 < r < d$

$$P_M = \int_{d-r}^{d} \frac{2\pi x}{S_0} \frac{S_1}{\pi r^2} dx$$

$$= \int_{d-r}^{d} \frac{2xS_1}{S_0 r^2} dx$$

where

$$S_1 = (\pi - \alpha_2)r^2 - (\alpha_1 d^2 - dx \sin \alpha_1),$$

$$\alpha_1 = \angle CAB = \arccos \frac{x^2 + d^2 - r^2}{2xd},$$

$$\alpha_2 = \angle CBA = \arccos \frac{x^2 + r^2 - d^2}{2xr}.$$

- Case II: $d \leq r < 2d$

$$P_M = \int_0^{r-d} \frac{2\pi x}{S_0} \frac{\pi(r^2 - d^2)}{\pi r^2} dx + \int_{r-d}^{d} \frac{2\pi x}{S_0} \frac{S_1}{\pi r^2} dx$$

$$= \int_0^{r-d} \frac{2\pi x}{S_0} \frac{\pi(r^2 - d^2)}{\pi r^2} dx + \int_{r-d}^{d} \frac{2xS_1}{S_0 r^2} dx$$

$$= \frac{\pi(r+d)}{S_0 r^2}(r-d)^3 + \int_{r-d}^{d} \frac{2xS_1}{S_0 r^2} dx$$

- case III: $r \geq 2d$.

$$P_M = \int_0^{d} \frac{2\pi x}{S_0} \frac{\pi(r^2 - d^2)}{\pi r^2} dx$$

$$= \frac{\pi(r^2 - d^2)d^2}{S_0 r^2}$$

The probability that the SCH does not move out of its PCH transmission range is (1 - P_M) and the probability that the BPL of the SCH does not drop below the $THRESHOLD_1$ is (1 - P_{P-SCH}). Thus, the probability of the SCH changes in a cluster with m members is

$$P_{change} = 1 - (1 - P_M)^m (1 - P_{P-SCH})^m \tag{4.3-22}$$

The probability of the SCH changes has a major impact on the transition probabilities P_{TN} and P_{TF}. This is because the SCH changes can give information whether the SCH is still qualified, as it is selected, to be a clusterhead in the near future or its qualification has been changed and it becomes not suitable to be a clusterhead. Therefore, the transition probabilities P_{TN} and P_{TF} depend on the probability that the SCH selection is updated, P_{update}. Hence, by estimating P_{change}, we can estimate the maximum update interval of the SCH selection.

Given a predetermined update probability P_{update}, we can determine the SCH selection update interval t such that $P_{change} \leq P_{update}$.

To demonstrate how the period of SCH selection is affected by the maximum speed v_{max} and the update probability P_{update}, we consider a scenario in which 50 nodes are randomly distributed inside 1000x1000 unit rectangular flat area. Assuming that the average of cluster members is 5 and the traffic load is CBR with a rate of one packet per second per node. The maximum transmission range is $d_{max} = 250$ unit. Figure 4.9 shows the curve of the SCH selection update period versus the maximum speed with respect to different values of P_{update}. For example, to ensure the probability of cluster changes is below 0.2, the SCH selection update period decreases from 50 sec to 5 sec when the maximum nodal speed increases from 2 unit/sec to 20 unit/sec.

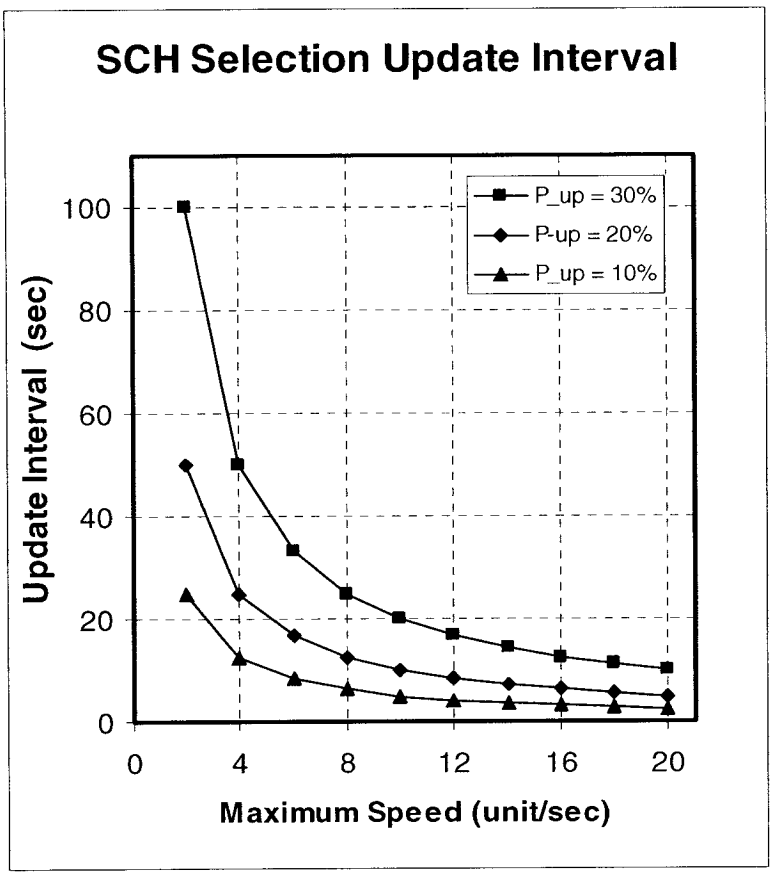


Figure 4.9: The SCH Update Interval vs. the Maximum Speed with Respect to Different Values of P_{update}

4.3.3.5 Impact of the Wireless Channel Characteristics

It is well known that the wireless network is not able to achieve the fixed network performance because wireless communications experience higher error rates than wired communications. The wireless channel is considered as a lossy and unreliable link because its quality is changing with time. It is easily affected by attenuation, interference, and multipath fading. Moreover, MANET, in particular, might be affected by the hidden terminal phenomenon. It is challenging for wireless network protocol developers to consider the large number of factors that affect the error performance of wireless channels. Since this is out of the scope of this work, the reader is referred to [KON03, BAI03] to know more about the wireless channel error models. In the above Markov Model, the cluster enters the *Fail* state when it initiates an unsuccessful clusterhead transfer due to wireless channel errors.

4.3.4 Model Analysis

Now, we proceed to calculate the transition probabilities of the Markov chain. P_{NT} depends on the probability of changes, in the PCH due to mobility (leaving its cluster) or power consumption (its BPL becomes less than $THRESHOLD_2$). This probability in its turn depends on the probability that the PCH leaves its cluster (moves out of its SCH transmission range), P_L, and depends on the probability that the BPL of the PCH drops below $THRESHOLD_2$, P_{P-PCH}. Then, the probability that the PCH does not move out of its SCH transmission range is (1 - P_L) and the probability that the BPL of the PCH does not drop below $THRESHOLD_2$ is (1 - P_{P-PCH}). Thus the probability of the clusterhead changes and the transition probability from the Normal State to the Transfer is

$$P_{NT} = 1 - (1 - P_L)(1 - P_{P-PCH}) \tag{4.3-23}$$

The transition probabilities P_{TN} and P_{TF}, as stated earlier, depend on the probability that the SCH selection is updated, P_{update}. Therefore, if the probability

of the changes in the SCH behavior, P_{change} is less than P_{update}, the SCH selection is updated and the transfer will be done successfully. Then

$$P_{TN} = \begin{cases} 1 & if \quad P_{change} \leq P_{update}, \\ 0 & if \quad P_{change} > P_{update} \end{cases} \tag{4.3-24}$$

and

$$P_{TF} = \begin{cases} 0 & if \quad P_{change} \leq P_{update}, \\ 1 & if \quad P_{change} > P_{update} \end{cases} \tag{4.3-25}$$

The transition from the *Normal* state to the *Fail* state, P_{NF}, depends on the wireless channel characteristics and the network MAC protocol. We assume that the IEEE 802.11 standard is used. The node enters the *Fail* state when it initiates an unsuccessful clusterhead transfer due to collision, fading, interference, or hidden terminal phenomenon. This is then represented by the wireless channel probability of error, P_{error}.

$$P_{NF} = P_{error} \tag{4.3-26}$$

The transition probability from the *Fail* state to the *Normal* state, P_{FN}, is one as we assume that the cluster will be partitioned or recovered and then it will return to the *Normal* state. Finally, from Figure 4.6 we can obtain the probability that the cluster remains in the *Normal* state, P_{NN}. Since the probability that the PCH does not move out of its cluster or its SCH transmission range is (1 - P_L), the probability that the BPL of the PCH does not drop below the $THRESHOLD_2$ is (1 - P_{P-PCH}), and the probability that there is no error in the wireless channel is (1 - P_{error}), then P_{NN} is given by

$$P_{NN} = (1 - P_L)(1 - P_{P-PCH})(1 - P_{error}) \tag{4.3-27}$$

Now we can calculate the steady state probabilities of the *Normal*, *Transfer*, *Fail* states which are denoted as π_N, π_T and π_F, respectively. From Figure 4.6, we have

$$\pi_N P_{NN} + \pi_T P_{TN} + \pi_F = \pi_N \tag{4.3-28}$$

if $P_{change} \leq P_{update}$, then

$$\pi_N P_{NN} + \pi_T + \pi_F = \pi_N. \tag{4.3-29}$$

From the Markov properties

$$\pi_N + \pi_T + \pi_F = 1, \tag{4.3-30}$$

then

$$\pi_N = \frac{1}{2 - P_{NN}}, \tag{4.3-31}$$

and

$$\pi_T = \pi_N P_{NT} = \frac{P_{NT}}{2 - P_{NN}}. \tag{4.3-32}$$

On the other hand, if $P_{change} > P_{update}$, then

$$\pi_N P_{NN} + \pi_F = \pi_N, \tag{4.3-33}$$

and

$$\pi_N P_{NF} + \pi_T = \pi_F. \tag{4.3-34}$$

From the Markov properties

$$\pi_T = 1 - \pi_N - \pi_F, \tag{4.3-35}$$

from (4.3-34) and (4.3-35)

$$2\pi_F = 1 + \pi_N P_{NF} - \pi_N; \tag{4.3-36}$$

from (4.3-33) and (4.3-36)

$$\pi_N = \frac{1}{3 - 2P_{NN} - P_{NF}} \tag{4.3-37}$$

and

$$\pi_T = \pi_N P_{NT} = \frac{P_{NT}}{3 - 2P_{NN} - P_{NF}} \tag{4.3-38}$$

Hence, the stationary probabilities for the *Normal* state and the *Transfer* state can be rewritten as

$$\pi_N = \begin{cases} \frac{1}{2-P_{NN}} & if \quad P_{change} \leq P_{update} \\ \frac{1}{3-2P_{NN}-P_{NF}} & if \quad P_{change} > P_{update} \end{cases} \tag{4.3-39}$$

and

$$\pi_T = \begin{cases} \frac{P_{NT}}{2-P_{NN}} & if \quad P_{change} \leq P_{update} \\ \frac{P_{NT}}{3-2P_{NN}-P_{NF}} & if \quad P_{change} > P_{update} \end{cases} \tag{4.3-40}$$

The stationary probability for the *Fail* state can simply be obtained by using the normalizing probability condition, so

$$\pi_F = 1 - \pi_N - \pi_T \tag{4.3-41}$$

Based on the above discussions, to have a successful clusterhead transfer, an updated SCH is needed. The maximum update SCH selection interval is shown in Section 4.3.3.4. Once the SCH is still qualified, the clusterhead transfer becomes smooth. Also, since the successful transfer is done in one *Hello* message and the SCH is known by its cluster members, the cluster is still considered stable. On the contrary, when there is no qualified SCH, like the other clustering protocols,

the cluster takes longer time to be reformed. Therefore, the time spent in the re-clustering in the presence of the SCH is less than that when there is no SCH.

The time spent in the *Transfer* and the *Fail* states, T_T and T_F, depends on the length of the beacon interval τ or arrival *Hello* messaging rate (λ_h). λ_h has an impact on the hosts sensitivity to environmental changes and power consumptions. Increasing this rate will increase the update speed which reduces the transition time of the clusterhead in the *Transfer* state and reduces the re-clustering time in the *Fail* state.

On the contrary, the time spent in the *Normal* state, T_N, depends on the mobility and the power consumption rates of the PCH. No doubt, with no mobility and with small power consumption rate, the PCH has longer time to be a clusterhead.

The goals then are to improve the steady state of the *Normal* state and reduce the time spent in the *Transfer* state. In the SERC protocol, these two goals enhance the communication efficiency and the packet delivery time.

4.3.5 Numerical Analysis

The cluster model stated above is mainly driven by two main probabilities, P_{NT} and P_{update}. P_{NT} represents the need to have a new clusterhead, while P_{update} represents if the new clusterhead is already assigned and available. To demonstrate their impacts on the π_N, π_T and π_F, Table 4.1 and Table 4.2 are provided. P_{NT} is calculated by (4.3-23) based on different P_L and P_{P-PCH} gotten by different nodal speed and traffic load handled by the PCH. Having different traffic loads at the PCH can be either by varying the traffic generated by each node or by varying the number of nodes in the network. We assume that the channel probability of error, P_{error}, is 10^{-2}.

From the results shown in Table 4.1 and Table 4.2, the stationary probability for the *Fail* state, which is the undesired state, is reduced by 3 to 23 times in the

P_{NT}	P_{NN}	π_N	π_T	π_F
.02	.9702	.9711	.0194	.0095
.08	.9108	.9181	.0734	.0084
.14	.8514	.8706	.1219	.0075
.2	.792	.828	.1656	.0066

Table 4.1: The steady state probabilities with respect to different values of P_{NT} when there is updated SCH

P_{NT}	P_{NN}	π_N	π_T	π_F
.02	.9702	.9527	.0191	.0282
.08	.9108	.8559	.0684	.0757
.14	.8514	.7769	.1088	.1143
.2	.792	.707	.141	.152

Table 4.2: The steady state probabilities with respect to different values of P_{NT} when there is no updated SCH

presence of the SCH with different values of P_{NT}. This reduction enhances the stationary probability of the *Normal* state. This is an important result because it indicates that the cluster members with known future clusterhead are spending more time in sending or receiving packets instead of just building or reforming clusters. The stationary probability for the *Transfer* state is almost similar in both the presence and the absence of the SCH. However, the time spent in this case is less when the SCH is available. Also, this saved time is spent in sending or receiving packets. The results of Table 4.1 and Table 4.2 are shown in Figure 4.10. These results will be reflected on the communication efficiency and the average packet delivery time, which are provided by the computer simulation in the next chapter. As stated earlier, having different values of P_{NT} can be obtained by varying the nodal speed as well as varying the number of nodes in the network.

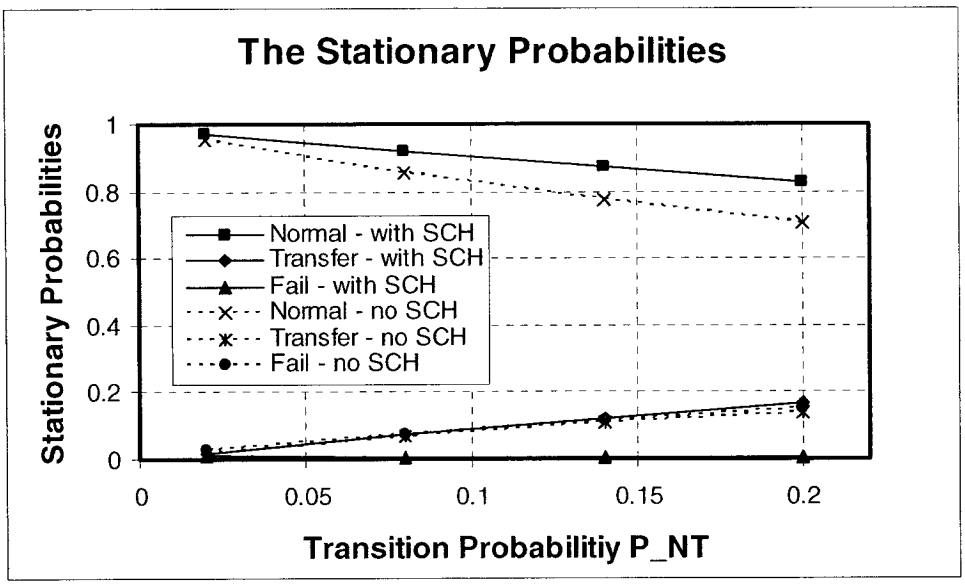


Figure 4.10: The stationary probabilities for *Normal*, *Transfer*, and *Fail* states with and without SCH vs. different values of P_{NT}

4.4 Summary

In this chapter, we have provided the network model of the SERC protocol and the conditions of the SCH to be elected among the cluster members. We have also provided an analysis for the clustering overhead of both the SERC protocol and the WEAC protocol. We have shown the parameters that have major impact on those protocols and we have discussed how they are affected by those parameters. It is important to emphasize that when the SCH is closer to its PCH, the probability that the transmission range of the SCH covers most of the cluster members and then the re-clustering overhead is improved. Moreover, in this report, the cluster residence time which is an important stability metric has been discussed and analyzed. A list of the SERC protocol properties has been illustrated and proved.

Since the main contribution of the SERC protocol is the transfer of the clusterhead from a primary to a secondary status, we have provided a clusterhead transfer modeling by using the Markov Chain Model. The states that the cluster

might be in at a certain time, the parameters that have impact on the transition from a state to another, and the conditions to have a successful transfer have been shown and discussed. The topology update speed, λ_h, and the SCH selection update interval (P_{update}) have been found to be the major important parameters to have a successful clusterhead transfer in the SERC protocol.

In the case when the SERC protocol has no SCH at all, that means it is similar to the other clustering protocols. The presence of the SCH enhances the cluster stability represented by the *Normal* state and reduces the re-clustering overhead that is needed in the *Transfer* state. Therefore, SERC with its SCH outperforms the others in the communication efficiency and the packet delivery time.

Chapter 5

SIMULATION AND PERFORMANCE EVALUATION

5.1 Introduction

In this chapter, we will present the performance evaluation of the SERC and the LC3R protocols. We carry out a comparison between our protocols and the WEAC protocol. The WEAC protocol has been chosen because of its positive features regarding load balancing, energy saving, maximizing throughput, and reducing delay as compared to some well recognized protocols like AODV, CGSR, VBS, and PA-VBS [SHE03]. In Section 5.2, the simulation models are described and the simulation parameters and the performance metrics are given. Section 5.3 includes the results of the conducted simulation experiments. The simulation results show that the SERC and the LC3R protocols outperform the WEAC protocol in overall performance.

5.2 Simulation Description

A simulation model is developed using discrete event simulation to evaluate the inherent stability, reliability, and efficiency of the SERC and the LC3R protocols. This simulator is written in Java programming language. The simulation results and the comparisons provided in this chapter are based on simulation models and parameters described in the following sections.

5.2.1 Mobility Model

The mobility models used in simulations can be roughly divided into two categories: independent and group-based. In the independent models, the movement of each mobile terminal is modelled independently of any other terminals in the simulation. In the group mobility models, there is some relationship among the nodes and their movements throughout the cells or the field. In this work, we assume that each mobile terminal moves in a way independent of any other terminals in the network.

The simplest independent mobility model used is known as the Random Walk. The Random Walk Mobility Model was first described mathematically by Einstein in 1926 [SAN01]. Since many entities in nature move in extremely unpredictable ways, the Random Walk Mobility Model was developed to mimic this unexpected movement. In this mobility model, a mobile terminal moves from its current location to a new location by randomly choosing a direction and speed in which to travel. The new speed and direction are both chosen uniformly from pre-defined ranges $[v_{min}, v_{max}]$ and $[0, 2\pi]$, respectively. Each movement in the Random Walk Mobility Model occurs in either a constant time interval t or a constant distance travelled d, at the end of which a new direction and speed are calculated. If a mobile terminal which moves according to this model reaches a simulation boundary, it bounces off the simulation border with an angle determined by the incoming direction. The mobile terminal then continues along this new path.

5.2.2 Traffic Model

Data communication among the mobile terminals is modelled as simultaneous network sessions. Constant Bit Rate (CBR) data sessions with randomly selected sources and destinations are used in our simulation. Each of the network mobile terminals send a CBR flow to another node chosen randomly, and each source

transmits data packets at a rate of 4 packets/sec. The size of each packet is 1024 bytes and the bit rate is 2.0 Mbps. In this study, we assume ideally shared channel media access. Thus, a session is blocked at the session initiation time if either the source or the destination node has no available battery energy, or if there is no route between the source-destination pair. After a session has begun, it is maintained as long as there is a route between the source and the destination. Since the number of data sessions is equal to the number of terminals in the network, we vary the traffic load by changing the number of mobile terminals to examine its effect on the packet delivery time and the communication efficiency.

5.2.3 Power Consumption Model

Since the proposed protocols in this study depend on the mobile terminals remaining energy level, we need to construct a model for ad hoc node energy consumption. Typically, the various tasks performed by a MAC protocol correspond to different radio modes, which exhibit different power requirements. In the particular case of the IEEE 802.11 DCF, two power management mechanisms are supported: *active* and *power-saving*. The *active* mechanism, a mobile terminal may be in one of three different radio modes, namely, *transmit*, *receive*, and *idle* modes. While it is active, the network radio transceiver consumes an average power of P_{active}, based on the proportion of the time that it spends in one of the three active modes. In the *transmit* mode, a mobile terminal is transmitting data at power $P_{transmit}$. Likewise, in the *receive* mode, the mobile terminal, while receiving a data packet, consumes power $P_{receive}$. Finally, in the *idle* mode, the receiver consumes power P_{idle}, while sensing the channel for the arrival of new data. In order to conserve power, the transceiver may transition into *sleep* mode, consuming reduced power level of P_{sleep}.

We assume that each terminal is equipped with an IEEE 802.11 based WaveLAN Network Interface Card (NIC) with a bandwidth of 2 Mbps and and a nominal

State	Power
$P_{transmit}$	1.3272 J/sec
$P_{recieve}$	0.96696 J/sec
P_{idle}	0.84372 J/sec
P_{sleep}	0.06636 J/sec

Table 5.1: Power Consumption in Various States

transmission range of 250m. Our energy consumption model is based on Feeney and Nilssons measurements of an IEEE 802.11b Lucent WaveLAN wireless network interface operating in an ad hoc networking environment [FEE01]. Their measurements are summarized in Table 5.1, where other measurements in the literature evaluating other 802.11b vendor equipments show the similar costs. When a mobile terminal sends or receives a packet, its available energy is decremented according to the specific NIC characteristics, the size of the packets, and the used bandwidth. The following equations represent the energy used (in Joules) when a packet is transmitted ($E_{transmit}$) or received ($E_{receive}$); packet size is represented in bits:

$$E_{transmit} = \frac{P_{transmit} * PacketSize}{2 * 10^6} \tag{5.2-1}$$

$$E_{receive} = \frac{P_{receive} * PacketSize}{2 * 10^6} \tag{5.2-2}$$

As stated in Table 5.1, a mobile terminal consumes energy not only when sending and receiving but also while listening and sleeping. In the *idle* and the *sleep* states, the consumed energy is based on the time spent in each state which is given by t_{idle} and t_{sleep}, respectively. Then

$$E_{idle} = P_{idle} * t_{idle} \tag{5.2-3}$$

and

$$E_{sleep} = P_{sleep} * t_{sleep} \tag{5.2-4}$$

Based on the results stated in Table 5.1, we can make some valuable observations about the power consumption in each mode. As one might expect, the transceiver consumes more power while receiving. However, receiving does not consume considerably more power than listening or sensing the channel. Measurements from a WaveLAN PC card demonstrate that the transceivers *idle* power consumption is only slightly less than the power consumption while receiving. This is an important result because it indicates that the SCH will not consume more power while receiving the packets that are sent to its PCH.

5.2.4 Simulation Parameters

The simulation experiments are conducted for the following parameters. The simulated environment is $1000m$ x$1000m$ rectangular flat area. We assume that every mobile node is aware of its location which can be obtained from GPS or some other location service systems. We also assume that all mobile nodes have the same transmission range. The wireless transmission range of MTs is set to $250m$. Initially, each mobile node is assigned a unique node ID, a random x-y position, a random mobility speed, and a random power level greater than 90% of its maximum battery power. We have set the values of $THRESHOLD_1$, $THRESHOLD_2$ and $THRESHOLD_3$ to 75%, 50% and 25% of the maximum battery power respectively. The node moves based on the random walk mobility model described above with random speed uniformly distributed between 10 and 20 *m/sec.* The Hello messages are broadcasted every 1 second. Each simulation scenario is run for enough time to reach and collect the desired data at the steady state. Each simulation scenario was run at least six times and the 95% confidence interval was found to be so small (in order of 0.001) to be shown on the different performance charts. The simulation experiments are conducted for MANET with 50, 75, 100, 125 and 150 mobile nodes. The simulation parameters are summarized in Table 5.2.

Parameters	Values
Mobility Model	Random Walk
Traffic Model	CBR
MAC Layer	IEEE 802.11b
Area Size	$1000m$ x $1000m$
Node Placement	Random
Network Size	50, 75, 100, 125, and 150 *nodes*
Packet size	1024 *bytes*
Packet Rate	4 *packets/sec*
Bandwidth	2 *Mbps*
Max. speed	20 *m/sec*
Min. Speed	10 *m/sec*
Transmission Range	$250m$

Table 5.2: Simulation Parameters

5.2.5 Performance Metrics

the following are the metrics we used to evaluate the performance of our protocols. In our results, we represent each metric as a value averaged over the number of nodes involved and over the number of simulation runs conducted.

- **Average Cluster Residence Time:** The time a node remains associated with a given cluster. This is measured as the duration from the merge time until the disjoint time for such a node. This metric will be used to assess the stability of the cluster topology.

- **Average Cluster Merge Operations Latency:** This metric measures the average time spent by a node to join its clusters. This time is the total between the merge-request issuing time and the merge-accept receiving time.

- **Clustering Overhead:** This metric measures the average number of cluster formation and maintenance messages (*Merge request, Merge accept, and*

Disjoint messages) that generated by each node to join or build a cluster. This is an important metric because it shows how efficient the scheme is in reducing the clustering communication overhead.

- **Normalized Routing Overhead:** The ratio of the amount, in bytes, of update packets of routing and cluster member tables transmitted to the amount, in bytes, of data packets received. The byte is used because of the fluctuation in size of those update packets. This metric is very important because it measures the efficiency of the protocol in consuming network resources (e.g., bandwidth and battery power).

- **Average Route Lifetime:** The route lifetime is measured as the time duration from route establishment between a pair of end nodes until the route breaks and the route recovery is invoked.

- **Average Packet Delivery Time:** This metric is measured from the moment when the packet is injected into the network until it is delivered to its destination.

- **Communication Efficiency:** This metric is measured by the packet delivery fraction. The packet delivery fraction is defined as the ratio of the number of data packets received correctly by the destination to the number of data packets initiated by the source. This metric is important because it affects the throughput which the network can support.

5.2.6 Factors

In our study, we considered the following factors:

- **No. of Nodes**: We evaluated our protocol for different number of nodes in the network. Doing this allowed us to assess the scalability of the protocol in larger networks.

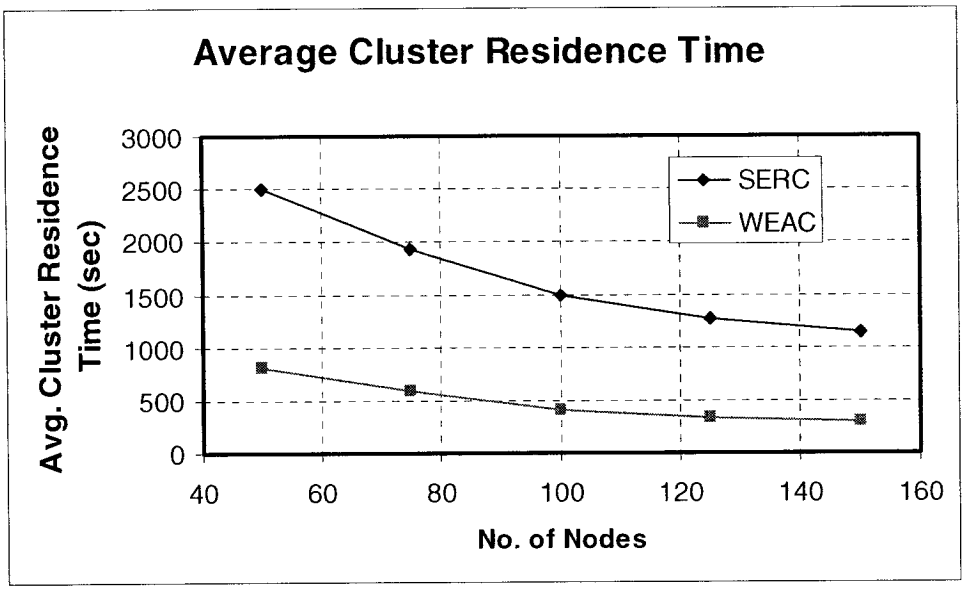


Figure 5.1: Average Cluster Residence Time vs. Number of Nodes

- **Cluster Density**:We evaluated our protocol for different cluster densities, i.e., average number of nodes per cluster. We achieved this by keeping the simulation area constant and varying the transmission range of the node for the same number of nodes in the network.

5.3 Simulation Results

In this section, the simulation results will be provided. First, the SERC protocol will be evaluated to show its features in the clustering process. Then, the LC3R protocol integrated with the SERC protocol will be evaluated to present how it can improve the overall performance.

5.3.1 SERC Performance Evaluation

Fig. 5.1 shows that the average cluster residence time in case of the SERC protocol is larger than the average time in the case of the WEAC protocol. Therefore, we can conclude that the cluster topology in the SERC protocol is more stable. This is because the cluster in the WEAC protocol is associated only with

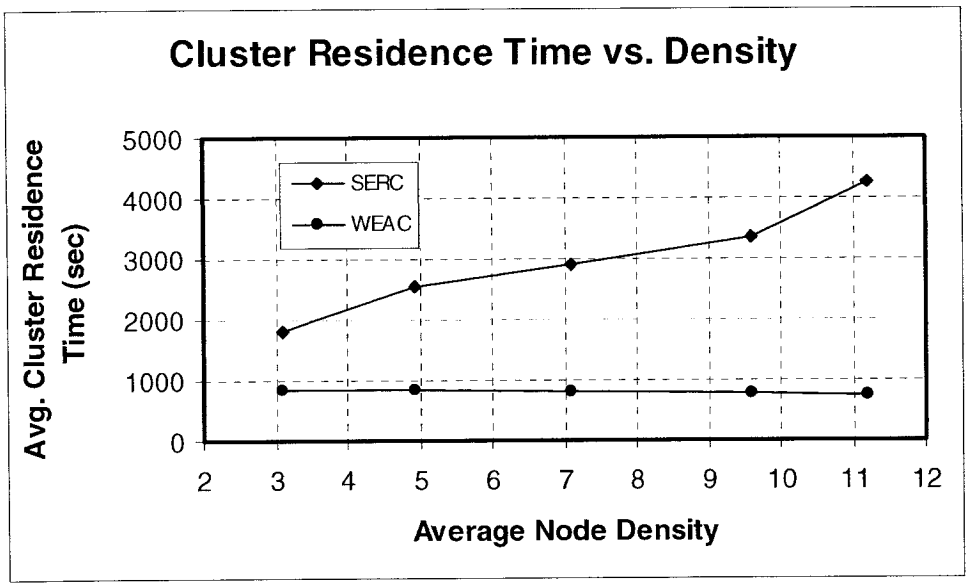


Figure 5.2: Average Cluster Residence Time vs. Average Node Density in the Cluster

the clusterhead. Therefore, in the absence of the clusterhead for any reason, the cluster will collapse and a new cluster would be reformed. However, in the SERC protocol, the SCH will take the clusterhead responsibility when the PCH can no longer be a clusterhead. The cluster leadership will be transferred, at that time, from the PCH to the SCH. The cluster members that were being heard by their SCH will continue their cluster residency with the SCH when it becomes a PCH. Hence the cluster will still survive and then the cluster residence time of the member will be extended. Fig. 5.1 shows also that with the increase of the number of nodes from 50 to 150, this improvement is increased gradually from three to four times respectively. This is because, in the network with a large population, the node density of the cluster is increased. Therefore, the large network has more stability using the SERC protocol than using the WEAC protocol. This result shows that the SERC protocol attains better scalability in addition to its features.

As a result of the analysis in Section 4.2.2 above, the increase of the node density increases the cluster residence time of the nodes with the SERC protocol.

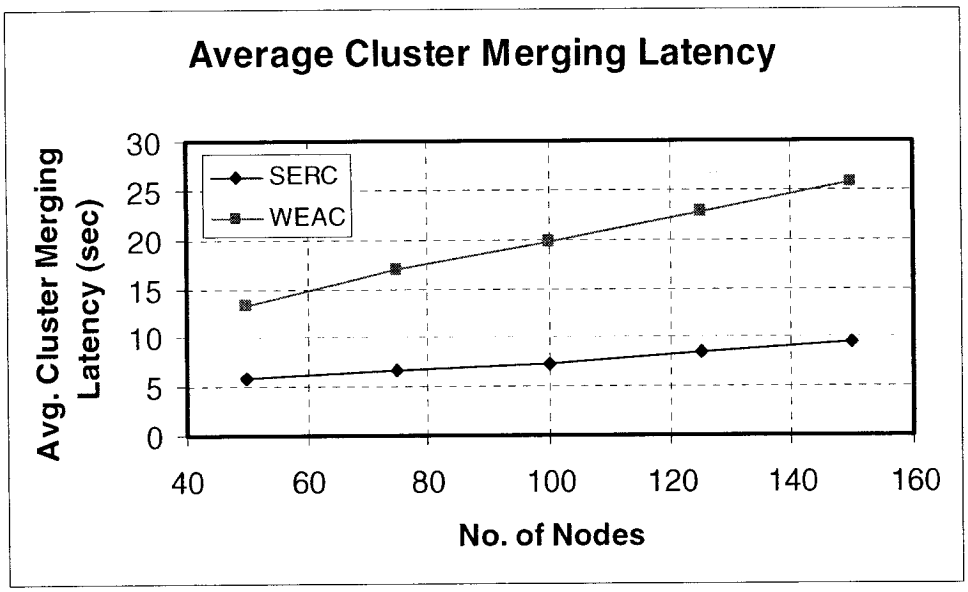


Figure 5.3: Average Cluster Merging Latency vs. Number of Nodes

This result is shown by the simulation in Fig. 5.2. It is obtained by having the same number of nodes and varying the transmission range to have different cluster densities. Increasing the node density increases the number of qualified SCH which increases first the cluster life time and then the cluster residence time.

Figs. 5.3 and 5.4 show the average time that the node spent to complete the merge operations with another node or with a clusterhead with a different number of nodes and different node densities, respectively. These results show that the node with the SERC protocol spends less time than the node with the WEAC protocol. The cluster residence time has a major impact on this metric because longer residence time means shorter time is needed to merge with another cluster. The node only spends this time when there is no qualified SCH to accommodate it. Therefore, with a large number of nodes or node density, the average cluster merging latency is reduced . Hence, with the SERC protocol, the node will spend its time in sending or forwarding packets instead of just building or joining clusters.

In Fig. 5.5, in the case of the WEAC protocol, the average number of cluster formation and maintenance messages generated by each node is proportionally

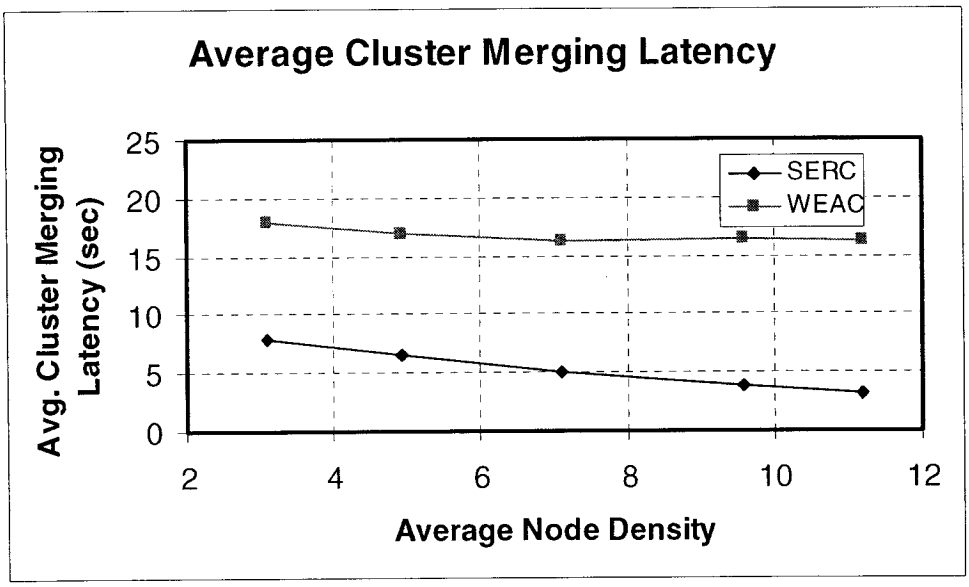


Figure 5.4: Average Cluster Merging Latency vs. Average Node Density in the Cluster

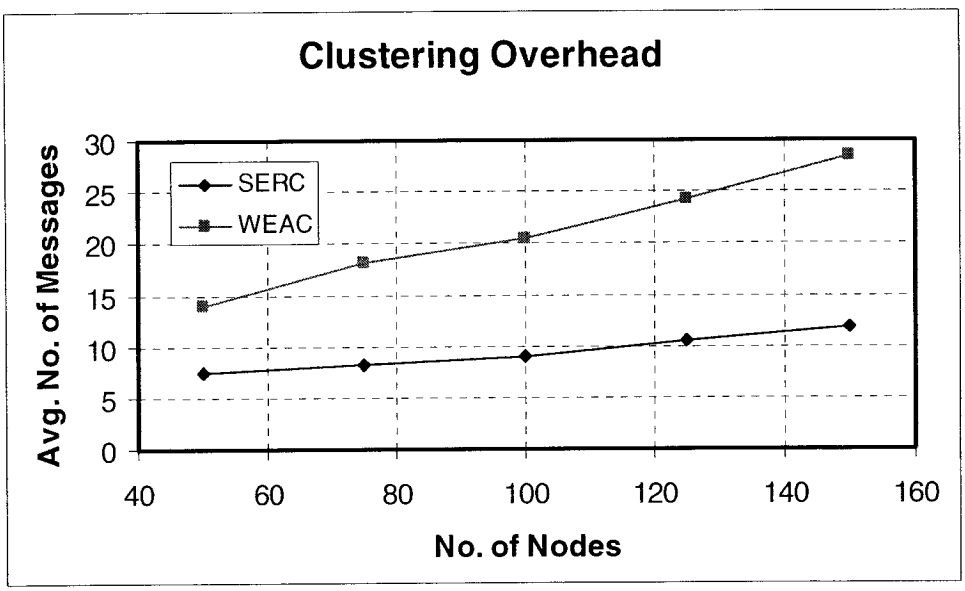


Figure 5.5: Average Number of Clustering Massages vs. Number of Nodes

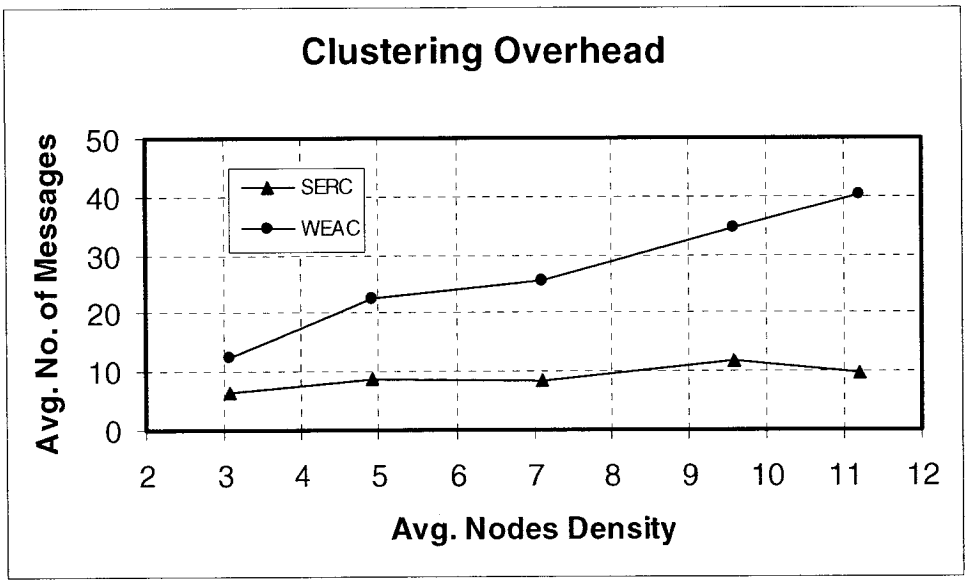


Figure 5.6: Average Number of Clustering Massages vs. Average Node Density in the Cluster

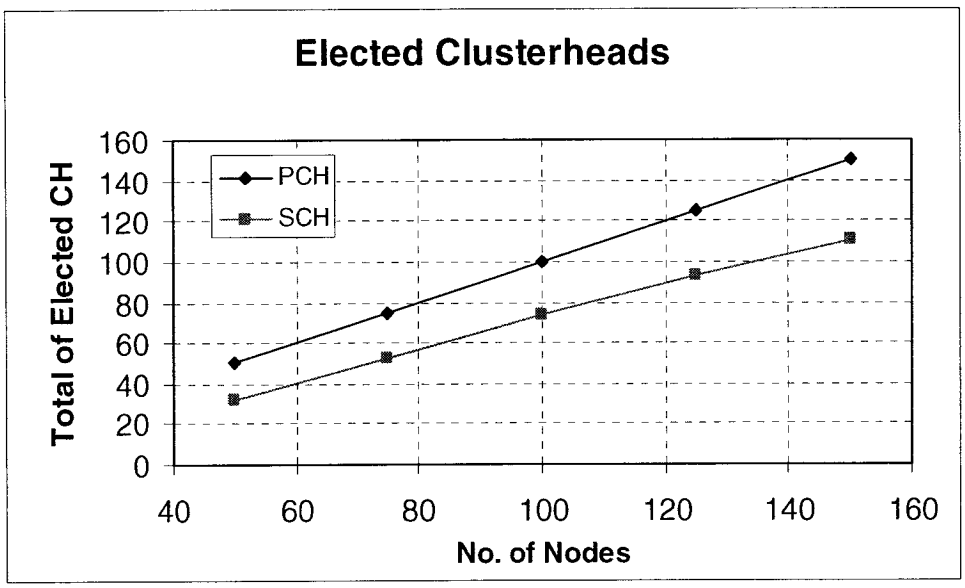


Figure 5.7: Total of Elected CHs (PCH & SCH) vs. Number of Nodes

increased with the increase of the number of nodes. This is obviously right because increasing the number of nodes will increase the number of created clusters and decrease the CH duration which are considered as a feature to balance the load (see [SHE03]). Therefore, the node will join more clusters during its lifetime, and then it needs to exchange more massages to do so. On the other hand, since the SERC protocol is mainly dependent on the *Hello* messages to reform the cluster, the average number of cluster formation and maintenance messages generated by each node is reduced. The clustering overhead in the case of the SERC protocol will drop by 180% to 240% as a result of increasing the number of nodes. This is because the higher node density increases the probability that the candidate SCH is closer to its PCH location and then its transmission range can cover most of the cluster member nodes. The impact of node density with a constant number of nodes is shown in Fig 5.6. As a result of the analysis above in Section 4.2.1, with the SERC protocol, the increase of the node density decreases the clustering communication overhead that is needed by each node to build or join another cluster. As we stated earlier, increasing node density increases the number of qualified SCH that use the *Hello* message to reform the cluster with no need to exchange the clustering messages. Therefore, with higher density, it is very rare for the cluster member to invoke the re-clustering mechanism that is used by the WEAC protocol to build or join a new cluster.

In Fig. 5.7, it is shown that each node can be elected as a clusterhead during the network lifetime. This is an important feature because it proves that the load is still balanced among all the nodes as it is proved by the WEAC protocol. Also, it is shown that increasing the number of nodes from 50 to 150, the ratio of the assigned SCHs that are triggered to be PCHs to the total elected PCHs is increased gradually from 64% to 77%, respectively. This also represents an increase in node density.

5.3.2 LC3R Performance Evaluation

A desirable routing protocol should offer a small routing overhead, long route lifetime, small average delivery time, and large packet delivery fraction.

Clustering in general is a well-known design technique that could reduce routing overhead by reducing the size of the routing tables. Each clusterhead periodically broadcasts its local topology changes to other clusterheads. Upon receiving an updated message from another clusterhead, a clusterhead updates the global topology. The process of detecting, collecting and distributing the topology changes produces a routing overhead. The role of the SCH in reducing the reclustering overhead has been provided in the SERC performance evaluation. In this section we will show how the SCH reduces the routing overhead, defined as the bit rate needed to maintain the global topology, which includes the overhead of distributing local topologies among the clusterheads.

In both of LC3R and WEAC protocols, each node broadcasts its cluster member table periodically using the well known Destination-Sequenced Distance-Vector (DSDV) algorithm. Also, their routing table updates are similar to those that are used by the DSDV protocol and can be sent in two ways: a "full dump" or an incremental update. A full dump sends the full routing table whereas an incremental update sends only those entries, from the routing table, that have a metric change since the last update. By design, the full dump will most likely need multiple network protocol data units (NPDUs), even for relatively small populations of mobile nodes. The incremental routing update should fit in one (NPDU). If there is space in the incremental update packet, then those entries whose sequence number has changed may be included. Periodically or immediately when the network topology changes are detected, clusterheads broadcast a routing table update packet. When the network is relatively stable, incremental updates are sent to avoid extra traffic and full dump are relatively infrequent. In a fast-changing network, incremental

packets can grow big so full dumps will be more frequent. Therefore, when the size of the incremental update approaches the size of the NPDU, the full dump can be scheduled so that the next incremental will be smaller.

Although the routing table is smaller in the WEAC protocol, the overhead of periodic broadcasting (full or incremental updates) for maintaining the routing table and the cluster member table is as heavy as in DSDV. However, in the LC3R protocol, when the cluster leadership transformation has been done successfully between the PCH and the SCH, an incremental update is used instead of a full update which is used by the WEAC protocol to update the routing table and the cluster member table. We argue that a reduction in the frequency of full update messages will reduce the protocol routing overhead. Therefore, the LC3R protocol usually uses full update less frequently than the WEAC Protocol. We considered more frequent full updates as a routing overhead for two reasons:

- If the network is flooded by the full routing table, this large packet will be fragmented into several NPDU, thus resulting in more MAC control packets and low network utilization.

- More routing computation time is needed by the node to build the full routing table, so reducing this time is necessary when it is possible. This can be done by using incremental update.

Thus, the LC3R protocol largely reduces the routing and cluster update traffic overhead. Therefore, the routing overhead in LC3R outperforms the overhead in WEAC as shown in Fig. 5.8. Also, this figure shows how the LC3R protocol improves routing scalability to large mobile ad hoc networks. With a large population of nodes in such networks, the routing overhead in general will be increased because of the increase in the size of the routing tables. Also, the characteristics of both the WEAC and LC3R protocols indicate that with a large number of nodes,

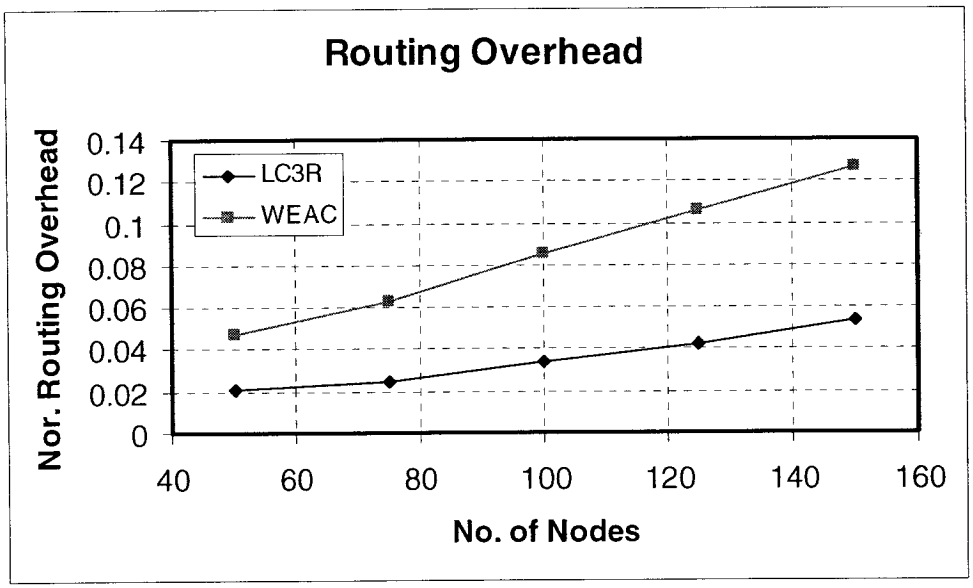


Figure 5.8: Normalized Routing Overhead vs. Number of Nodes

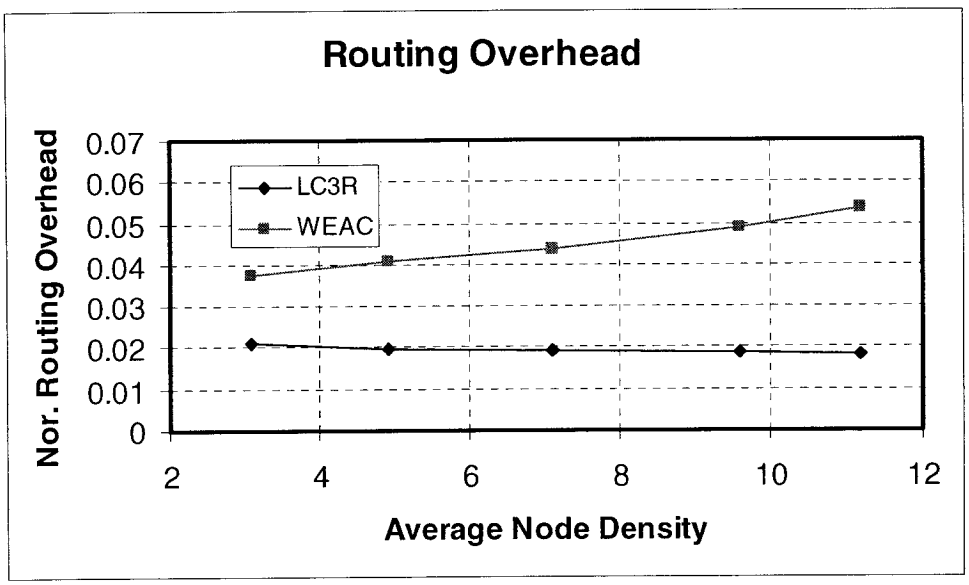


Figure 5.9: Normalized Routing Overhead vs. Average Node Density in the Cluster

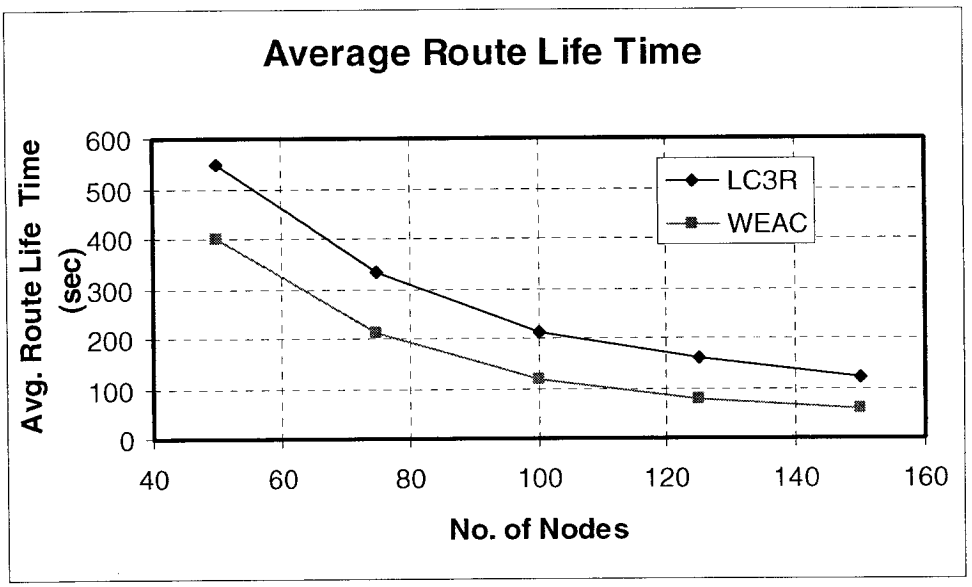


Figure 5.10: Average Route Life Time vs. Number of Nodes

the number of clusters is increasing and the clusterhead duration is decreasing. Therefore, the frequent clusterhead changes will be increased and consequently the routing overhead will be increased. However, in the case of LC3R, increasing the number of nodes increases the number of the candidate SCH, thus increasing the number of successful clusterhead transfers. As shown in Fig. 5.9, this is also applicable with the increase of the node density. Thus, in LC3R, the number of full updates is decreased and consequently the routing overhead has significantly improved.

The routing overhead will be reduced too if our algorithm is used over an on-demand cluster-based routing. The presence of SCH will reduce the number of route discoveries and the route reconfiguration time at route failure in the absence of any PCH. As it is already known, protocols that send less amounts of routing overhead decrease the probability of packet collisions, and data packets may have shorter delay in the network interface queues.

Sometimes, a node forwards its data packets to a route that is no more valid. This provides packet loss and then needs packet retransmissions once the node

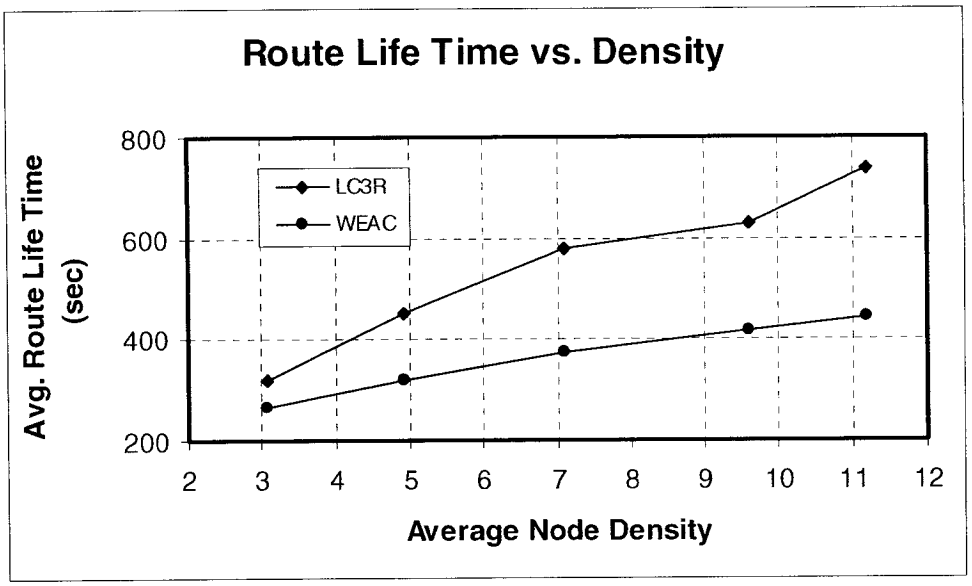


Figure 5.11: Average Route Life Time vs. Average Node Density in the Cluster

finds a valid route. Therefore, the long route lifetime means the less route interruptions which leads to a lower packet loss and a lower packet retransmission. In cluster-based routing protocols, the average route lifetime decreases with increasing the number of nodes (Fig. 5.10). This is obvious because the duration of the clusterhead is decreased by increasing the number of nodes [SHE03]. However, since the LC3R protocol can recover the route locally in the absence of the PCH, the average route lifetime is improved gradually with the increase of the number of nodes. This is because the higher the node density is the higher the probability that the candidate SCH is closer to its PCH location. Then the SCH transmission range can cover most of its PCH's range and recover more routes that are going through it. Therefore, in Fig. 5.10, with increasing the nodes from 50 to 150, the average route lifetime has been improved from 40% to 100%. This result is so important because it can help improve the communication efficiency and reduce the packet delay.

Figure 5.11 shows the effect of the node density in the cluster on the route lifetime. In this experiment, we set different node densities by varying the trans-

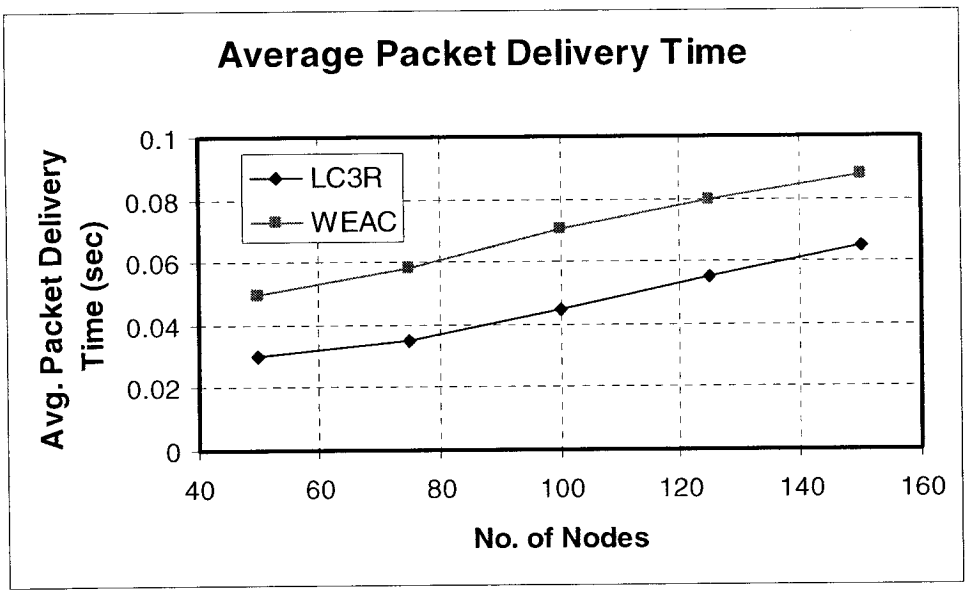


Figure 5.12: Average Packet Delivery Time

mission range and fixing the topology size with a constant number of nodes. In LC3R, with 50 nodes, when the average node density is increased from 3 to 11 in cluster, the route lifetime is improved from 20% to 60%, respectively.

Now the above illustrated features of the SERC/LC3R framework in stability and in saving of the communication overhead will be reflected on the main performance parameters: packet delivery time and communication efficiency. Fig. 5.12 shows that the LC3R protocol delivers the packets faster than the WEAC protocol. This is because the LC3R protocol has more reliable routes and more stable clusters. The route reliability provides less route interruptions which cause packet loss and then packet retransmissions. Also, the cluster stability will lead the node to be involved in packet transmissions instead of the clustering process and the cluster formation.

Moreover, Fig. 5.13 shows that LC3R performs better than the WEAC protocol in terms of the communication efficiency. Although the performance generally decreases with a greater number of nodes in the network, LC3R still outperforms the WEAC protocol. This is because the LC3R protocol has more reliable routes

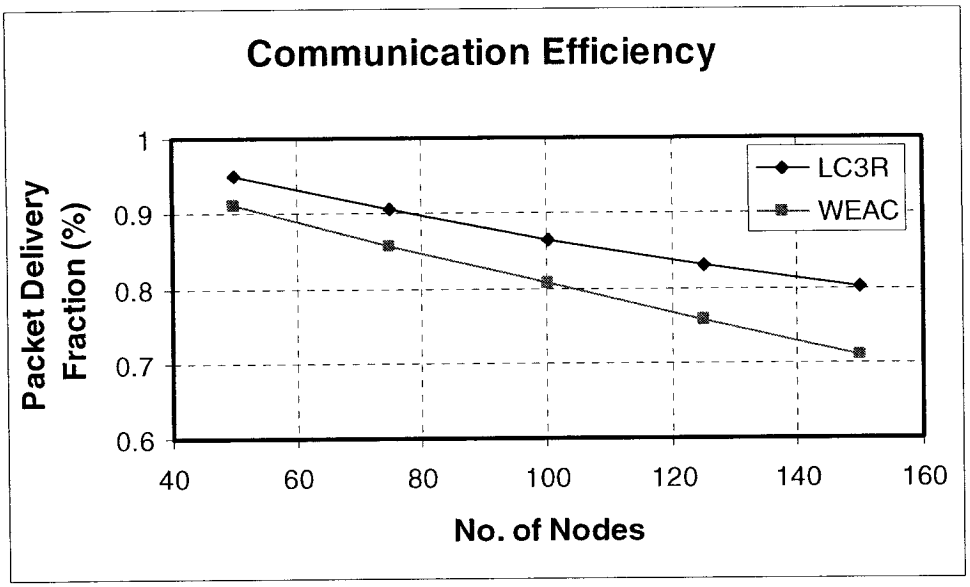


Figure 5.13: Packet Delivery Fraction

and more stable clusters. Route reliability provides less route interruptions that cause packet loss and then packet retransmissions. Also, cluster stability will lead the node to be involved in packet transmissions instead of the clustering process and cluster formation. It is necessary to keep in mind that more control packets and packet retransmissions increase the competition for network resources (e.g. bandwidth, medium access etc.) for all (control and data) transmissions.

5.4 Summary

This work is motivated by the lack of reliability and stability in multihop mobile ad hoc networks. The traditional schemes to provide reliability are based on the multipath forwarding that are based on packet level redundancy. However, these schemes waste the bandwidth of the already bandwidth-scarce environments. Therefore, the SERC/LC3R framework has been proposed to improve the stability and to enhance the reliability of the network with no waste of bandwidth and in reasonable time. This work has proposed algorithms and protocols to provide

solutions in three main areas: clustering, routing, and QoS. It is expected to improve their performance in the following ways:

- **Low Overhead in the Clustering Process:** Once the primary clusterhead (PCH) can no longer be a clusterhead, the secondary clusterhead (SCH) will reform the cluster very fast to save the overhead that is caused by the re-computation of the clusterheads and the frequent information that is exchanged between the participating nodes.

- **Very Fast Routing Process:** The delay caused by the rerouting and the resource reservations will be reduced in the presence of route failures because the route life time is increased.

- **More Reliable:** Using a virtual secondary path will increase the packet delivery ratio, and decrease the packet loss ratio. Consequently it will improve the communication efficiency and the network reliability with no waste of bandwidth.

Moreover, since the power consumption is an important issue in such resource scarce environments, the SERC/LC3R framework aims to reduce the power that might be consumed by: the re-clustering overhead, routing recovery overhead, and retransmitting of the lost packets.

Chapter 6

CONCLUSION AND FUTURE RESEARCH

6.1 Concluding Remarks

Since the cluster stability in MANETs affects the performance of other protocols such as scheduling, routing, and signaling, this dissertation has introduced a clustering algorithm for stable MANETs, namely the Smooth and Efficient Re-Clustering (SERC) protocol. The basic idea of this protocol is the election of a primary clusterhead (PCH) and a secondary clusterhead (SCH) for each cluster. The SCH works as a backup for the PCH and the future leader for the cluster. In SERC, the cluster member that was supervised by the PCH and was being heard by the SCH will stay associated with the cluster. This member will be associated with the SCH immediately when its PCH can no longer be a clusterhead. Once the SCH is triggered to be a PCH, a new SCH is assigned by the new PCH. The smooth clusterhead transfer, when needed, from the PCH to the SCH will give the cluster more lifetime and will save the re-clustering communication overhead.

In the SERC protocol, the SCH will reform the cluster very fast to save the overhead caused by the re-computation of the clusterheads and the frequent messages that are exchanged between the participating nodes. The simulation results show that, in the network with large population, the cluster residence time for each node is improved. This metric is so important because it has a direct correlation with the cluster stability as well as the algorithm scalability. Also the

average cluster merge operation latency and the number of messages needed to reform the cluster are reduced. Therefore, the saved time will be spent in sending or forwarding packets instead of just building or joining clusters. This reduction comes by deploying SCH for each PCH and using the *Hello* messages to transfer the leadership from PCH to SCH with no need for more clustering communication overhead. Moreover, in this study, analytical results show that deploying the SCH can provide significant gains relative to the clustering communication overhead and the cluster residence time.

Multipath routing is one way to improve the reliability of the transmitted information. Therefore, in this thesis, we introduce a new approach to multipath cluster-based routing protocol, which is named a Localized Cluster-based Rerouting and Resource Reservation Protocol (LC3R). LC3R utilizes the SERC infrastructure protocol with packet redundancy over its primary and secondary clusterhead (PCH and SCH) chains. The SCH works as a backup for the PCH and is the future leader for the cluster. The smooth transfer of the cluster leadership from the PCH to the SCH will localize the effect of the PCH failure. Since the SCH might fall into the transmission range of the upstream and downstream nodes of the routes that are going through its PCH, it can provide a new approach to the multipath routing protocol with no waste of bandwidth. The SCH could gather the reservation signaling messages and the data packets that were sent to its PCH. Simply, the main route can be established through the PCHs chain, while the pack-up route can be established through the SCHs chain. The route and its resource reservations would be recovered locally by the SCH once it becomes a PCH.

It is clear that the SERC/LC3R framework aims at reducing the control packets in the clustering process, as well as the routing and the resource reservation

process. These reductions have several positive impacts: decreased channel contention at the MAC layer, decreased reclustering and rerouting setup delay, and decreased timeouts and retransmissions at the MAC layer. In this study, we show how this approach can increase the route life time, reduce the routing overhead, decrease the packet delivery time, and achieve a high packet delivery rate with no more overhead. The delay caused by the rerouting and the resource reservations will be reduced in the presence of route failures because the route life time is increased. Using a virtual secondary path will increase the packet delivery ratio, and decrease the packet loss ratio. Consequently, it will improve the network reliability with no waste of bandwidth.

We present SERC/LC3R as a framework that runs on top of existing protocols, instead of replacing them. This framework is proposed to have efficient and effective utilization of any clustering algorithm and any cluster-based routing protocol running with it. Briefly, SERC/LC3R has two main features. First, it responds very fast to the failure of the PCH and then recovers rapidly the cluster as well as the data routes that are going through it. Second, the presence of the SCH reduces the communication overheads caused by the cluster reformations and the routing updates. These features allow this framework to outperform the other protocols in term of the communication efficiency and the packet delivery time. In this thesis, SERC/LC3R framework is described based on WEAC/VBS-O, but it can work on top of any existing ad hoc cluster-based routing protocols. SERC/LC3R adopt the basic concept of WEAC, but several modifications are made to take advantage of the secondary clusterhead. To the best of our knowledge, this is the first work that proposes two clusterheads for each cluster which is the main contribution of this work.

Analytical and simulation results show that the SERC/LC3R framework can provide significant gains relative to the clustering stability, routing reliability, and

QoS support. SERC/LC3R is a promising QoS management framework for mobile ad hoc networks. However, it is observed that its performance gains are limited by the cost that should be paid to elect and assign the SCH. First, the SCH election is based on the information availability of the location and the power level of each cluster member. Tracking this information may be considered as an overhead eventhough it is widely used in most of the clustering algorithms. Second, eventhough we allowed the SCH to go back and forth to and from the sleeping mode, with high mobility and high traffic, the SCH will be awake most of the time. Despite these limitations, we believe that SERC/LC3R is a good candidate framework to support QoS management in MANETs.

6.2 Future Research

Our research in the future falls into the following:

- **Interaction with other layer protocols:** The analysis provided in this dissertation was based on a simplified model of the network layer and the interaction with the other layers such as medium access control and transport layers was not taken deeply into account. Since the presence of the SCH might have some impacts on these layers in a way or another, the objective of our future work will be the development of a detailed analytical model which encompasses this interaction.

- **SCH in the wireless sensor networks:** Wireless Sensor Networks (WSNs) have recently emerged as an important computing platform. Sensor nodes are typically less mobile and more densely deployed than mobile ad-hoc networks (MANETs). As in MANETs, clustering in the wireless sensor networks is also a useful and important technique for extending the network lifetime. Applications requiring efficient data aggregation are natural candidates for

clustering. Routing protocols can also employ clustering. The clusterhead in the WSNs plays the same role in the MANETs. Since the WSNs have different physical characteristics, resource limitations, and clustering strategies, investigating the impact of deploying the secondary clusterhead should be done. Therefore, in our future work, we will also discuss how the deploying of the SCH in the WSNs is applicable and how the WSNs benefit from these new approaches presented in this thesis.

- **The integration of MANETs with the cellular networks:** Despite all the successes so far and the promising future of adhoc networks, Internet connectivity will still be very important and could quite likely be supported by cellular networks like GSM/GPRS or UMTS. Another possibility is to use ad-hoc networks as an extension of the cellular air-interface, leading to improvements in cellular capacity and coverage. In a cellular communication system, terminals that are far away from the base station need to use an overproportionally large amount of transmission power. Therefore, using an intermediate mobile in between, the transmission power for the cellular link can be reduced. Since the integration of MANETs with the cellular networks in an efficient manner is still an open issue, in the future we will study this interesting scenario to have seamless integration.

Bibliography

[ACE99] J.J. Garcia-Luna-Aceves and M. Spohn, "Source-Tree Routing in WirelessNetworks," *Proceedings of the IEEE International Conference on Network Protocols (ICNP)*, Toronto, Canada, October, 1999, pp. 273-282.

[AGA00] S. Agarwal, A. Ahuja, J.P. Singh, and R. Shorey, "Route-Lifetime Assessment Based Routing (RABR) Protocol for Mobile Ad-Hoc Networks," *Proceedings of the IEEE International Conference on Communications (ICC)*, New Orleans, LA, June 2000, pp. 1697-1701.

[AHN99] G-S Ahn, A. T. Campbell, S-B Lee, and X. Zhang, "INSIGNIA," *IETF Internet Draft*, ¡draft-ietf-manet-insignia-01.txt¿, Oct. 1999.

[ALK04] M. S. Al-kahtani, H. T. Mouftah, "An Efficient Reclustering Algorithm in Mobile Ad Hoc Networks," *Proceedings of 1st IEEE Iternational Computer Engineering Conference* (ICENCO-2004), Cairo, Egypt, Dec. 2004, pp. 230-235.

[ALK05] M. S. Al-Kahtani and H.T. Mouftah, "Enhancements for Clustering Stability in Mobile Ad Hoc Networks," *1st ACM Workshop on QoS and Security for Wireless and Mobile Networks*, Montreal, Canada, October, 2005.

[ALK05a] M. S. Al-Kahtani and H.T. Mouftah, "SERC/LC3R: A New Paradigm for Cluster-based Routing in MANETs,", *IEEE International Conference on Broadband Networks*, BroadNets'05, Boston, USA, October 2005.

[ALK05b] M. S. Al-Kahtani and H.T. Mouftah, "A Stable Clustering Formation Infrastructure Protocol in Mobile Ad Hoc Networks," *Proceedings IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob'2005)*, Montreal, Canada, Vol. 3, pp. 406-413, Aug. 2005.

[ALK05c] M. S. Al-Kahtani and H.T. Mouftah, "Localized Cluster-Based Rerouting and Resource Reservation in Mobile Ad Hoc Networks," *Advanced Industrial Conference on Wireless Technologies*, Montreal, Canada, pp. 13-18, August 2005.

[ALK05d] M. S. Al-Kahtani and H.T. Mouftah, "A New Approach to Cluster-based Multipath Routing Protocols in MANETs," *Proceedings International Symposium on Performance Evaluation of Computer and Telecommunication Systems, SPECTS'05*, Philadelphia, USA, pp. 433-446, 2005.

[AMI00] A. D. Amis, R. Prakash, T. H.P. Vuong and D. T. Huynh, "Max-Min D-cluster formation in wireless ad hoc networks," *Proceedings of IEEE INFOCOM'2000,* pp. 32-41.

[AND01] L. Andersson, P. Doolan, N. Feldman, A. Fredette and B. Thomas, "LDP Specification," *RFC 3036*, IETF, Jan. 2001.

[ARM00] G. Armitage, Quality of Service in IP Networks. *Pearson Higher Education; 1st edition*, Apr. 2000.

[AWD99] Awduche et al, "Requirements for Traffic Engineering over MPLS," *RFC2702*, September 1999

[AWD01] D. Awduche, L. Berger, D. Gan, T. Li, V. Srinivasan and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels," *RFC 3209*, Dec. 2001.

[AWD02] Awduche et al, "Overview and Principles of Internet Traffic Engineering," *RFC3272*, May 2002.

[BAD01] B. R. Badrinath, Arup Acharya, "MRSVP: A Reservation Protocol for an Integrated Services Packet Network with Mobile Hosts", *Wireless Networks* Vol. 7, pp. 5-19, 2001 Kluwer Academic Publishers.

[BAI03] H. Bai, H. Aerospace, M. Atiquzzaman, "Error Modeling Schemes for Fading Channels in Wireless Communications: A Survey," I*EEE Communications Surveys*, Fourth Quarter 2003, vol. 5, No. 2, pp.2-9.

[BAR81] D. J. Baker and A. Ephremides, "The architectural organization of a mobile radio network via a distributed algorithm, *IEEE Trans. on Comm.*, vol. COM-29, pp. 1694-1701, Nov. 1981.

[BAS99] S. Basagni, "Distributed Clustering Algorithm for Ad-hoc Networks," *In International Symposium on Parallel Architectures, Algorithms, and Networks (I-SPAN)*, pp. 310-315, 1999.

[BAS01] P. Basu, N. Khan, and D. C. Little, "Mobility Based Metric for Clustering in Mobile Ad Hoc Networks", Workshop on Distributed Computing Systems, pp. 413-418, 2001.

[BAS04] S. Basagni, M. Mastrogiovanni, and C. Petrioli, "A performance comparison of protocols for clustering and backbone formation in large scale ad hoc networks," *IEEE International Conference on Mobile Ad-hoc and Sensor Systems,* pp. 70-79, 2004.

[BER00] Y. Bernet, P. Ford, R. Yavatkar, F. Baker, L. Zhang, M. Speer, R. Braden, B. Davie, J. Wroclawski, E. Felstaine, "A Framework for Integrated Services Operation over DiffServ Networks," *RFC 2998*, Nov. 2000.

[BET04] C. Bettstetter, "The cluster density of a distributed clustering algorithm in ad hoc networks," *IEEE International Conference on Communications,* Vol. 7, pp. 4336-4340, 2004.

[BLA99] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang and W. Weiss, "An Architecture for Differentiated Services," *RFC 2475*, IETF, Dec. 1999.

[BOR98] J. Broch, D. Maltz, D. Johnson, Y.-C. Hu, and J. Jetcheva, "A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols," In *Proceedings of the IEEE/ACM MOBICOM*, pp. 85-97, 1998.

[BRA94] R. Braden, D. Clark and S. Shenker, "Integrated Services in the Internet Architecture: an Overview," *RFC 1633*, IETF, June 1994.

[BRA97] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin, "Resource ReSerVation Protocol (RSVP)," *RFC 2205*, IETF, Sep. 1997.

[CHA99] D. Chalmers M. Sloman, "A survey of QoS in mobile computing environments," *IEEE Communications Surveys & Tutorials*, Vol. 2, No. 2, 1999, pp. 2-10.

[CHA02] M. Chatterjee, S. K. Das, and D. Turgut, "WCA: A Weighted Clustering Algorithm for Mobile Ad Hoc Networks," *Journal of Cluster Computing,* No. 5, 2002, pp. 193-204.

[CHE98] T.-W. Chen and M. Gerla, "Global State Routing: A New Routing Scheme for Ad-hoc Wireless Networks," *Proceedings of the IEEE International Conference on Communications (ICC)*, Atlanta, GA, June 1998, pp. 171-175.

[CHE99] S. Chen, and K. Nahrstedt, "Distributed Quality of Service Routing in Ad Hoc Networks," *IEEE JSAC*, vol. 17, no. 8, pp. 1488-1505, August 1999

[CHI97] C.-C. Chiang, H.-K. Wu, W. Liu, and M. Gerla, "Routing in Clustered Multihop, Mobile Wireless Networks with Fading Channel," *Proceedings of the IEEE Singapore International Conference on Networks (SICON)*, Singapore, pp. 197-211, April 1997.

[CLA00] T. Clausen, P. Jacquet, A. Laouiti, P. Muhlethaler, A. Qayyum et L. Viennot "Optimized Link State Routing Protocol," *Internet Draft*, draft-ietf-manet-olsr-03.txt, Nov. 2000.

[COR95] S. Corson and A. Ephremides "A distributed routing algorithm for mobile wireless networks", *ACM/Baltzer Journal of Wireless Networks*, February 1995, Vol. 1, No. 1, pp. 61-81.

[DAS00] S. R. Das, C.E. Perkins, and E.M. Royer, "Performance Comparison of Two On-Demand Routing Protocols for Ad Hoc Networks," *Proceedings of the IEEE INFOCOM'2000*, Tel Aviv, Israel, pp. 3-12, March 2000.

[DE02] S. De, S. Das, H. Wu and C. Qiao, "Trigger-Based Distributed QoS Routing in Mobile Ad Hoc Network," *Mobile Computing and Communications Review*, Vol 6, No 3, pp. 22-35, July 2002.

[DE03] S. De and C. Qiao, "Does Packet Replication along Mulitpath Really Help?," *IEEE International Conference on Communications 2003*, Alaska, USA, Vol. 26, No. 1, pp. 1069-1073, May 2003.

[DUB97] R. Dube, C.D. Rais, K.-Y. Wang, and S.K. Tripathi, "Signal Stability-Based Adaptive Routing (SSA) for Ad Hoc Mobile Networks", *IEEE Personal Communications Magazine*, vol. 4, no. 1, February 1997, pp. 36-39

[EVA02] R. Lloyd-Evans, QoS in Integrated 3G Networks. *Artech House; 1st edition,* July 2002.

[FEE01] L. M. Feeney and M. Nilsson, "Investigating the energy consumption of a wireless network interface in an ad hoc networking environment," *in Proceeding of IEEE INFOCOM*, 2001, pp. 1548-1557.

[GER95] M. Gerla and J. T.-C. Tsai, "Multicluster, mobile, multimedia radio network," *ACM/Baltzer Journal of Wireless Networks*, 1 (1995), pp. 255-265.

[GRA98] E. Crawely, R. Nair, B. Rajagopalan, and H. Sandrick, " A Framework for QoS Based Routing in the Internet," *RFC 2386*, August 1998.

[GRI94] G.R. Grimmett and D.R. Stirzaker, "Probability and Random Processes," 2nd ed., *Clarendon Press: Oxford,* 1994

[HAS01] H. Hassanein and A. Safwat, "Virtual Base Stations for Wireless Mobile Ad-hoc Communications: An Infrastructure for the Infrastructure-less," *The International Journal of Communications Systems*, Vol. 14, pp. 763-782, 2002.

[HU00] Y.-C. Hu and D. Johnson, "Caching strategies in On-demand Routing Protocols for Wireless Ad Hoc Networks," In *Proceedings of the IEEE/ACM MOBICOM*, pp. 231-242, 2000.

[ISA95] Richard Isaac, "The Pleasures of Probability," *Springer: New York,* 1995.

[JAM02] B. Jamoussi, Ed., "Constraint-Based LSP Setup Using LDP," *RFC 3212*, January 2002

[JIA99] M. Jiang, J. Li, and Y.C. Yay, "Cluster Based Routing Protocol (CBRP) Functional Specification," *Internet-Draft*, draft-ietf-manet-cbrp-spec-01.txt, August 1999.

[JOH96] D. B. Johnson, David A. Maltz, "Dynamic Source Routing in Ad Hoc Wireless Networks," *Mobile Computing*, Kluwer Academic Publishers, pp. 153-181, 1996.

[KON03] A. Konrad, B. Y. Zhao, A. D. Joseph, and R. Ludwig, " A Markov-Based Channel Model Algorithm for Wireless Networks," *Wireless Networks Journal*, vol. 9, pp. 189-199, 2003.

[LEE00a] B. Lee, G.S. Ahn,, X. Zhang, and A. T. Campbell, "INSIGNIA: An IP-Based Quality of Service Framework for Mobile Ad Hoc Networks", *Journal of Parallel and Distributed Computing, Special issue on Wireless and Mobile Computing and Communications,* Vol. 60 No. 4 pp. 374-406, April 2000

[LEE00b] S. J. Lee and M. Gerla, "AODV-BR: Backup Routing in Ad hoc Networks," In *Proceedings of the IEEE Wireless Communications and Networking Conference(WCNC)*, pp. 1311-1316, 2000.

[LEE01] S. J. Lee and M. Gerla, "Split Multipath Routing with Maximally Disjoint Paths in Ad Hoc Networks," *Proceedings of the IEEE ICC*, pp. 3201-3205, 2001.

[LI03] Ning Li, Jennifer C. Hou, and Lui Sha. Design and analysis of an MST-based topology control algorithm. *IEEE Transactions on Wireless Communications*, Vol. 4, No. 3, pp. 1195-1207, May 2005.

[LIN99] C.R. Lin and J.-S. Liu, "QoS Routing in Ad Hoc Wireless Networks," *IEEE Journal on Selected Areas in Communications, special issue on Wireless Ad Hoc Networks*, vol. 17, no. 8, August 1999, pp. 1426-1438.

[LIU02] J. Liu, Q Zhang, B. Li, W. Zhu, and J. Zhang, "A Unified Framework for Resource Discovery and QoS-Aware Provider Selection in Ad Hoc Networks,"

Mobile Computing and Communications Review, Vol. 6, No. 1, pp. 13-21, Jan. 2002.

[LLY02] M. Ilyas, The Handbook of Ad hoc Wireless Networks. *CRC Press;* Dec. 2002.

[MAL99] D.A. Maltz, J. Broch, J. Jetcheva, and D.B. Johnson, "The Effects of On-Demand Behavior in Routing Protocols for Multihop Wireless Ad Hoc Networks," *IEEE Journal on Selected Areas in Communications, special issue on Wireless Ad Hoc Networks*, vol. 17, no. 8, August 1999, pp. 1439-1453.

[MAR01] M. K. Marina and S. R. Das, "Performance of Route Caching Strategies in Dynamic Source Routing," In P*roceedings of the Intl Workshop on Wireless Networks and Mobile Computing (WNMC) in conjunction with Intl Conf. on Distributed Computing Systems (ICDCS)*, pp. 425-432, 2001.

[MAX93] N. F. Maxemchuk, "Dispersity Routing in High-speed Networks," *Computer Networks and ISDN system* vol. 25, no. 6, Jan. 1993, pp. 645-661.

[MCD99] A. B. McDonald and T. F. Znati, "A mobility-based framework for adaptive clustering in wireless ad hoc networks," *IEEE Journal on Selected Areas in Communications,* Vol. 17 No. 8 pp. 1466-1486, Aug. 1999.

[MCQ80] J. M. McQuillan, I. Richer, Rosen, Eric C., "The New Routing Algorithm for the ARPANET," *IEEE Transactions On Communications*, Vol. COM-28, No. 5, May 1980, pp. 711-719.

[MIR00] M. Mirhakkak, N. Schultz, and D. Thompson, "Dynamic QoS for Mobile Ad Hoc Networks," *MobiHoc'2000*, pp. 137-138, April 2000.

[MOY98] J. Moy, "OSPF Version 2," *Request For Comments 2328*, Internet Engineering Task Force, April 1998.

[MUR96] S. Murthy and J.J. Garcia-Luna-Aceves, "An Efficient Routing Protocol for Wireless Networks," *ACM/Baltzer Mobile Networks and Applications, special issue on Routing in Mobile Communications Networks*, vol. 1, no. 2, October 1996, pp. 183-197.

[NAS99] A. Nasipuri and S. R. Das, "On-demand multipath routing for mobile ad hoc networks," *Proceedings of IEEE ICCCN99*, pp.64-70. Oct. 1999.

[NAS01] A. Nasipuri, R. Castaneda and S. R. Das, "Performance of Multipath Routing for On-Demand Protocols in Ad Hoc Networks," *ACM/Kluwer Mobile Networks and Applications (MONET) Journal*, Vol. 6, No. 4, 2001, pp. 339-349.

[NIK02] N. Nikaein and C. Bonnet, "A Glance at Quality of Service models in Mobile Ad Hoc Networks", *In proceedings of DNAC 2002 : 16th conference of New Architectures for Communications*, France/Paris 2002.

[PAL04] R. Palit, E Hossain and P. Thulasiraman, "Mobility-aware pro-active low energy (MAPLE) clustering in ad hoc mobile wireless networks," *Proceedings IEEE Global Telecommunications Conference, GLOBECOM'04*, Vol. 6, pp. 3426-3430, 2004.

[PAR97] V.D. Park and M.S. Corson, "A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks," *Proceedings of the IEEE Conference on Computer Communications (INFOCOM)*, Kobe, Japan, April 1997, pp. 1405-1413.

[PAR01] V. Park, S. Corson, "The Temporally-Ordered Routing Protocol (TORA) Specification," *Internet Draft*, July 2001.

[PEI00a] G. Pei, M. Gerla, T.-W. Chen, "Fisheye State Routing: A Routing Scheme for Ad Hoc Wireless Networks," *Proceedings of the IEEE International Conference on Communications (ICC)*, New Orleans, LA, June 2000, pp.70-74.

[PEI00b] G. Pei, M. Gerla, and X. Hong, "LANMAR: Landmark Routing for Large Scale Wireless Ad Hoc Networks with Group Mobility," *Proceedings of the ACM/IEEE Workshop on Mobile Ad Hoc Networking and Computing (MO-BIHOC)*, Boston, MA, August 2000, pp. 11-18.

[PER94] C. E. Perkins, Pravin Bhagwat, "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers," *Proceedings of the SIGCOMM '94 Conference on Communications, Architectures, Protocols and Applications*, pp 234-244, August, 1994.

[PER99] C. E. Perkins and E. M. Royer, "Ad hoc On-Demand Distance Vector Routing, Ad hoc Networking," *Proceedings of the Second IEEE Workshop on Mobile Computing Systems and Applications (WMCSA '99)*, pp. 90-100, 1999.

[PER00a] c. Perkins et al., "Ad Hoc On-Demand Distance-Vector Routing (AODV)," *IETF draft*, 14 July 2000, http://www.ietf.org/internet-drafts/draftietf-manet-aodv-06.txt.

[PER00b] C. Perkins, E. Royer, and S. R. Das, "Quality of Service for Ad Hoc On-Demand Distance Vector (AODV) Routing," *Internet Draft*, July 2000.

[PER02] D. D. Perkins and H. D. Hughes, "A survey on QOS support for mobile ad hoc networks". *Wireless Communications and Mobile Computing*, 2. pp. 503-513, 2002.

[PRA04] N. Prabagarane, C. Ajoy navin, B. Partibane, V. Nagarajan and R. Krishnakiran, "Hierarchical routing algorithm for cluster based multihop mobile ad hoc network," *Wireless Communications and Networking Conference, WCNC'2004* IEEE Vol. 2, pp. 1116-1120, 2004.

[PUR93] M.B. Pursley and H.B. Russell, "Routing in Frequency-Hop Packet Radio Networks with Partial-Band Jamming," *IEEE Transactions on Communications*, vol. 41, no. 7, July 1993, pp. 1117-1124.

[RAJ99] J. Raju and J.J. Garcia-Luna-Aceves, "A New Approach to On-demand Loop-Free Multipath Routing," *Proceedings of the IEEE International Conference on Computer Communications and Networks (ICCCN)*, Boston, MA, October 1999, pp. 522-527.

[ROS01] E. Rosen, A. Viswanathan, and R. Callon, "Multi-Protocol Label Switching Architecture," *RFC 3031*, IETF, Jan. 2001.

[ROY99a] E. M. Royer and C.Toh., "A Review of Current Routing Protocols for Ad Hoc Mobile Wireless Networks," *IEEE Personal Communications*, pp. 46-55, Apr 1999.

[ROY99b] E.M. Royer and C.E. Perkins, "Multicast Operation of the Ad-hoc On-Demand Distance Vector Routing Protocol," *Proceedings of ACM/IEEE MOBICOM'99*, Seattle, WA, pp. 207-218, Aug. 1999.

[SAF01a] A. Safwat and H. Hassanein, "Structured routing in wireless mobile ad-hoc networks," *IEEE International Symposium on Computers and Communications*, pp. 332-337, Hammamat, Tunisia, July 2001.

[SAF01b] A. Safwat, H. S. Hassanein, and H. Mouftah, "Power-Aware Fair Infrastructure Formation for Wireless Mobile Ad Hoc Communications," *Proceedings of IEEE Globecom 2001*, San Antonio, Arizona, November 2001, pp. 2832-2836.

[SAN01] M. Sanchez, P. Manzoni, "ANEJOS: A Java based simulator for ad-hoc networks", *Future Generation Computer Systems*, Elsevier Science, Vol. 17, No. 6, 2001, pp. 573-583.

[SHE02a] T. R. Sheltami, and H. T. Mouftah, "Virtual Base Station On-demand," *The International Conference on Wireless Networks (ICWN'02)* June, 2002, Las Vegas, Nevada, USA, pp 421-425.

[SHE02b] T. Sheltami and H.T. Mouftah, "Performance Comparison of Three Clustering Protocols in Mobile Wireless Ad Hoc Networks", *Proceedings 21st Biennial Symposium on Communications*, Kingston, Ontario, June 2002, pp. 139-143.

[SHE03] T. Sheltami and H. Mouftah, "An Efficient Energy Aware Clusterhead Formation Infrastructure Protocol for MANETs," *Proceedings IEEE International Symposium on Computers and Communications (ISCC2003)*, Antalya, Turkey, July 2003, pp. 203-208.

[SHE04] C-C Shen, C. Srisathapornphat, R. Liu, Z. Huang, C. Jaikaeo, and E.L. Lloyd, "CLTC: a cluster-based topology control for ad hoc networks," *IEEE Transactions on Mobile Computing,* Vo. 3, No. 1, pp. 18-32, 2004.

[SIN99] P. Sinha, R. Sivakumar, V. Bharghavan, "CEDAR: a core extraction distributed ad hoc routing algorithm," *In IEEE Infocom'99*, New York, pp. 202-209, March 1999.

[SIV04] S. Sivavakeesar, S.; Pavlou, G.; Liotta, A, "Stable clustering through mobility prediction for large-scale multihop intelligent ad hoc networks," *Proceedings IEEE Wireless Communications and Networking Conference WCNC'2004,* Vol. 3 pp. 1488-1493, 2004.

[SIV04a] S. Sivavakeesar, G. Pavlou, C. Bohoris, and A. Liotta, "Effective management through prediction-based clustering approach in the next-generation ad hoc networks," *IEEE International Conference Communications,* Vol. 7, pp. 4326-4330, 2004.

[SU01] W. Su, S.J. Lee and M. Gerla, "Mobility prediction and routing in ad-hoc wireless networks," *International Journal of Network Management,* vol. 11, no. 1, 2001, pp. 3-30.

[SUC02] J. Sucec and I. Marsic, "Clustering Overehad for Hierarchical routing in Mobile Ad Hoc Networks," *Proceedings of IEEE INFOCOM'2002,* pp. 1698-1706, New York, NY, June 2002.

[TER99] A. Terzis, M. Srivastava and L. Zhang, "A Simple QoS Signaling Protocol for Mobile Hosts in Integrated Services Internet," *IEEE Infocom Proceedings,* Vol. 3, 1999.

[TOH01] C. K. Toh, Ad Hoc Mobile Wireless Networks: Protocols and Systems. *Prentice Hall; 1st edition,* Dec. 2001.

[TSE01] C. Tseng, G. Lee, R. Liu, "HMRSVP: A Hierarchical Mobile RSVP Protocol," *Wireless Networks* Vol. 9 , No. 2, pp. 95-102, 2003.

[TSE03] Y. Tseng, C-S Hsu, T-Y Hsieh, "Power-Saving Protocols for IEEE 802.11-Based Multi-Hop Ad Hoc Networks," *Computer Networks: The International Journal of Computer and Telecommunications Networking,* vol 43, Issue 3 (October 2003), pp. 317-337

[XIA00] H. Xiao, W. K. G. Seah, A. Lo, and K. C. Chua, "A Flexible Quality of Service Model for Mobile Ad-Hoc Networks," *in IEEE VTC2000-spring*, Tokyo, Japan, Vol. 1, pp. 445-449, May 2000.

[YE03] Z. Ye, S. V. Krishnamurthy, S. K. Tripathi, "A Framework for Reliable Routing in Mobile Ad Hoc Networks," *Proceedings of IEEE INFOCOM'2003*, vol. 22, no. 1, Mar 2003 pp. 270-280 .

[YOU04] O. Younis and S. Fahmy, "Distributed Clustering in Ad-hoc Sensor Networks: A Hybrid, Energy-Efficient Approach," *Proceedings of IEEE INFOCOM'2004*, Hong Kong, March 7-11, 2004.

[WAN01] Z. Wang, Internet QoS: Architectures and Mechanisms for Quality of Service. *Morgan Kaufmann; 1st edition*, Mar. 2001.

[WIS02] D. Wisely, P. Eardly, and Louise Burnees, IP for 3G: Networking Technologies for Mobile Communications. West Sussex, PO, England, *John Wiley Son Ltd*, 2002.

Appendix A

MANETS ROUTING PROTOCOLS

In MANETs, an intelligent routing strategy is required to efficiently use the limited resources while at the same time being adaptable to the changing network conditions. Prior to the increased interests in wireless networking, in wired networks two main algorithms were used. These algorithms are commonly referred to as the link-state and distance vector algorithms. The traditional link-state and distance-vector algorithm do not scale in large MANETs. This is because periodic or frequent route updates in large networks may consume significant part of the available bandwidth, increase channel contention and may require each node to frequently recharge their power supply. To overcome the problems associated with the link-state and distance-vector algorithms, a number of routing protocols have been proposed for MANETs. In Chapter 2, we have provided a review for the routing protocols of the MANETs. These protocols have been classified in four categories according to the way the routes are created and maintained. Based on this classification, in this Appendix, we will provide the most prominent protocols for each class.

A.1 Table-Driven Routing Protocols

The most prominent protocols in the proactive class are the DSDV [PER94] and the Wireless Routing Protocol (WRP) [MUR96]. The areas in which they differ are the number of necessary routing related tables and the methods by which

changes in network topology are broadcast. All the proactive protocols try to keep the shortest path routers and routes are maintained to all potential destinations (possibly all nodes in the network) all the time, whether or not such routes are actually used. Route maintenance is obtained by route update traffic, and this can be a lot of overhead, especially for large networks. The proactive strategy autonomously distributes routing information and it is independent of the application. The application should experience no initial delay since the routes have already been established.

A.1.1 DSDV

The Destination-Sequenced Distance-Vector (DSDV) [PER94] Routing Algorithm is based on the idea of the classical Bellman-Ford Routing Algorithm with certain improvements. Each node maintains a list of all destinations and number of hops to each destination. Each entry is marked with a sequence number. The nodes periodically transmit their routing tables to their immediate neighbors. A node also transmits its routing table if a significant change has occurred in its table from the last update sent. So, the update is both time-driven and event-driven. It uses full dump or incremental packets to reduce network traffic generated by route updates. A full dump sends the full routing table to the neighbors whereas in an incremental update only those entries from the routing table that has a metric change since the last update are sent. When the network is relatively stable, incremental updates are sent to avoid extra traffic and full dump are relatively infrequent. In a fast-changing network, incremental packets can grow big so full dumps will be more frequent. The broadcast of route update is delayed by settling time to eliminate those updates that would occur if a better route were found very soon. The only improvement made here is avoidance of routing loops in a mobile network of routers. With this improvement, routing information can always be readily available, regardless of whether the source node requires route or not.

Unfortunately, the DSDV has some drawbacks. DSDV is inefficient because of the requirement of periodic update transmissions, regardless of network traffic. These update packets are broadcast throughout the network so every node in the network knows how to reach every other node. As the number of nodes in the network grows, the size of the routing tables and the bandwidth required to update them also grows. This overhead is DSDV's main weakness. Also, it requires carefully selection of the following parameters: periodic update interval, maximum value of the "settling time" for a destination and the number of update intervals which may transpire before a route is considered stale. These parameters will likely represent a tradeoff between the latency of valid routing information and excessive communication overhead.

A.1.2 WRP

The Wireless Routing Protocol (WRP) [MUR96] is a table-based distance-vector routing protocol. It belongs to the class of path-finding algorithm with the exception of avoiding the count-to-infinity problem by forcing each node to perform consistency checks of predecessor information reported by all its neighbors. Each node in the network maintains a Distance table, a Routing table, a Link-Cost table and a Message Retransmission list. Node exchange routing tables with their neighbors using update messages periodically as well as on link changes. If there is no change in routing table since last update, the node is required to send an idle Hello message to ensure connectivity. Hello messages are periodically exchanged between neighbors. On receiving an update message, the node modifies its distance table and looks for better paths using new information. The node also updates its routing table if the new path is better than the existing path. A unique feature of this algorithm is that it checks the consistency of all its neighbors every time it detects a change in link of any of its neighbors. Consistency check in this manner

helps eliminate looping situations in a better way and also has fast convergence. Therefore, this protocol avoids count-to-infinity problem by forcing each node to check predecessor information reported by all its neighbors.

Routing Protocols

A.2 On-Demand Routing Protocols

A.2.1 DSR

The Dynamic Source Routing Protocol (DSR) [JOH96, DAS00, ROY99a] is based on the concept of source routing. For this protocol, mobile nodes are required to maintain route caches that contain the source routes of which the mobile is aware. Entries in the route cache are continually updated as new routes are learned. There are two major phases of the protocol - route discovery and route maintenance. Route discovery uses route request and route reply packets. Route maintenance uses route error packets and acknowledgements. When the source node wants to send a packet to a destination, it looks up its route cache to determine if it already contains a route to the destination. If it finds an unexpired route to the destination, then it uses this route to send the packet. However if the node does not have such a route, then it initiates the route discovery process by broadcasting a route request packet. The route request packet contains the address of the source, the address of the destination, and a unique identification number. Each intermediate node checks whether it knows a route to the destination. If it does not, it appends its address to the route record of the packet and forwards the packet to its neighbors. To limit the number of route requests propagated, a node processes the route request packet only if it has not already seen the packet and it's address is not present in the route record of the packet.

An advantage of DSR over the other on-demand protocols is that DSR does not make use of periodic routing advertisements, thereby saving bandwidth and

reducing power consumption. Hence the protocol does not cause control overhead when there are no topology changes. However, DSR suffers from a scalability problem due to the nature of source routing. As the network becomes larger, the control packets and message packets also become larger. This gives a negative impact due to limited bandwidth. Furthermore, the need to place the entire route in both route replies and data packets causes more control overhead.

A.2.2 AODV

Ad hoc On-demand Distance Vector (AODV) [PER99, PER00a, ROY99a, ROY99b] routing algorithm builds on the DSDV algorithm described above. AODV minimizes the number of required broadcasts by creating routes on an on-demand basis, as opposed to maintaining a complete list of routes as in DSDV algorithm. A path discovery is initiated when a route to a destination does not exist. Broadcast is used for route request.

When a node has a packet and wants to send it to a destination for which it does not already have a valid route, it broadcasts a route request (RREQ) packet across the network. The RREQ packet contains the source address, a broadcast ID, the destination address, the destination sequence number, and a hop count. The pair, the source address and the broadcast ID, uniquely identifies a RREQ. As the RREQ packet travels from a source to various destinations, the reverse path from all the nodes back to the source is set up. To set up a reverse path, a node records the address of the neighbor from which it received the first copy of the RREQ. These reverse route entries are maintained for a sufficient time period to allow the RREQ to traverse the network. Nodes receiving this packet update their information for the source node and set up backwards pointers to the source node in the route tables.

A node receiving the RREQ packet may send a route reply (RREP) if it is either the destination or if it has a route to the destination with a corresponding

sequence number greater than or equal to that contained in the RREQ. If this is the case, it sends a RREP back to the source. Otherwise, it rebroadcasts the RREQ. A RREP packet is unicasted to the node from which the RREQ was received, and the node sets a forward pointer to the issuer of the RREP. Every mobile node forwards the RREP packet to the neighbor to which it previously set a reverse path. By the time the RREP reaches the source, a route is set up from the source to the destination. The source node may now begin to forward data packets to the destination.

In AODV, routes are maintained as follows. If the source moves then it can reinitiate route discovery to the destination. If one of the intermediate nodes move then the moved nodes neighbor realizes the link failure and sends a link failure notification to its upstream neighbors and so on till it reaches the source upon which the source can reinitiate route discovery if needed.

The most notable here is that the overhead of AODV is potentially smaller than that of DSR since each DSR packet must carry full routing information as we stated above, whereas in AODV packets just need to contain the destination address. On the other hand, AODV requires symmetric links between nodes, and hence cannot utilize routes with assymetric links when symmetric links are not available.

A.2.3 TORA

The Temporally Ordered Routing Algorithm (TORA) [PAR01, PAR97] is a highly adaptive, efficient and scalable distributed routing algorithm based on the concept of link reversal. TORA is proposed for highly dynamic mobile, multihop wireless networks. It is a source-initiated on-demand routing protocol. It finds multiple routes from a source node to a destination node. The main feature of TORA is that the control messages are localized to a very small set of nodes near

the occurrence of a topological change. To achieve this, the nodes maintain routing information about adjacent nodes. The protocol has three basic functions: Route creation, Route maintenance, and Route erasure

TORA Protocol provides loop free paths at all instants. It also provides multiple routes so that if one path is not available, other is readily available. It establishes routes quickly so that they may be used before the topology changes. It minimize algorithmic reactions/communication overhead and thus conserves available bandwidth and increases adaptability. It is also able to detect network partitions very quickly. However, since it uses internodal co-ordination, it exhibits instability behavior similar to "count-to-infinity" problem in distance vector routing protocols. Although such oscillations are temporary and route convergence will ultimately occur. In TORA, there is a potential for oscillations to occur, especially when multiple sets of coordinating nodes are concurrently detecting partitions, erasing routes, and building new routes based on each other.

A.3 Cluster-Based Routing Protocols

In Cluster-based routing protocols, an ad hoc network is represented as a set of clusters. Contrary to on-demand protocols, clustering protocols take advantage of their cluster structure, which limits the scope of route query flooding. Because clusterheads and gateways (nodes that lie within the transmission range of more than one clusterhead) are only nodes flooded with query packets, the bandwidth required by the route discovery mechanism is reduced. The clusterheads will be responsible for routing messages between nodes, while the gateways are used to maintain communication between two or more clusterheads. Clusterheads and gateways form the virtual backbone of the network.

The objective of a clustering algorithm is to produce and maintain a connected cluster. Connectivity is defined as the probability that a node is reachable from

any other node. A clustering algorithm consists of two phases: the set up and the maintenance. Algorithms differ on the criterion for the selection of the clusterheads (CHs) in the cluster set up phase. Choosing the CHs optimally is an NP-hard problem. Any node can become an CH if it has the necessary functionality, such as processing and transmission power.

A.3.1 CGSR

The Clusterhead Gateway Switch Routing (CGSR) [ROY99a, CHI97] protocol, instead of a flat network, is a clustered multihop mobile wireless network. It uses DSDV [PER94] as underlying routing protocol. However, DSDV has been modified by using a hierarchical cluster head-to-gateway routing approach to route traffic from source to destination. Gateway nodes are nodes that are within communication range of two or more clusterheads. Each node maintains two tables: a cluster member table, which maps the destination node address to its clusterhead address, and a routing table, which shows the next hop to reach the destination cluster. Both tables contain sequence numbers to eliminate stale routes and prevent looping. These tables are broadcasted by each node periodically using DSDV algorithm.

When a source node has a packet to sent, it is first routed to its clusterhead. Then the packet is routed from the clusterhead to a gateway to another clusterhead, and so on until the clusterhead of the destination is reached. The packet is then transmitted to the destination. On receiving a packet, a node will check its cluster member table and its routing table to determine the nearest clusterhead along the route to the destination. Next the node will check its routing table to determine the node in order to reach the selected cluster head. It then transmits the packet to this node.

In CGSR, because routing performance is dependent on the status of specific nodes (clusterhead and gateway), time complexity of a link failure associated

with a clusterhead is higher than DSDV, Also, additional time is needed to perform clusterhead reselection. Since each node declares it is a gateway node to its neighbors if it responding to multiple radio codes. Therefore, there is no specific gateway selection algorithm. If a gateway moves out of range, the routing protocol is responsible for routing the packet to another gateway.

A.3.2 VBS

In Virtual Base Station (VBS) [HAS01] routing protocol, each VBS is in charge of a set of MTs. Only certain nodes are eligible to acquire knowledge of the full network topology (VBS and BMT). Route requests are not flooded to the rest of the ad hoc network. If an MT wishes to send a packet to another MT in the network, first, it sends the packet to its VBS, which forwards the packet to the VBS in charge and then to the destination or the correct BMT. The sent packet contains the address of the destination. When the VBS receives the message, it searches the destination address in its table. If the destination is found, the VBS of the source MT will forward the packet to the VBS in charge of the destination. This is done by consulting the BMT field of that VBS. The message is then forwarded to the MT (or VBS) whose ID number is stored in the BMT field. The BMT, after receiving the message, forwards it to its own VBS. This process is repeated until the message reaches the destination. It is obvious that MTs are neither responsible for discovering new routes, nor maintaining existing ones. As a result this routing scheme eliminates the initial search latency, which degrades the performance of interactive and multimedia applications.

Although, VBS outperforms CGSR in more stressful situations with a widening performance gap, all the nodes require the aid of their VBS(s) all the time, this results in a very high MAC contention on the VBSs. Also, the periodic hello message updates are not efficiently utilized by MTs (other than VBSs and BMTs).

Therefore, VBS-O, which will be described in the next section, is suggested to illuminate the VBS's drawbacks and then improve the performance. Since the VBS-O has been proven to be very efficient, as it maximizes the throughput and reduces the packet delivery time, we adapt it in this work.

A.4 Constraint Routing Protocols

A.4.1 WEAC Infrastructure Protocol

As in all cluster-based protocols, in this scheme, some of the MTs, based on an agreed-upon policy, are elected to be in charge of all the MTs within their transmission ranges, or a subset of them. This can be achieved by electing one to be a clusterhead. Every MT acknowledges its location via hello packets, sometimes called beacon packets. The method of electing a clusterhead from a set of nominees is based on its battery power level (BPL). Another issue to be addressed is the handing of responsibilities of a clusterhead over from one clusterhead to another. MTs are classified as follows:

- Clusterhead: as it is named, the leader of the cluster.
- ZoneMT: an MT supervised by a CH.
- FreeMT: an MT that is neither a CH nor zoneMT, (i.e. it is not associated with a cluster).
- Gateway or Border Mobile Terminal: MT that lies between more than one CH or FreeMT, it can be a CH or a zoneMT or a FreeMT.

Each MT has a $myCH$ variable. An MTs $myCH$ variable is set to the ID number of its CH; however, if that MT itself is a CH, then the $myCH$ variable will be set to zero, otherwise it will be set to -1, indicating that it is a CH of itself or a free node. A CH collects complete information about all other CHs and their lists

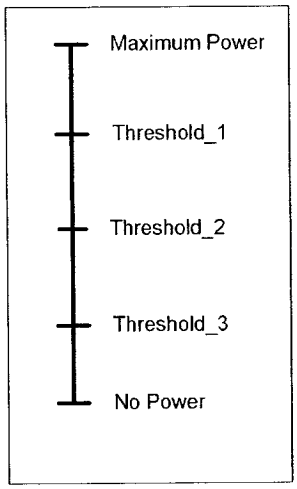


Figure A.1: Four power levels of each MT

of MTs and broadcasts this information in its periodic hello messages. Zone-MTs accumulate information about the network from their neighbours between hello messages, and they broadcast their neighbour-list to their neighbours in their hello packets. MTs announce their ID number with their periodic hello message. An MT sends a *merge-request* message to another MT if the latter has a higher BPL and it should be more than or equal to THRESHOLD_1, (it will be explained later). The receiver of the *merge-request* message responds with *accept-merge* message and sets its $myCH$ variable to zero. When an MT receives the *accept-merge* message it will set its $myCH$ variable to the ID number of its CH. The BPL of each and every MT is characterized into one of the following four categories, 1:

- MT $BPL \geq THRESHOLD_1$: An MT is eligible to be a CH and willing to accept other MTs to be under its supervision if these MTs have a lower BPL. If the MTs with the same BPL, which is almost impossible, then the one with more number of neighbours wins.

- $THRESHOLD_1 >$ MT $BPL \geq THRESHOLD_2$: An MT will ignore any *merge request* messages that are sent to it by other MTs. If the MT is serving as a CH, it will remain a CH, but it will add no more nodes under its supervision.

- $THRESHOLD_2$ > MT $BPL \geq THRESHOLD_3$: If an MT is serving as a CH, it sets a warning flag to inform all MTs under its supervision to look for another CH, nonetheless, they can remain with it till its BPL drains to $THRESHOLD_3$.

- MT $BPL < THRESHOLD_3$: If it was serving other nodes, an MT will ignore any *merge request* messages and will set $iAmNoLongerYourCH$ flag to inform all the nodes under its supervision.

In all cases above, if the myCH > 0, no *merge request* will be sent by MTs. However, if myCH = -1, then it will send a *merge request* to an MT whose BPL is greater than or equal to $THRESHOLD_1$.

A.4.2 VBS-O Routing Protocol

In this section we introduce the Virtual Base Station On-demand routing protocol [SHE02a]. This protocol can be run on top of any infrastructure protocol. As it is named, there are some on-demand features added to the VBS-O routing protocol. Each Virtual Base Station (VBS) or clusterhead is in charge of a set of nodes of its neighbor. Only VBSs and freeMTs are eligible to acquire knowledge of the full network topology. Therefore, VBSs, BMTs and freeMTs construct the virtual backbone of the network. In the VBS-O routing protocol, hello messages are periodically broadcasted by each MT in the network.

Since neighboring nodes, which are one hop away from each other, can hear each other, neighboring nodes are suggested to communicate with each other without the aid of their clusterheads. This is as opposed to the VBS routing protocol case [HAS01]. Additionally, the new technique of broadcasting the neighbor list, used by the WEAC protocol, speeds up even more packet delivery (especially between neighboring nodes) and improves significantly the throughput.

If an MT wishes to send a packet, it executes the Packet Decision Forwarding algorithm [SHE02a]. The process of this algorithm is as follows: first it looks into its neighbor list, if the destination is not found, it looks into the neighbor lists of its neighbors (i.e. if the destination is the neighbor or the neighbor's neighbor) and then it sends the packet to that particular neighbor. If it has more than one access to the destination, it checks their ELs, then it shortlists the ones with EL more than $THRESHOLD_2$ and sends the packet to the one with the least number of neighbors. If none of the MTs ELs are more than $THRESHOLD_2$ it sends the packet to the one with the highest EL.

However, if an MT wishes to send a packet to another MT that is more than two hops away, the behavior differs from one class of MTs to another. If the MT is zoneMT, first, it sends the packet to its VBS. The VBS looks up its routing table and forwards the packet to the correct neighbor or BMT or the destination. At any time, if the destination is the neighbor or neighbor's neighbor, the packet will be forwarded to that neighbor . The sent packet contains the destination address and the source node.

If the BMT is a zoneMT, it first performs the Packet Decision Forwarding algorithm, if the destination is more than two hops away, it broadcasts the destination to all VBSs and freeMTs in its transmission range. If one of them has an access to the destination, which is most probably the case, it replies to the issuer of the request. The issuer of the request waits for some time then sends the packet to its VBS.

The main properties of the VBS-O protocol are:

- Reduces the MAC contention especially on clusterheads and balances the load: Since MTs are free to contact the neighboring nodes and even the

neighbor's neighbor, the MAC contention will be distributed among the network. In this case clusterheads will not carry out packet delivery all the time. As a result the load will be balanced and MAC contention will be reduced.

- Minimizes the energy consumption and expends the lifetime of the network: The selection of gateways and clusterheads is based on their EL (and / or number of neighbors in the case of gateways), as a result, the consumed energy will be distributed among all MTs and the network will be highly unlikely to suffer any lack of communication in its lifetime.

- Reduces the delay time of packet delivery, especially between the neighboring nodes and the neighbor's neighbors nodes in the network: VBS-O takes advantage of using the WEAC protocol. Since all MTs broadcast their neighbor list, then any MT can access any other MT within two hops. This speeds up packet delivery and minimizes the energy consumed in packet delivery.


تم بحمد الله