



ADDIS ABABA UNIVERSITY  
SCHOOL OF GRADUATE STUDIES  
FACULTY OF TECHNOLOGY  
ELECTRICAL AND COMPUTER ENGINEERING  
DEPARTMENT

**Design, Implementation and Performance Evaluation of Cluster Based AODV  
Routing Protocol**

By  
Miftah Seid Hamid

A thesis submitted to the school of Graduate studies of Addis Ababa University in  
partial fulfillment of the requirements for the degree of

Masters of Science in Electrical and Computer Engineering

(Computer Engineering)

October, 2010

Addis Ababa, Ethiopia

ADDIS ABABA UNIVERSITY  
SCHOOL OF GRADUATE STUDIES  
FACULTY OF TECHNOLOGY  
DEPARTMENT OF ELECTRICAL AND COMPUTER  
ENGINEERING

**Design, Implementation and Performance Evaluation  
of Cluster Based AODV Routing Protocol**

By

Miftah Seid Hamid

Advisor

Dr. Kumudha Raimond

ADDIS ABABA UNIVERSITY  
SCHOOL OF GRADUATE STUDIES

ADDIS ABABA UNIVERSITY  
SCHOOL OF GRADUATE STUDIES  
FACULTY OF TECHNOLOGY

Design, Implementation and Performance Evaluation  
of Cluster Based AODV Routing Protocol

By  
Miftah Seid Hamid

APPROVAL BY BOARD OF EXAMINERS

\_\_\_\_\_  
Chairman Dept. of Graduate  
Committee

\_\_\_\_\_  
Signature

Dr. Kumudha Raimond  
Advisor

\_\_\_\_\_  
Signature

Dr. Dereje Hailemariam  
Internal Examiner

\_\_\_\_\_  
Signature

Mr. Yalemzewd Negash  
External Examiner

\_\_\_\_\_  
Signature

ADDIS ABABA UNIVERSITY  
SCHOOL OF GRADUATE STUDIES

Declaration

I, the undersigned student declare that, this thesis work is my original work, has not been presented in any other universities and all sources of materials used for the thesis work have been fully acknowledged.

Name: Miftah Seid

Signature: \_\_\_\_\_

Place: Addis Ababa

Date of submission: September 2010

This thesis has been submitted for examination with my approval as a university advisor.

Dr. Kumudha Raimond

Signature: \_\_\_\_\_

Advisor's Name

# Contents

Acknowledgement .....	iv
List of Figures .....	v
List of Tables.....	vi
Acronyms.....	vii
Abstract.....	ix
Chapter One: Introduction .....	10
1.1 Background .....	10
1.2 Problem Description .....	11
1.3 Motivation.....	12
1.4 Objective .....	13
1.4.1 General Objective .....	13
1.4.2 Specific objectives.....	13
1.5 Thesis Scope .....	13
1.6 Contribution .....	14
1.7 Methodology.....	14
1.8 Related Work.....	15
1.9 Outline of the Thesis .....	16
Chapter Two: MOBILE AD HOC NETWORKS (MANET).....	17
2.1 Introduction MANET .....	17
2.2 MANET Characteristics .....	18
2.3 MANET Applications.....	19
2.4 MANET ROUTING PROTOCOLS.....	20
2.4.1 PROACTIVE ROUTING PROTOCOLS (PRP) [2].....	20
2.4.1.1 DSDV .....	20
2.4.1.2 OSPF .....	20
2.4.1.3 OLSR.....	21
2.4.2 REACTIVE ROUTING PROTOCOL (RRP).....	21
2.4.2.1 AODV .....	21
2.4.2.2 DSR.....	23

2.4.2.3 Cluster Based Routing Protocol (CBRP) .....	23
2.4.3 HYBRID ROUTING PROTOCOLS.....	23
2.4.3.1 ZRP .....	24
2.4.3.2 LAR: .....	24
Chapter Three: AODV Routing Protocol .....	26
3.1 Introduction.....	26
3.2 Routing tables.....	27
3.3 Control messages .....	27
3.3.1 Routing request.....	27
3.3.2 Routing reply.....	28
3.3.3 Route error.....	28
3.3.4 HELLO messages.....	29
3.4 Route discovery .....	29
3.4.1 Reverse path setup .....	29
3.4.2 Forward path setup .....	30
3.5 Link breakage .....	30
3.6 PROPERTIES OF AODV .....	31
3.6.1 Merits of AODV.....	31
3.6.2 Drawbacks of AODV .....	31
Chapter Four: Clustering Algorithms.....	34
4.2 Identifier-based clustering.....	36
4.2.1 Lowest ID cluster algorithm (LIC) .....	36
4.2.2 Max-Min d-cluster formation algorithm .....	37
4.3 Connectivity-based clustering.....	37
4.4 Mobility-aware clustering .....	38
4.6 Power-Aware Clustering .....	39
4.7 Combined-weight based clustering .....	41
CHAPTER Five: Design of CBAODV routing protocol.....	42
5.1 Network Simulator (NS) .....	42
5.1.1 Marc Greis' Tutorial.....	43

5.1.2 NS by Example .....	43
5.1.3 NS Manual, NS Search, and NS Mailing List .....	44
5.2 Network Animator (NAM) .....	44
5.3 Design of CBAODV routing protocol .....	45
5.3.2 Cluster Formation .....	47
5.3.3 Adjacent Cluster Discovery [26] .....	50
5.3.4 Routing Considerations .....	52
5.4 Design of CAODV .....	55
Chapter Six: Simulation, Results and Result Analysis .....	56
6.1 Simulation Scenarios and Model .....	56
6.2 The traffic and mobility models.....	57
6.3 Evaluation Metrics .....	58
6.3.1 Average Packet Delivery Ratio (APDR) .....	58
6.3.2 Average end-to-end Delay (AEED) .....	59
6.3.2 Average Normalized Routing Load (ANRL) .....	59
6.4 Simulation Results .....	60
6.4.1 Scenario 1: Thirty Connections .....	60
6.4.1.1 Average Packet Delivery Ratio (APDR).....	61
6.4.1.2 Average Normalized Routing Load (ANRL).....	62
6.4.1.3 Average end-to-end Delay (AEED).....	63
6.4.2 Scenario 2: Twenty Connections .....	64
6.4.2.1 Average Packet Delivery Ratio (APDR).....	64
6.4.2.2 Average Normalized Routing Load (ANRL).....	65
6.4.2.3 Average end-to-end Delay (AEED).....	66
Chapter Seven: Conclusions and future work.....	69
References .....	70
Appendix A: AWK SCRIPTS FOR METRICS CALCULATIONS.....	74
Appendix B: TCL SCRIPTS .....	76

## Acknowledgement

My deepest gratitude goes to My Advisor, Dr. Kumudah Raimond, for her dedication, continuous follow-up and support throughout the thesis work.

I wish to extend my acknowledgement to my families, Mahi-Seids' and friends for their invaluable encouragement and supports.

Finally, I would like to thank The Almighty Allah, as nothing would have been possible without his help.

## List of Figures

Figure 1.1. Mobile ad hoc network .....	18
Figure 2.1. Propagation of RREQ in AODV .....	22
Figure 2.2. Path taken by the RREP in AODV .....	24
Figure 3.1 AODV Protocol Messaging .....	27
Figure 3.2 AODV flow chart .....	33
Figure 4.1. Cluster structure Illustration .....	35
Figure 5.1 Modified HELLO message format .....	46
Figure 5.2 Neighbor Table Extension in a HELLO message .....	46
Figure 5.3 A state transition diagram of cluster formation .....	48
Figure 5.4 Format of CAT in each node .....	50
Figure 5.5 Format of the cluster Adjacency Extension to Hello Message ..	51
Figure 5.6 CBAODV Route Establishment .....	52
Figure 5.7 Receive Request (recvRequest()) operation at intermediate node.....	54
Figure 6.1 Packet Delivery Fraction vs different mobility, scenario1.....	61
Figure 6.2 Average Normalization Routing Load vs different mobility, scenario1.....	62
Figure 6.3 Average End to End Delay vs different mobility, scenario1.....	63
Figure 6.4 Packet delivery fraction vs. different Node Mobility, scenario2. .....	64
Figure 6.5 Average Normalization Routing Load vs different nodes mobility, scenario2. ....	64
Figure 6.6 Average End to End Delay vs different node mobilities, scenario2 .....	66

## List of Tables

Table 6.1: Simulation parameters value .....	57
--	----

## Acronyms

AEED	Average End to End Delay
ANRL	Average Normalized Routing Load
AODV	Ad hoc On-demand Distance Vector
APDR	Average Packet Delivery Ratio
ARC	Adaptive Routing using Clusters
ARCH	Adaptive Routing using Clustered Hierarchies
CAT	Cluster Adjacency Table
CBAODV	Cluster Based AODV
CBR	Constant Bit Rate
CBRP	Cluster Based Routing Protocol
CH	Cluster Head
DSDV	Destination-Sequenced Distance-Vector
DSR	Dynamic Source Routing
EED	End to End Delay
FIFO	First-In-First-Out
GW	Gateway
HCC	Highest Connectivity Clustering
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IP	Internet Protocol
LAN	Local Area Network
LAR	Location Aided Routing
LBC	Load balancing clustering
LIC	Lowest ID Cluster
MAC	Media Access Control
MANET	Mobile Ad hoc Network
NAM	Network Animator

NRL	Normalized Routing Load
NS	Network simulator
OTcl	Object Tool command language
OLSR	Optimized Link State Protocol
OSPF	Open Shortest Path First
PDR	Packet Delivery Ratio
PRP	Proactive Routing Protocol
RREP	Route REPLY
RREQ	Route REQuest
RERR	Route ERRor
RRP	Reactive Routing Protocol
TCP	Transmission Control Protocol
TTL	Time To Live
UDP	User Datagram Protocol
VID	Virtual ID
WCA	Weighted Clustering Algorithm
ZRP	Zone Routing Protocol

## Abstract

Ad Hoc Networks are multi-hop wireless networks with dynamically changing network connectivity due to mobility. Each node participating in the network acts both as host and a router and must therefore is willing to forward packets for other nodes. A number of routing protocols like Dynamic Source Routing (DSR), Ad Hoc On-Demand Distance Vector Routing (AODV) and Cluster Based Routing Protocol (CBRP) were proposed and their performance was compared by several researchers.

In this paper, Cluster Based AODV (CBAODV) routing protocol is designed and implemented on ns-2.29 simulator by using Energy Based clustering algorithms to solve the scalability problem and by retaining the merits of AODV. A detailed simulation has been carried out and its performance has been compared with pure AODV, DSR and CBRP routing protocols and also with its preliminary version, Clustered AODV (CAODV) which used lowest ID clustering algorithm for cluster formation. The performance differentials are analyzed using varying network load, mobility, and network size. The metrics used for performance analysis are Packet Delivery Ratio (PDR), End-to-End Delay (EED), and Normalized Routing Load (NRL).

Results show that the CBAODV routing has 3.12% and 1.15% improvement in PDR over the flat AODV and CAODV protocols respectively. It also introduces less overhead than the two. However, the results also indicate that the two source routing based protocols, DSR and CBRP, have very high PDR which is up to 94% in highest mobility models, whereas AODV and CBAODV achieve only up to 81%. Meanwhile, the three distance vector based protocols, AODV, CBAODV and CAODV, exhibits a very short end-to-end delay of data packets, up to 0.4 second, but 1.2 second in case of DSR and CBRP.

**Key words:** MANET, AODV, CBAODV, DSR, CBRP, Clustering

# Chapter One: Introduction

## *1.1 Background*

Mobile networks can be classified into infrastructure networks and Mobile Ad Hoc Networks (MANET) according to their dependence on fixed infrastructures [2]. In an infrastructure mobile network, mobile nodes have wired access points (or base stations) within their transmission range. In contrast, MANET are autonomously self-organized networks without support of infrastructure; It has become increasingly popular in recent years as a way to provide instant networking between groups of people that may not all be within transmission range of one another.

Applications for these types of networks range from campus and conference scenarios to emergency operations and to military scenarios. The projected number of users in each of these applications ranges from a handful of people in some emergency situations, to tens and hundreds of people in campus and conference scenarios, to thousands and tens of thousands of people in military applications [5].

In MANET, a message sent by a mobile node may be received simultaneously by all the nodes in its neighbors. Messages directed to mobiles not within the sender's transmission range must be forwarded by neighbors, which thus act as routers. Due to mobility, it is not possible to establish fixed paths for message delivery through the network.

A number of routing protocols have been proposed for MANET networks. They can be divided into two basic categories: Proactive routing protocols like Destination-Sequenced Distance-Vector (DSDV) [36] and Reactive routing protocols like DSR and AODV. Proactive protocols maintain routes between each

pair of nodes throughout the lifetime of the network. While this approach has the benefit that a route is generally available the moment it is needed, proactive protocols have poor scaling properties due to their overhead. Also, previous work has shown that these protocols do not perform as well as reactive routing protocols in most scenarios [5]. Reactive protocols, on the other hand, only establish routes on-demand, or when needed. These protocols only incur overhead for route construction and maintenance when those routes are actually needed, since they do not maintain routes that are not utilized. The drawback to these protocols is that they introduce a route acquisition latency, or a period of waiting to acquire a route after the route is needed. These protocols have been shown to also have limited scalability, due to their route discovery and maintenance procedures. One alternative to these protocols for improving scalability is clustering, or hierarchical, routing protocols like CBRP. Hierarchical protocols place nodes into groups, often called clusters. These groups may have some sort of cluster leader that is responsible for route maintenance within its cluster and between other clusters. The motivation for implementing hierarchical routing algorithms is that they tend to be more scalable, due to their intrinsic characteristics [3].

## ***1.2 Problem Description***

In MANET, two nodes communicate directly if they are within transmission range of each other. Otherwise, they must communicate via a multihop route. To find such a multi-hop route, MANETs commonly employ on demand routing algorithms like AODV that use flooding or broadcast messages. In flooding, a node transmits a message to all of its neighbors. The neighbors in turn relay to their neighbors and so on until the message propagates to the entire network. As one can easily see, the performance of flooding is closely related to the average number of neighbors (neighbor degree) in the network. As the neighbor degree gets higher, flooding suffers from the increase of (1) redundant and superfluous

packets, (2) probability of collision, and (3) congestion of wireless medium [33]. Performance of flooding is severely impaired especially in large and dense networks [34].

AODV is one of MANET's flat routing protocol in which its multicast schemes depend on massive flooding. Several researches [5] and [1] show that AODV is well suited for MANET in that it has low processing and memory overhead and low network utilization. Nevertheless, it has a problem of poor scalability because of its flooding scheme. Research has demonstrated its performance bottleneck both theoretically and through simulation experiments[5]. It's thought that building a hierarchical ad hoc network is a very promising way to achieve scalability [4]

### ***1.3 Motivation***

There have been several attempts to cluster AODV and to direct its scalability problem [3] , [4], and [10]. Because of the possibility of using various clustering algorithms for cluster formation the search for an effective clustering and scalable AODV is still an open issue. Due to this, the motivation behind this thesis work arises from the availability of various clustering algorithms and their potential advantages for improving scalability of AODV as well MANET networks.

Researches in [29] and [32] made a performance comparisons of AODV, DSR and CBRP protocols which is a comparisons of one distance vector (AODV) and two source routing protocols (DSR and CBRP), in another expression a comparison of two flat routing (AODV and DSR) and one hierarchical (CBRP) protocols. As this thesis work come up with CAODV and CBAODV which are cluster based distance vectors routing protocols, due to this, another motivation come which can be considered an extension of the researchers that evaluates the performance of equivalent distance vectors and source routings in terms of flat and hierarchical routing protocols.

## ***1.4 Objective***

The objective of this thesis work was to analyze, design, implement and evaluate the Cluster Based AODV routing protocols for wireless ad-hoc networks. At the beginning of this thesis, no implementation of clustering algorithms with AODV routing protocols had been released, so the first main task was to analyze, design and implement the algorithm.

### **1.4.1 General Objective**

The main objective of this thesis is to analyze, design and implement CAODV and CBAODV and to compare its performance with the flat AODV and DSR and the clustered CBRP protocols.

### **1.4.2 Specific objectives**

- Get a general understanding of Mobile ad-hoc networks.
- Design CBAODV.
- Implement the Energy aware clustering algorithm on the existing AODV routing protocols.
- And finally performance measurement of the CBAODV through simulation under different scenarios will be carried out, and its performance will be compared with the existing AODV, DSR and CBRP protocols.

## ***1.5 Thesis Scope***

This research focuses on scaling AODV routing protocol by using clustering approaches. The energy based and lowest ID clustering algorithms are used for clusterhead election.

Cluster Based AODV (CBAODV) routing protocol is developed by using Energy Based clustering algorithms to solve the scalability problem of AODV. A detailed

simulation has been carried out and its performance has been compared with pure AODV, DSR and CBRP routing protocols and also with its preliminary version, Clustered AODV (CAODV) which used lowest ID clustering algorithm for cluster formation.

In this paper the metrics used for performance analysis are Packet Delivery Ratio (PDR), End-to-End Delay (EED), and Normalized Routing Load (NRL).

### ***1.6 Contribution***

The two main contributions of this thesis are:

- (a) Analysis and design of CBAODV and CAODV
- (b) Performance comparison of CBAODV, CAODV, AODV, DSR and CBRP protocols.

### ***1.7 Methodology***

The main goal of this thesis is to develop, implement and evaluate CBAODV by using Energy aware clustering algorithm. The work requires a good knowledge in computer network, clustering and MANET, which requires vigorous study of the systems, literatures and articles related to this field.

**Literature Review:** The first phase of the project is reviewing the literatures about clustering and routing protocols in MANET, which includes articles, books, research papers, white papers, class lecture notes, research publications and information available through the Internet

**Analysis:** The second phase includes in-depth analysis of the Energy aware clustering algorithms and the AODV routing protocol.

Analyze and select the suitable metrics for the evaluating the models.

**Design:** The third phase focus on the Design of CBAODV based on the analysis.

**Experiment:** The fourth phase deals with conducting experiments using NS-2. The platform used for the research work possesses the following software and hardware components:

- Operating System: Open SuSE (ver. 10.3)
- Simulator: NS2 (ver. 2.29)
- Hardware: Dual-core processor machine 2.0 GHz, 3.00GB memory

**Evaluation:** The last phase is associated with evaluating the results of the models using different scenarios.

### ***1.8 Related Work***

There are numerous proposals for clustering and hierarchical routing schemes. This section presents samples of those protocols, including those that are most closely related to the Clustered AODV.

In [3] the researcher presents two hierarchical clustering protocols that improve the scalability of ad hoc routing protocols. The Adaptive Routing using Clusters (ARC) protocol creates a one-level clustered hierarchy across an ad hoc network, while the Adaptive Routing using Clustered Hierarchies (ARCH) protocol creates a multi-level hierarchy which is able to dynamically adjust the depth of the hierarchy in response to the changing network topology. It is experimentally shown that these protocols, when coupled with AODV routing protocol, produce throughput improvements of up to 80% over the AODV protocol alone. But these protocols performance with high mobility network and its end to end delay and normalization load didn't evaluated.

In [4] the scalability problem of AODV is tried to be solved by using clustering scheme. The cluster head selecting is made by using RREP message, On the way which RREP pass by, one or several nodes are selected as cluster head (CH) using a defined rule, but this defined rule is not stated clearly in the paper.

Denko et al. in [10] proposed an AODV-based clustering for MANET, a mobility-based clustering algorithm used for cluster formation. In this algorithm, a node is elected as a cluster head only when its mobility index is below a certain threshold. The mobility index is computed based on cluster membership changes and the number of cluster head changes [10]. In the network each node broadcasts its mobility information to its neighbors during the cluster head election phase. After collecting information from neighbors, each node checks whether it has the lowest mobility index. Once it confirms this, it sets itself as a cluster head and notifies its neighbors.

In [29] and [32] the researchers made a performance comparison of CBRP, AODV and DSR routing protocols. The results indicate that the two source routing based protocols, DSR and CBRP, have very high throughput while the distance vector based protocol, AODV, exhibits a very short end-to-end delay of data packets. Furthermore, despite its improvement in reducing route request packets, CBRP has a higher routing overhead than DSR because of its periodic hello messages. DSR has much smaller routing overhead than AODV and CBRP. AODV has the largest overhead among the three protocols. As a future work the researcher suggested clustering structure to reduce the routing overhead of the protocols, which is one of the intentions of this thesis work.

### ***1.9 Outline of the Thesis***

The rest of this paper is organized as follows: chapter two deals with the general basic concepts about mobile ad hoc networking. AODV routing protocol and existing clustering algorithms are discussed in chapter three and four respectively. In chapter five the design of CBAODV and its implementation procedure are discussed. In chapter six results of the simulation and the corresponding discussions are provided. Finally, the conclusion and recommendations together with the possible outlooks for future work is provided in chapter seven.

# Chapter Two: MOBILE AD HOC NETWORKS (MANET)

## *2.1 Introduction MANET*

Mobile ad hoc networks are formed dynamically by an autonomous system of mobile nodes that are connected via wireless links without using an existing network infrastructure or centralized administration [2]. The mobile nodes form a network as they become aware of each others presence, they communicate directly with devices inside their radio range in a peer-to-peer nature. They are free to move randomly and organize themselves arbitrarily; thus, the network's wireless topology may change rapidly and unpredictably. Mobile ad hoc networks are infrastructureless networks since they do not require any fixed infrastructure such as a base station for their operation. In general, routes between nodes in an ad hoc network may include multiple hops and, hence, it is appropriate to call such networks "multihop wireless ad hoc networks" [2]. Figure 1.1 show an example mobile ad hoc network and its communication topology.

As shown in Figure 1.1, an ad hoc network might consist of several home-computing devices, including notebooks, handheld PCs, and so on. Each node will be able to communicate directly with other nodes that reside within its transmission range. For communicating with nodes that reside beyond this range, the node needs to use intermediate nodes to relay messages hop by hop.

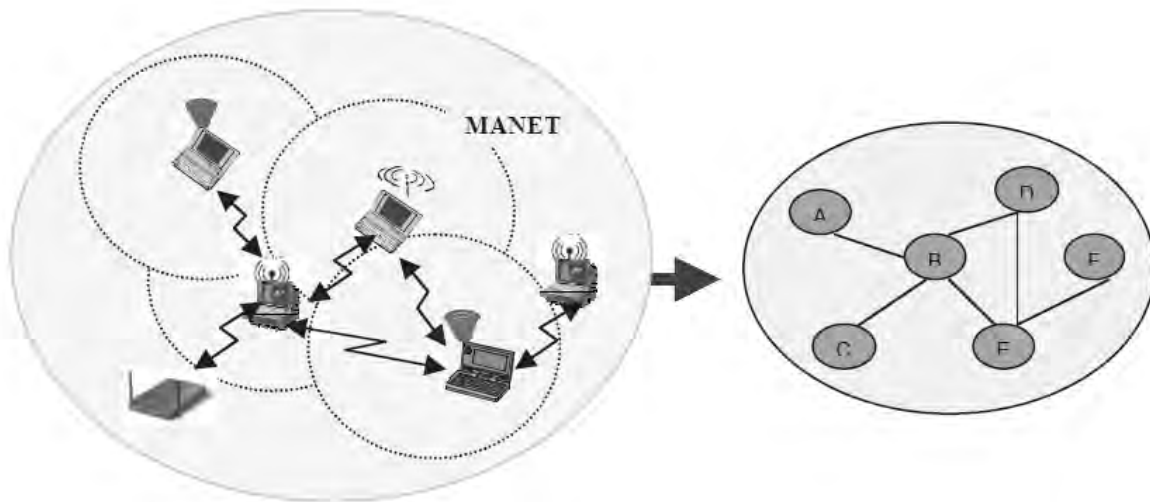


Fig 1.1. Mobile ad hoc network [2]

## ***2.2 MANET Characteristics***

As described above MANET is a collection of independent mobile nodes that can communicate to each other via radio waves. The mobile nodes that are in radio range of each other can directly communicate, whereas others need the aid of intermediate nodes to route their packets. These networks are fully distributed, and can work at any place without the help of any infrastructure. The characteristics of these networks are summarized as follows [12]:

- Communication via wireless means.
- Nodes can perform the roles of both hosts and routers.
- No centralized controller and infrastructure. Intrinsic mutual trust.
- Autonomous, no infrastructure needed.
- Can be set up anywhere.
- Energy constraints.
- Limited security

Generally, the communication terminals have a mobility nature which makes the topology of the distributed networks time varying. The dynamical nature of the network topology increases the challenges of the design of ad hoc networks. Each radio terminal is usually powered by energy limited power source (as

rechargeable batteries). The power consumption of each radio terminal could be divided generally into three parts, power consumption for data processing inside the node, power consumption to transmit its own information to the destination, and finally the power consumption when the node is used as a router, i.e. forwarding the information to another node in the network. The energy consumption is a critical issue in the design of the ad hoc networks.

The mobile devices usually have limited storage and low computational capabilities. They heavily depend on other hosts and resources for data access and information processing. A reliable network topology must be assured through efficient and secure routing protocols for Ad Hoc networks.

### ***2.3 MANET Applications***

Nowadays MANET is gaining importance with the increasing number of widespread applications due to its portable devices. Because ad hoc networks are flexible that can be set up anywhere at any time, without infrastructure, including pre-configuration or administration, people have come to realize the commercial potential and advantages that mobile ad hoc networking can bring[2].

The following are the applications of ad hoc wireless networks [8]:

- Community network
- Enterprise network
- Home network
- Emergency response network
- Vehicle network
- Sensor network

In addition, mobile ad hoc networks have primarily been used for tactical network-related applications to improve battlefield communications and survivability. The dynamic nature of military operations means it is not possible

to rely on access to a fixed preplaced communication infrastructure on the battlefield [2].

## **2.4 MANET ROUTING PROTOCOLS**

Numerous routing protocols have been developed for ad hoc mobile networks, which can be broadly classified into proactive, reactive and hybrid protocols as discussed bellow.

### **2.4.1 PROACTIVE ROUTING PROTOCOLS (PRP) [2]**

In proactive (table-driven) protocols, nodes periodically search for routing information within a network. The control overhead of these protocols is foreseeable, because it is independent to the traffic profiles and has a fixed upper bound. This is a general advantage of proactive routing protocols.

#### **2.4.1.1 DSDV**

The Destination-Sequenced Distance-Vector (DSDV) Routing protocol is based on the idea of the classical Bellman-Ford Routing Algorithm with certain improvements such as making it loop-free [36]. The DSDV is the foundation of many other distance vector routing protocols such as AODV that is addressed later. The distance vector routing is less robust than link state routing due to problems such as count to infinity and bouncing effect. Consequently, the proactive routing protocols prefer link state routing because additional route calculation of link state routing doesn't contribute to delay.

#### **2.4.1.2 OSPF**

Open Shortest Path First (OSPF) is the dominating link state routing protocol in wired IP networks [37]. Consequently, it is possible to adapt OSPF to the wireless networks in order to establish a seamless ubiquitous IP network. The main goal of OSPF is to quickly update the routing tables after the topology changes in a consistent way. OSPF uses Dijkstra's shortest path algorithm to construct the

forwarding tables based on the network link state database. OSPF is not suitable for the ad hoc wireless networks that have higher topology change, lower bandwidth, lower security and so forth than the wired networks.

### **2.4.1.3 OLSR**

Optimized Link State Routing (OLSR) is a proactive, link state routing protocol specially designed for ad hoc networks [38]. OLSR maintains Multipoint Relays (MPRs), which minimizes the control flooding by only declaring the links of neighbors within its MPRs instead of all links.

## **2.4.2 REACTIVE ROUTING PROTOCOL (RRP)**

The reactive (on-demand) routing protocols represent the true nature of ad hoc network, which is much more dynamic than infrastructured networks. Instead of periodically updating the routing information, the reactive routing protocols update routing information when a routing request is presented. Consequently reducing the control overhead, especially in high mobility networks where the periodical update will lead to significant useless overhead.

### **2.4.2.1 AODV**

Ad hoc On-demand Distance Vector Routing (AODV) is an improvement of the DSDV algorithm [42]. AODV minimizes the number of broadcasts by creating routes on-demand as opposed to DSDV that maintains the list of all the routes. The on-demand routing protocols suffer more from frequent broken source-to-destination links than table driven routing due to the delay caused by on-demand route recalculation. AODV avoids such additional delay by using distance vector routing.

It supports only symmetric links with two different phases:

- Route Discovery, Route Maintenance and Data forwarding.

In Route discovery the source node broadcasts a route request packet (RREQ) as in Fig 2.1. then the destination or an intermediate node with “fresh enough” route to the destination replies a route reply packet (RREP) as shown bellow, Fig 2.2.

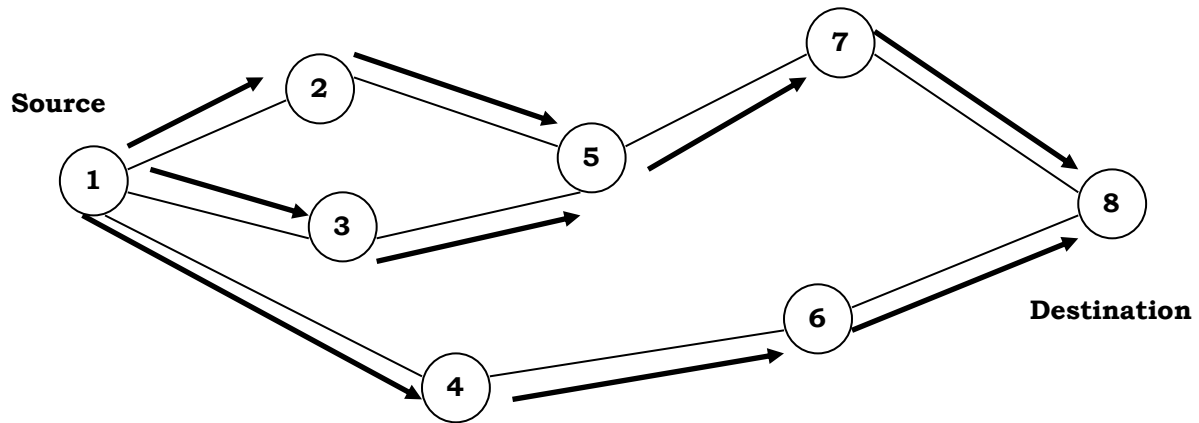


Fig 2.1. Propagation of RREQ in AODV [1]

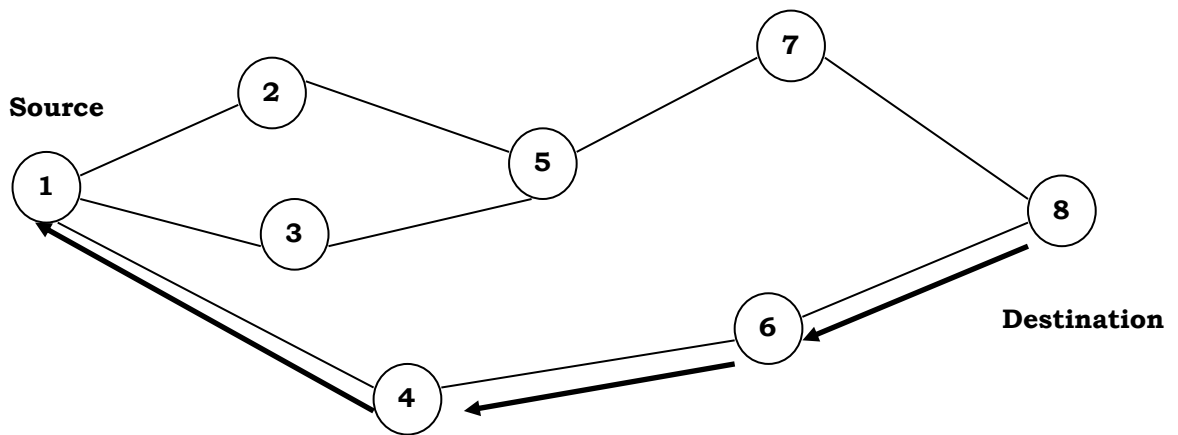


Fig 2.2. Path taken by the RREP in AODV [1]

### **2.4.2.2 DSR**

The key feature of DSR is the use of source routing, which means the sender knows the complete hop-by-hop route to the destination [41]. The node maintains route caches containing the source routes that it is aware of. Each node updates entries in the route cache as and when it learns about new routes. The data packets carry the source route in the packet headers. The delay and throughput penalties of DSR are mainly attributed to aggressive use of caching and lack of any mechanism to detect expired stale routes or to determine the freshness of routes when multiple choices are available. Aggressive caching, however, helps DSR at low loads and also keeps its routing load down. Several additional optimizations have been proposed and evaluated to be very effective.

### **2.4.2.3 Cluster Based Routing Protocol (CBRP)**

In CBRP the nodes of a wireless network are divided into several disjoint or overlapping clusters [27]. Each cluster elects one node as the so-called clusterhead. These special nodes are responsible for the routing process. Neighbors of clusterheads cannot be clusterheads as well. But clusterheads are able to communicate with each other by using gateway nodes. A gateway is a node that has two or more clusterheads as its neighbors or when the clusters are disjoint at least one clusterhead and another gateway node.

The routing process itself is performed as source routing by flooding the network with a route request message. Due to the clustered structure there will be less traffic, because route requests will only be passed between clusterheads.

## **2.4.3 HYBRID ROUTING PROTOCOLS**

The Ad Hoc network can use the hybrid routing protocols that have the advantage of both proactive and reactive routing protocols to balance the delay and control overhead (in terms of control packages) [1]. Hybrid routing protocols try to maximize the benefit of proactive routing and reactive routing by utilizing proactive routing in small networks (in order to reduce delay), and reactive

routing in large-scale networks (in order to reduce control overhead). Hybrid routing protocols are compared with proactive routing protocol OLSR. The results show the hybrid routing protocols can achieve the same performance as the OLSR and are simpler to maintain due to its scalable feature.

The difficulty of all hybrid routing protocols is how to organize the network according to network parameters. The common disadvantage of hybrid routing protocols is that the nodes that have high level topological information maintains more routing information, which leads to more memory and power consumption.

#### **2.4.3.1 ZRP**

The Zone Routing Protocol (ZRP) localizes the nodes into sub-networks (zones) [39]. Within each zone, proactive routing is adapted to speed up communication among neighbors. The inter-zone communication uses on-demand routing to reduce unnecessary communication. An improved mathematic model of topology management to organize the network as a forest, in which each tree is a zone, is introduced in. This algorithm guarantees overlap-free zones. Furthermore, the concept introduced in this algorithm also works with QoS control because the topology model is also an approach to estimate the link quality. An important issue of zone routing is to determine the size of the zone. An enhanced zone routing protocol, Independent Zone Routing (IZR), which allows adaptive and distributed reconfiguration of the optimized size of zone, is introduced in. Furthermore, the adaptive nature of the IZR enhances the scalability of the ad hoc network.

#### **2.4.3.2 LAR:**

Location Aided Routing (LAR) is another kind of hybrid routing protocol. LAR is a scalable routing protocol that uses landmarks, location and distance of the nodes to reduce the periodical update costs [40]. LAR is suitable for networks

with large number of nodes, which need to establish a hierarchy. This protocol is more complex than zone routing protocols due to the fact that the maintenance of hierarchical network is more difficult when determining the level of the nodes in the hierarchy.

Some research effort has been put on the adaptation of classic ad hoc routing protocols, such as DSR and AODV, to the scalable networks.

# Chapter Three: AODV Routing Protocol

## *3.1 Introduction*

The Ad-hoc On-Demand Distance Vector (AODV) [42] routing protocol is designed for use in ad-hoc mobile networks. AODV is a reactive protocol: the routes are created only when they are needed.

It uses traditional routing tables, one entry per destination, and sequence numbers to determine whether routing information is up-to-date and to prevent routing loops.

An important feature of AODV is the maintenance of time-based states in each node: a routing entry not recently used is expired. In case of a route is broken the neighbors can be notified.

Route discovery is based on query and reply cycles, and route information is stored in all intermediate nodes along the route in the form of route table entries. The following control packets are used: routing request message (RREQ) is broadcasted by a node requiring a route to another node, routing reply message (RREP) is unicasted back to the source of RREQ, and route error message (RERR) is sent to notify other nodes of the loss of the link. HELLO messages are used for detecting and monitoring links to neighbors.

AODV determines a route to a destination only when a node wants to send a packet to that destination. Routes are maintained as long as they are needed by the source. Sequence numbers ensure the freshness of routes and guarantee the loop-free routing.

### 3.2 Routing tables

Each routing table entry contains the following information [14], [42]:

- Destination
- Next hop
- Number of hops
- Destination sequence number
- Active neighbors for this route
- Expiration time for this route table entry.

Expiration time, also called lifetime, is reset each time the route has been used. The new expiration time is the sum of the current time and a parameter called active route timeout. This parameter, also called route caching timeout, is the time after which the route is considered as invalid, and so the nodes not lying on the route determined by RREPs delete their reverse entries.

### 3.3 Control messages

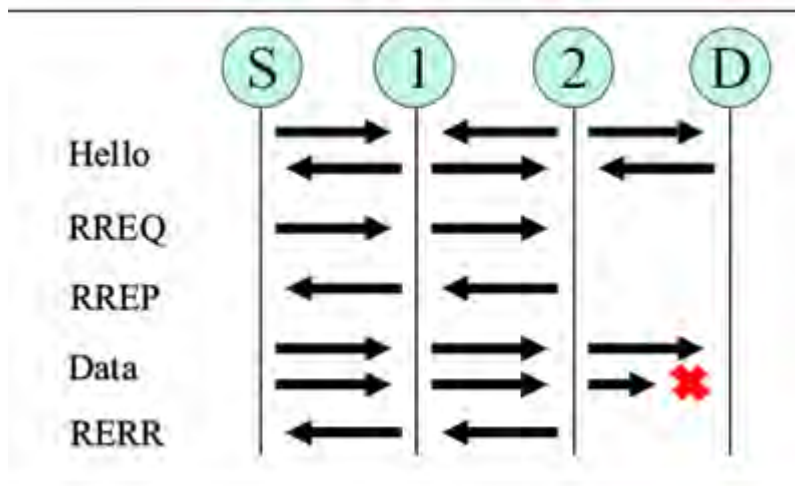


Fig 3.1 AODV Protocol Messaging [14]

#### 3.3.1 Routing request

When a route is not available for the destination, a route request packet (RREQ) is flooded throughout the network. The RREQ contains the following fields [3]:

The request ID is incremented each time the source node sends a new RREQ, so the pair (source address, request ID) identifies a RREQ uniquely. On receiving a RREQ message each node checks the source address and the request ID. If the node has already received a RREQ with the same pair of parameters the new RREQ packet will be discarded. Otherwise the RREQ will be either forwarded (broadcast) or replied (unicast) with a RREP message:

- if the node has no route entry for the destination, or it has one but this is no more an up-to-date route, the RREQ will be rebroadcasted with incremented hop count
- if the node has a route with a sequence number greater than or equal to that of RREQ, a RREP message will be generated and sent back to the source.

The number of RREQ messages that a node can send per second is limited. There is an optimization of AODV using an expanding ring (ESR) technique when flooding RREQ messages [14]. Every RREQ carries a time to live (TTL) value that specifies the number of times this message should be re-broadcasted. This value is set to a predefined value at the first transmission and increased at retransmissions. Retransmissions occur if no replies are received.

### **3.3.2 Routing reply**

If a node is the destination, or has a valid route to the destination, it unicasts a route reply message (RREP) back to the source. The reason one can unicast RREP back is that every node forwarding a RREQ message caches a route back to the source node.

### **3.3.3 Route error**

All nodes monitor their own neighborhood. When a node in an active route gets lost, a route error message (RERR) is generated to notify the other nodes on both sides of the link of the loss of this link.

### **3.3.4 HELLO messages**

Each node can get to know its neighborhood by using local broadcasts, so-called HELLO messages. Nodes neighbors are all the nodes that it can directly communicate with. Although AODV is a reactive protocol it uses these periodic HELLO messages to inform the neighbors that the link is still alive. The HELLO messages will never be forwarded because they are broadcasted with TTL = 1. When a node receives a HELLO message it refreshes the corresponding lifetime of the neighbor information in the routing table.

This local connectivity management should be distinguished from general topology management to optimize response time to local changes in the network.

### **3.4 Route discovery**

Route discovery process starts when a source node does not have routing information for a node to be communicated with. Route discovery is initiated by broadcasting a RREQ message. The route is established when a RREP message is received. A source node may receive multiple RREP messages with different routes. It then updates its routing entries if and only if the RREP has a greater sequence number, i.e. fresh information.

#### **3.4.1 Reverse path setup**

While transmitting RREQ messages through the network each node notes the reverse path to the source. When the destination node is found the RREP message will travel along this path, so no more broadcasts will be needed. For this purpose, the node on receiving RREQ packet from a neighbor records the address of this neighbor.

### **3.4.2 Forward path setup**

When a broadcast RREQ packet arrives at a node having a route to the destination, the reverse path will be used for sending a RREP message. While transmitting this RREP message the forward path is setting up. One can say that this forward path is reverse to the reverse path. As soon as the forward path is built the data transmission can be started.

Data packets waiting to be transmitted are buffered locally and transmitted in a FIFO-queue when a route is set up.

After a RREP was forwarded by a node, it can receive another RREP. This new RREP will be either discarded or forwarded, depending on its destination sequence number:

- if the new RREP has a greater destination sequence number, then the route should be updated, and RREP is forwarded
- if the destination sequence numbers in old and new RREPs are the same, but the new RREP has a smaller hop count, this new RREP should be preferred and forwarded
- Otherwise all later arriving RREPs will be discarded

### **3.5 Link breakage**

Because nodes can move link breakages can occurs. If a node does not receive a HELLO message from one of his neighbors for specific amount of time called HELLO interval, then

- the entry for that neighbor in the table will be set as invalid
- the RERR message will be generated to inform other nodes of this link breakage RERR messages inform all sources using a link when a failure occurs.

## **3.6 PROPERTIES OF AODV**

### **3.6.1 Merits of AODV**

The AODV routing protocol does not need any central administrative system to control the routing process. Reactive protocols like AODV tend to reduce the control traffic messages overhead at the cost of increased latency in finding new routes.

AODV reacts relatively fast to the topological changes in the network and updates only the nodes affected by these changes.

The AODV routing protocol saves storage place as well as energy. The destination node replies only once to the first request and ignores the rest. The routing table maintains at most one entry per destination.

If a node has to choose between two routes, the up-to-date route with a greater destination sequence number is always chosen. If routing table entry is not used recently, the entry is expired. A not valid route is deleted: the error packets reach all nodes using a failed link on its route to any destination.

### **3.6.2 Drawbacks of AODV**

It is possible that a valid route is expired. Determining of a reasonable expiry time is difficult, because the nodes are mobile, and sources' sending rates may differ widely and can change dynamically from node to node.

Moreover, AODV can gather only a very limited amount of routing information; route learning is limited only to the source of any routing packets being forwarded. This causes AODV to rely on a route discovery flood more often, which may carry significant network overhead. Uncontrolled flooding generates many redundant transmissions which may cause so-called broadcast storm problem.

The performance of the AODV protocol without any misbehaving nodes is poor in larger networks. The main difference between small and large networks is the

average path length. A long path is more vulnerable to link breakages and requires high control overhead for its maintenance.

Furthermore, as a size of a network grows, various performance metrics begin decreasing because of increasing administrative work, so-called administrative load.

AODV is vulnerable to various kinds of attacks, because it based on the assumption that all nodes will cooperate. Without this cooperation no route can be established and no packet can be forwarded. There are two main types of uncooperative nodes: malicious and selfish. Malicious nodes are either faulty and cannot follow the protocol, or are intentionally malicious and try to attack the network. Selfishness is noncooperation in certain network operations, i.e. dropping of packets which may affect the performance, but can save the battery power.

The following flow chart summarizes the action of an AODV node when processing an incoming message. HELLO messages are excluded from the diagram for brevity:

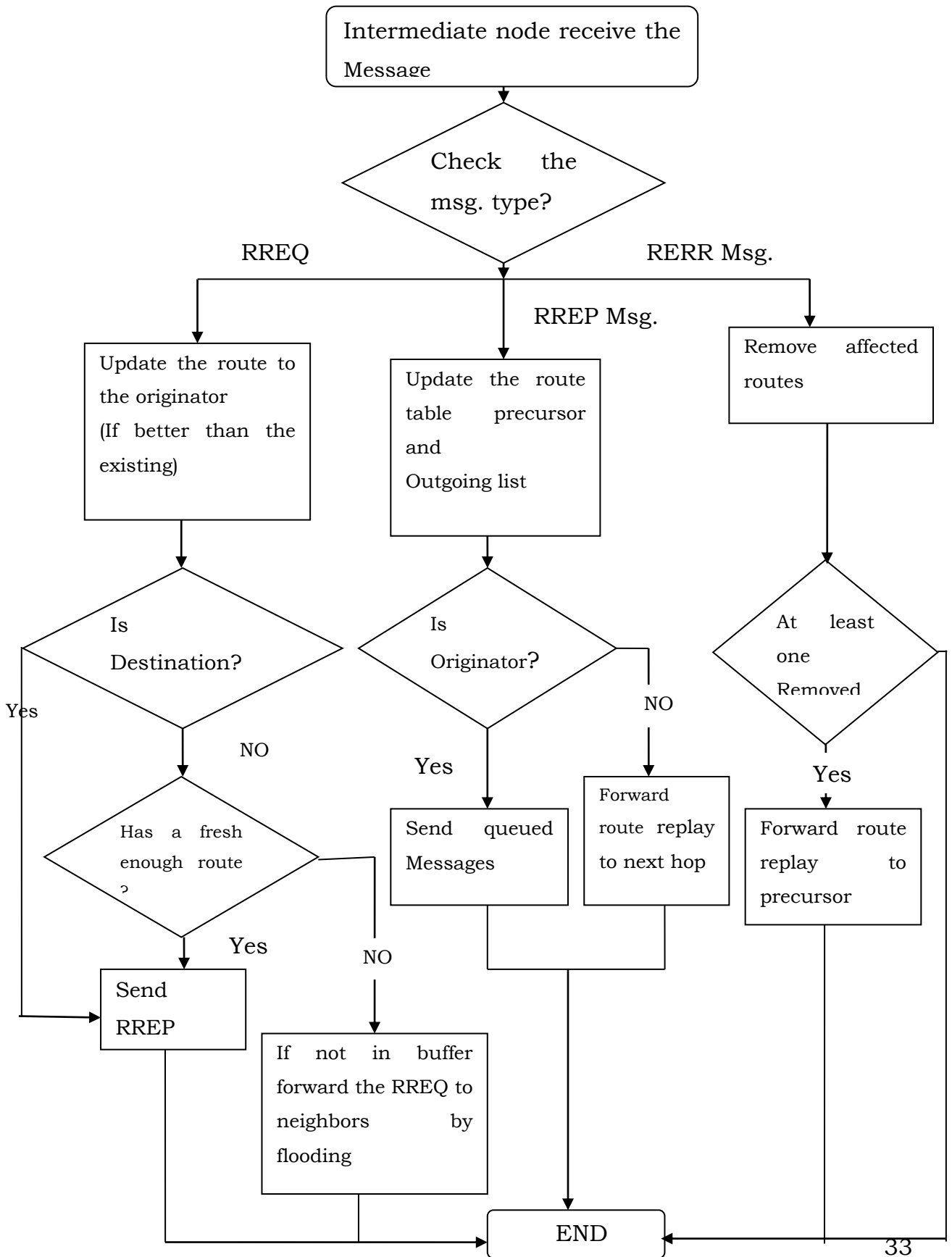


Fig 3.2 AODV  
flow chart

## Chapter Four: Clustering Algorithms

### 4.1 Introduction

The potentially large number of mobile nodes in a large-scale MANET makes routing extremely challenging. This is due to mobility of network members, limited battery energy, unpredictable behavior of radio channels, and time-varying bandwidth availability [19]. Table-driven routing suffers from excessive routing control overhead (proportional to the number of nodes) due to the necessity of maintaining a large routing table of the whole network and, therefore, is not scalable for a large and highly dynamic network topology. On the other hand, on-demand routing has high latency, caused by stale route cache information and frequent route reconstruction. As a hybrid of the previous two, cluster-based routing protocols maintain only partial topology information of the network. That is, each mobile node maintains a routing table only for the cluster of which it is a member; a backbone" is maintained by the cluster heads. A subset of nodes (cluster heads and gateway nodes) is designated to forward routing packets within and between clusters, thus reducing broadcast flooding overhead [19].

In a clustering scheme the mobile nodes in a MANET are divided into different virtual groups, and they are allocated geographically adjacent into the same cluster according to some rules with different behaviors for nodes included in a cluster from those excluded from the cluster. A typical cluster structure is shown in Fig 4.1. It can be seen that the nodes are divided into a number of virtual groups (with the dotted lines) based on certain rules. Under a cluster structure, mobile nodes may be assigned a different status or function, such as clusterhead, clustergateway, or clustermember. A clusterhead normally serves as a local coordinator for its cluster, performing intra-cluster transmission arrangement, data forwarding, and so on. A clustergateway is a non-clusterhead node with inter-cluster links, so it can access neighboring clusters and forward information between clusters. A clustermember is usually called an ordinary node, which is a non-clusterhead node without any inter-cluster links [16].

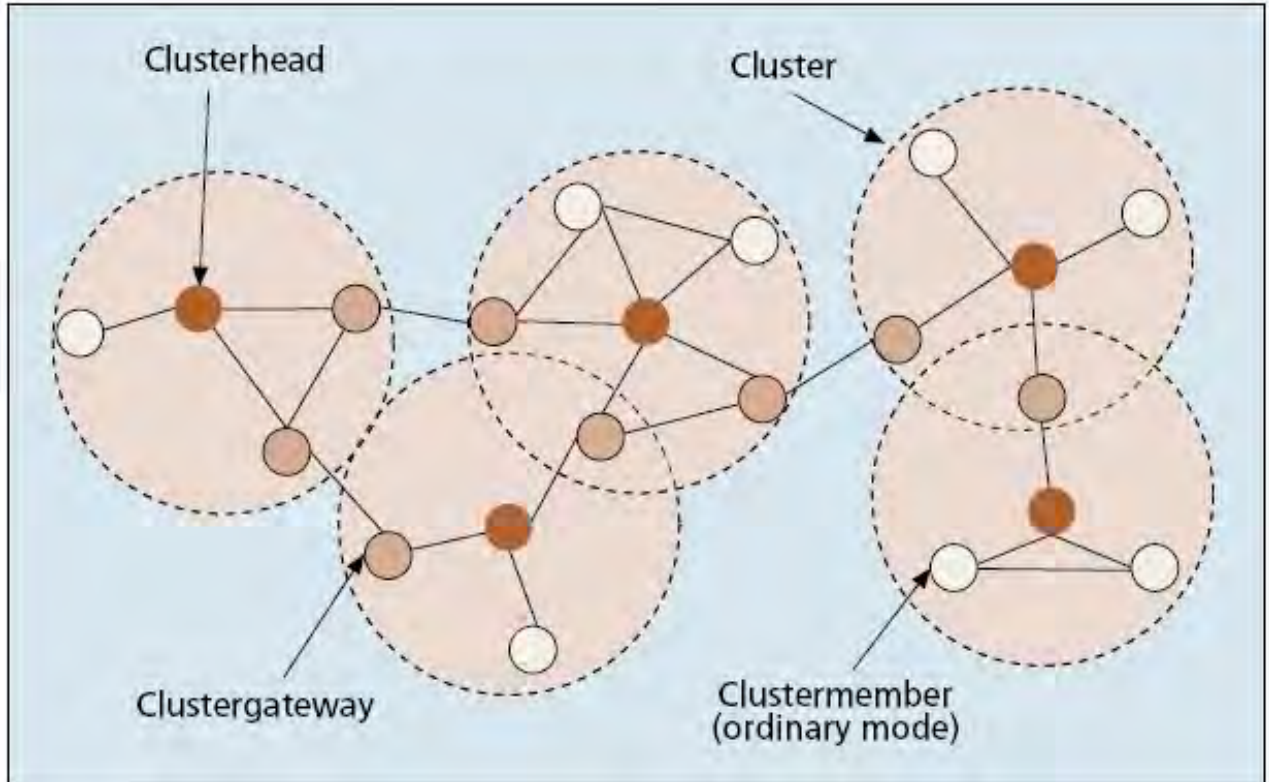


Fig 4.1. Cluster structure Illustration [16]

## ***4.2 Identifier-based clustering***

A unique ID is assigned to each node. Nodes know the Id of its neighbors and clusterhead is chosen following some certain rules as given below.

### **4.2.1 Lowest ID cluster algorithm (LIC)**

is an algorithm in which a node with the minimum ID is chosen as a clusterhead [15]. Thus, the ids of the neighbors of the clusterhead will be higher than that of the clustrehead. A node is called a gateway if it lies within the transmission range of two or more clusterheads. Gateway nodes are generally used for routing between clusters. Each node is assigned a distinct id. Periodically, the node broadcasts the list of nodes that it can hear (including itself).

- A node which only hears nodes with id higher than itself is a clusterhead.
- The lowest-id node that a node hears is its clusterhead, unless the lowest-id specifically gives up its role as a cluststerhead (deferring to a yet lower id node).
- A node which can hear two or more clusterheads is a gateway.
- Otherwise, a node is an ordinary node.

The lowest-ID scheme concerns only with the lowest node ids which are arbitrarily assigned numbers without considering any other qualifications of a node for election as a clusterhead. Since the node ids do not change with time, those with smaller ids are more likely to become clusterheads than nodes with larger ids. Thus, drawback of lowest if algorithm is that certain nodes are prone to power drainage due to serving as clusterheads for longer periods of time.

### 4.2.2 Max-Min d-cluster formation algorithm

Generalizes the cluster definition to a collection of nodes that are up to  $d$ -hops away from a clusterhead [15]. Due to the large member of nodes involved, it is desirable to let the nodes operate asynchronously. The clock synchronization overhead is avoided, providing additional processing savings. Furthermore the number of messages sent from each node is limited to a multiple of  $d$  the maximum number of hops away from the nearest clusterhead, rather than  $n$  the number of nodes in the network. This guarantees a good controlled message complexity for the algorithm. Additionally, because  $d$  is an input value to the heuristic, there is control over the number of clusterheads elected or the density of clusterheads in the network. The amount of resources needed at each node is minimal, consisting of four simple rules and two data structures that maintain node information over  $2d$  rounds of communication. Nodes are candidates to be clusterheads based on their node id rather than their degree connectivity. As the network topology changes slightly the node's degree of connectivity is much more likely to change than the node's id relative to its neighboring nodes. if a node  $A$  is the largest in the  $d$ -neighborhood of another node  $B$  then node  $A$ ,  $A$  will be elected a clusterhead, even though node  $A$  may not be the largest in its  $d$ -neighborhood. This provides a smooth exchange of clusterheads rather than an erratic exchange. This method minimizes the amount of data that must be passed from an outgoing clusterhead to a new clusterhead when there is an exchange.

### 4.3 Connectivity-based clustering

Highest connectivity clustering algorithm (HCC) [23] is one of connectivity based clustering algorithm in which the degree of a node is computed based on its distance from others. Each node broadcasts its id to the nodes that are within its transmission range. The node with maximum number of neighbors (i.e.,

maximum degree) is chosen as a clusterhead. The neighbors of a clusterhead become members of that cluster and can no longer participate in the election process. Since no clusterheads are directly linked, only one clusterhead is allowed per cluster. Any two in a cluster are at most two hops away since the clusterhead is directly linked to each of its neighbors in the cluster. Basically, each node either becomes a clusterhead or remains an ordinary node.

This system has a low rate of clusterhead change but the throughput is low. Typically, each cluster is assigned some resources which are shared among the members of that cluster. As the number of nodes in a cluster is increased, the throughput drops. The reaffiliation count of nodes is high due to node movements and as a result, the highest-degree node (the current clusterhead) may not be re-elected to be a cluster head even if it loses one neighbor. All these drawbacks occur because this approach does not have any restriction on the upper bound on the number of nodes in a cluster.

#### *4.4 Mobility-aware clustering*

Mobility is a prominent characteristic of MANETs, and is the main factor affecting topology change and route invalidation. Some believe that it is important to take the mobility metric into account in cluster construction in order to form a stable cluster structure and decrease its influence on cluster topology. Mobility-aware clustering indicates that the cluster architecture is determined by the mobility behavior of mobile nodes. The idea is that by grouping mobile terminals with similar speed into the same cluster, the intra-cluster links can become more tightly connected. Then the re-affiliation and re-clustering rate can be naturally decreased [16].

Mobility-based d-hop clustering algorithm presented in [21] Partitions an ad hoc network into d-hop clusters based on mobility metric. The objective of forming d-

hop clusters is to make the cluster diameter more flexible. This algorithm is based on mobility metric and the diameter of a cluster is adaptable with respect to node mobility. This clustering algorithm assumes that each node can measure its received signal strength. In this manner, a node can measure its received signal strength. In this manner, a node can estimate its received signal strength. In this manner, a node can estimate its distance from its neighbors. Strong received signal strength implies closeness between two nodes. This algorithm requires the calculation of five terms: the estimated distance between nodes, the relative mobility between nodes, the variation of estimated distance over time, the local stability, and the estimated mean distance. Relative mobility corresponds to the difference of the estimated distance of one node with respect to another, at two nodes move away from each other or if they become closer.

The variation of estimated distances between two nodes is computed instead of calculating physical distance between two nodes. This is because physical distance between two nodes is not a precise measure of closeness. For instance, if a node runs out of energy it will transmit packets at low power acting as a distanced node from its physically close neighbor. The variation of estimated distance and the relative mobility between nodes are used to calculate the local stability. Local stability is computed in order to select some nodes as clusterheads. A node may become a clusterhead if it is found to be the most stable node among its neighborhood. Thus, the clusterhead will be the node with the lowest value of local stability among its neighbors.

## *4.6 Power-Aware Clustering*

Mobile nodes in a MANET normally depend on battery power supply during operation; hence the energy limitation poses a severe challenge for network performance. A MANET should strive to reduce its energy consumption greedily in order to prolong the network lifespan. Also, a clusterhead bears extra work

compared with ordinary members, and it more likely “dies” early because of excessive energy consumption. The lack of mobile nodes due to energy depletion may cause network partition and communication interruption [15]. Hence, it is also important to balance the energy consumption among mobile nodes to avoid node failure, especially when some mobile nodes bear special tasks or the network density is comparatively sparse.

Load balancing clustering (LBC) [22] is one of Power aware clustering algorithm in which each mobile node has a variable, virtual ID (VID), and the value of VID is set as its ID number at first. Initially, mobile nodes with the highest IDs in their local area win the clusterhead role. LBC limits the maximum time units that a node can serve as a clusterhead continuously, so when a clusterhead exhausts its duration budget (Max\_Count); it resets its VID to 0 and becomes a non-clusterhead node.

When two clusterheads move into the reach range, the one with higher VID wins the clusterhead role. Each non-clusterhead node keeps a circular queue for its VID and shifts the VID value by one every time unit in one direction. Thus, when a clusterhead resigns, a non-clusterhead with the largest VID value in the neighborhood can resume the clusterhead function.

LBC tries to avoid possible node failure due to energy depletion caused by excessively shouldering the clusterhead role. When a mobile node resigns its clusterhead status because of the expiration of its duration budget, another mobile node with the highest VID in the local area is chosen to resume the clusterhead function. The newly chosen mobile node is the one whose previous total clusterhead serving time is the shortest in its neighborhood, and this should guarantee good energy level for being a new clusterhead. However, this kind of new clusterhead selection may introduce ripple effect of re-clustering over the whole network without considering the network topology. In addition,

the clusterhead re-election may require time synchronization of the VID value shift among different mobile nodes. Otherwise, the VID information may not be accurate enough to select the most suitable node to serve as new clusterhead. In addition, the clusterhead serving time alone may not be able to promise a good indication of energy consumption of a mobile node.

#### *4.7 Combined-weight based clustering*

Combined-metrics-based clustering takes a number of metrics into account for cluster configuration, including node degree, residual energy capacity, moving speed, and so on like in [5], [6] and [20]. For example the combined weight based clustering at [5] use Battery Remaining, Number of Neighbors, Number of Members, and Stability in order to calculate the node's score to form clusters. This category aims at electing the most suitable clusterhead in a local area, and does not give preference to mobile nodes with certain attributes, such as lowest ID or highest node degree. One advantage of this clustering scheme is that it can flexibly adjust the weighting factors for each metric to adjust to different scenarios. For example, in a system where battery energy is more important, the weighting factor associated with energy capacity can be set higher. However, not all of these parameters are always available and accurate, and the information inaccuracy may affect clustering performance.

For example, Weighted clustering algorithm (WCA) [20] is based on the use of a combined weight metric that takes into account several system parameters like the ideal node-degree, transmission power, mobility and the battery power of the nodes. Depending on specific applications, any or all of these parameters can be used in the metric to elect the clusterheads.

## **CHAPTER Five: Design of CBAODV routing protocol**

As mentioned earlier in the text, Network Simulator 2 (NS2) [24] is used as the simulation tool in this project. NS was chosen as the simulator partly because of the range of features it provides and partly because it has an open source code that can be modified and extended. In this project, the AODV routing protocol has been modified in NS 2 so that the modified performance compared with the original one.

There are several different versions of NS and at the current time the latest version is ns-2.34. Version Ns-2.29 has been used in this study because of its full documentation.

In this Chapter, first the NS-2 simulation tool and then the design and implementation details of the Energy aware clustering algorithm for AODV routing protocol to get CBAODV using NS-2.29 will be presented.

### *5.1 Network Simulator (NS)*

Network Simulator (NS) is an object-oriented, discrete event simulator for networking research [25]. NS provides substantial support for simulation of TCP, routing and multicast protocols over wired and wireless networks. The simulator is a result of an on-going effort of research and development. Even though there is a considerable confidence in NS, it is not a polished and finished product. Bugs are being discovered and corrected continuously.

NS is written in C++, with an OTcl interpreter as a command and configuration interface. The C++ part, which is fast to run but slower to change, is used for detailed protocol implementation.

The OTcl part, on the other hand, which runs much slower but can be changed very quickly, is used for simulation configuration. One of the advantages of this split-language programming approach is that it allows for fast generation of large scenarios. To simply use the simulator it is sufficient to know OTcl. On the other hand one disadvantage is that modifying and extending the simulator requires programming and debugging in both languages simultaneously.

### 5.1.1 Marc Greis' Tutorial

The very first thing to do for a new user of NS is to read Marc Greis' tutorial. There is a link to this tutorial on the web page of NS which can be found at [24]. The purpose of this tutorial is to make it easier for new NS users to use NS and NAM, to create their own simulation scenarios for these tools and to eventually add new functionality to NS.

### 5.1.2 NS by Example

On the web page of NS, there is a link to another tutorial for NS. This tutorial has been written by Jae Chung and Mark Claypool and its purpose is to give new users some basic idea of how the simulator works, how to setup simulation network codes, how to create new network components and so on. In particular, it explains the linkage between the two languages used in NS, namely C++ and OTcl. One can find some very good examples and brief explanations in this tutorial, which is the second tutorial to study after reading Marc Greis' tutorial.

### 5.1.3 NS Manual, NS Search, and NS Mailing List

In the NS Manual [25] one can find the answer to many questions. A link to this Manual can be found on the web page of NS. However, if no answer can be found in the Manual the NS mailing list archives should be searched. The archive keeps all previous emails sent to the ns-users mailing list. The ns-users mailing list should be used if an answer still (after looking in the Manual and searching the archives) has not been found. Everyone that has subscribed will receive this email and will hopefully reply.

### 5.2 Network Animator (NAM)

Network Animator (NAM) [24], [25] is an animation tool for viewing network simulation traces and real world packet traces. It supports topology layout, packet level animation and various data inspection tools.

Before starting to use NAM, a trace file needs to be created. This trace file is usually generated by NS. It contains topology information. E.g. nodes and links, as well as packet traces. During an Implementation of a Routing Protocols for Manet's simulation, the user can produce topology configurations, layout information and packet traces using tracing events in NS.

Once the trace file is generated, NAM can be used to animate it. Upon startup, NAM will read the trace file, create topology, pop up a window, do layout if necessary and then pause at time 0. Through its user interface, NAM provides control over many aspects of animation.

Although the NAM software contains bugs, as do the NS software, it works fine most of the times and causes only little trouble. NAM is an excellent first step to check that the scenario works as expected. NS and NAM can also be used together for educational purpose and to easily demonstrate different networking issues.

### 5.3 Design of CBAODV routing protocol

Design of CBAODV constitutes design of Cluster Formation, Adjacent Cluster Discovery and Routing Consideration. Details of Cluster Formation and Adjacent Cluster Discovery have been taken from reference [25]. But the design has been modified to be suitable to run on AODV. The routing consideration has been designed in this thesis work.

A cluster is identified by its Cluster Head ID. Clusters are either overlapping or disjoint. Each node in the network knows its corresponding Cluster Head(s) and therefore knows which cluster(s) it belongs to.

In NS-2, **aodv.cc** and **aodv.h** files implement and define the AODV routing protocol. **aodv.cc** is organized in to functions; the functions are designed based on their task on routing activity. This thesis work start by modifying these program files of AODV protocol to achieve CBAODV: some of the design and implementation procedure of CBAODV follows that of CBRP as stated in [26] and [27].

This thesis starts implementation of CBAODV by incorporating the computation of the existing energy of the node, selecting cluster heads based on the energy information available in neighbor tables (`NeighborTableCFormation()` function), and by putting contention solving mechanism when two cluster heads come to each other (`NeighborTableCContention()` function) functions in **aodv.cc**. *receiveRequest*, *sendRequest*, *receiveHello* and *sendHello* functions of **aodv.cc** are also modified to integrate energy aware clustering algorithm with the AODV.

The flat AODV hello packet uses header of RREP packet with time to leave (TTL) 1, in this thesis **hello** message made to have its own header and message format which contains the node status, clusterhead ID and the residual energy and also it hold neighbor and adjacent cluster tables as an extension as shown in Fig 5.1. . The neighbor table extension to the hello message is shown in Fig 5.2.

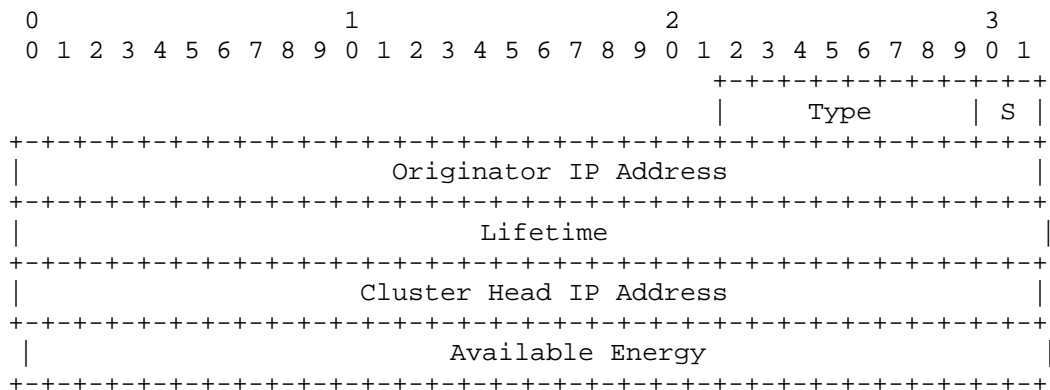


Fig. 5.1 Modified HELLO message format

Where:

*Type*            5 // Type of AODV packet, 5 for Hello packet

*S (Status)*    The current status of the sender.

0 --- Undecided (C\_UNDECIDED)

1 --- Cluster Head (C\_HEAD)

2 --- Cluster Member (C\_MEMBER)

*Originator IP Address* The IP address of the node which originated the HELLO packet

*Lifetime*    ALLOWED\_HELLO\_LOSS \* HELLO\_INTERVAL

*Available Energy* the node residual energy at the time the hello message sent

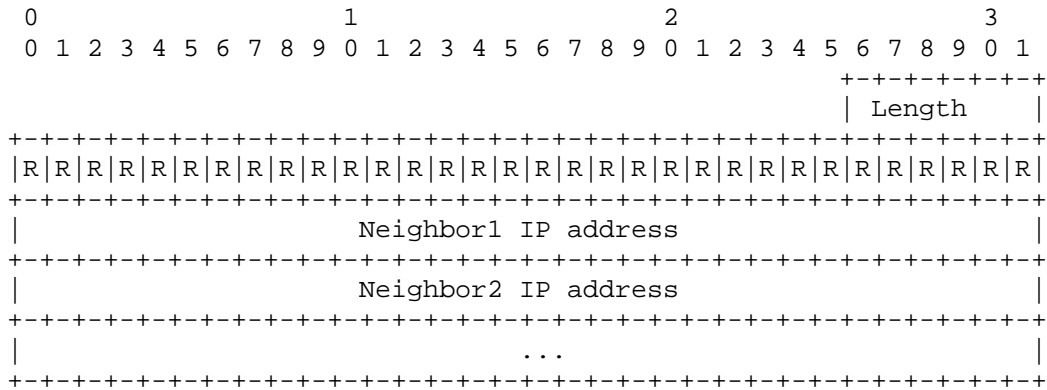


Fig. 5.2 Neighbor Table Extension in a HELLO message

Where:

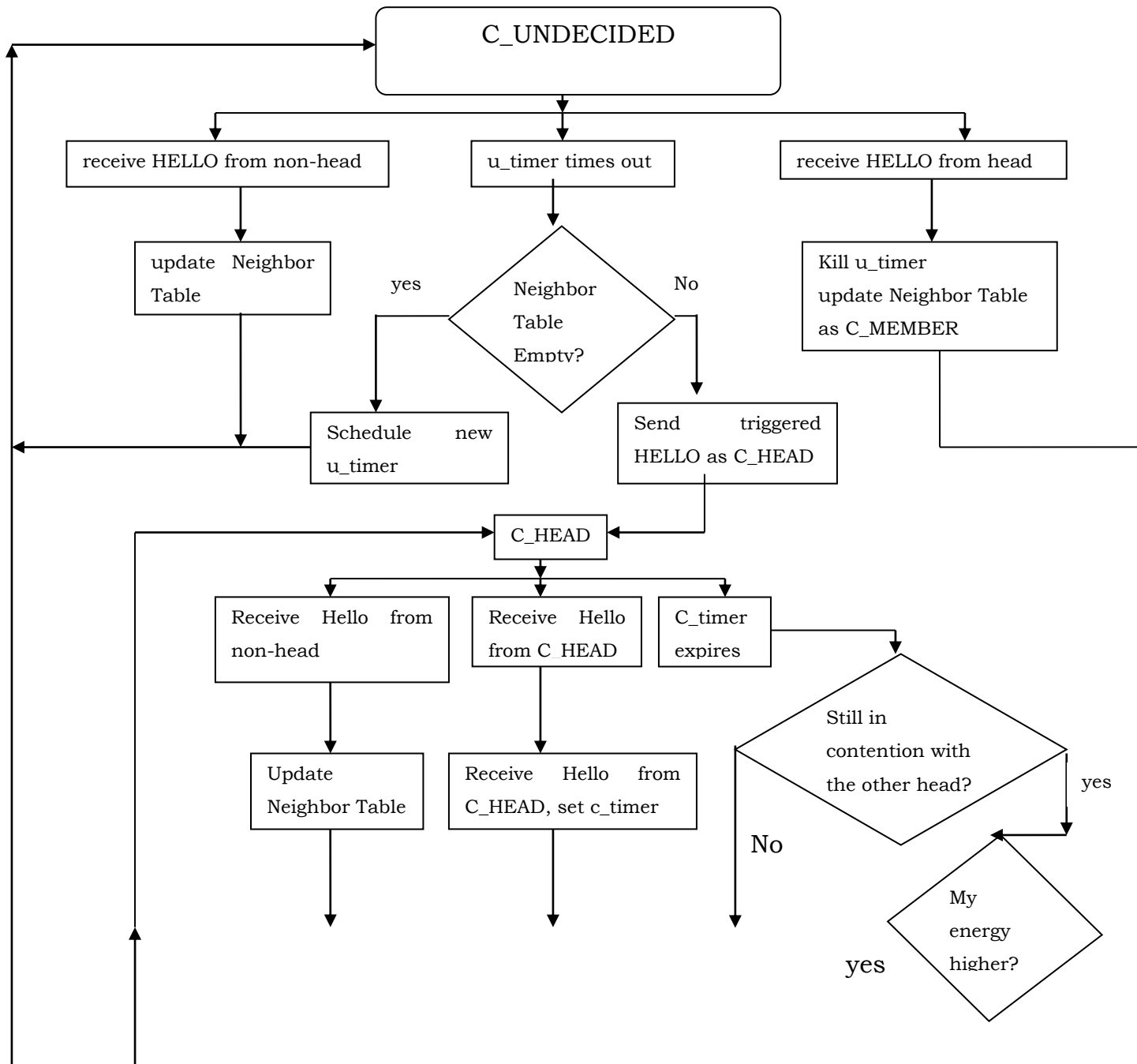
- Length*      The number of neighbors listed.
- R*              Role of the corresponding neighbor of the sender.
  - 0 --- Non-Cluster-Head (C\_MEMBER or C\_UNDECIDED)
  - 1 --- Cluster Head (C\_HEAD)

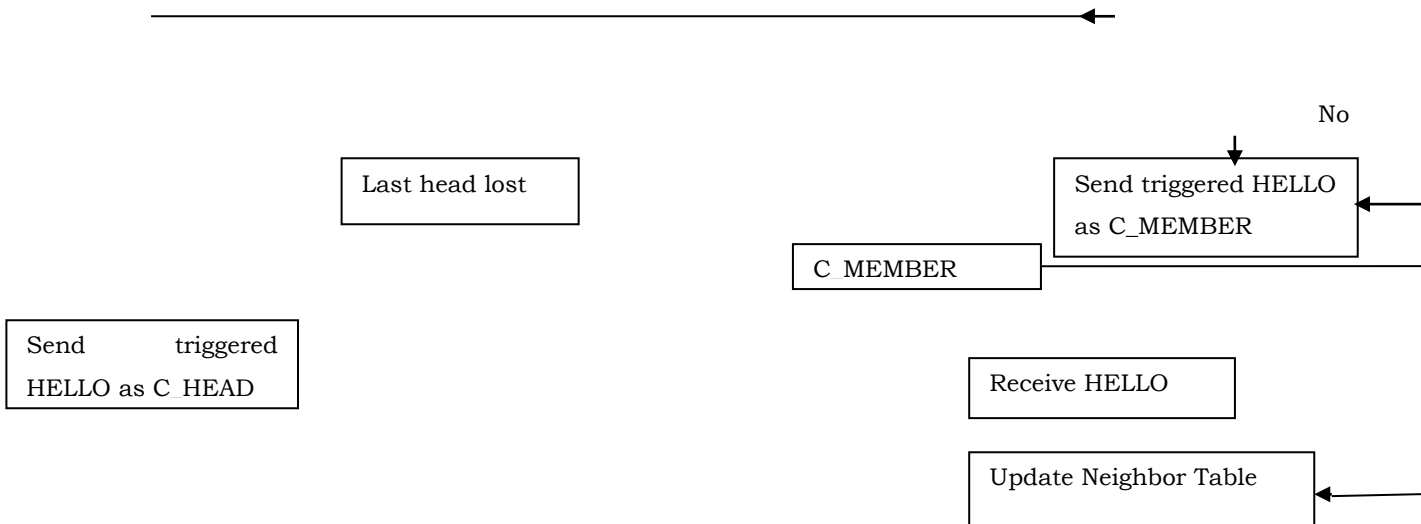
### 5.3.2 Cluster Formation

The design of CBAODV considers the available energy in each node, in which the node with a highest available energy among its neighbors is elected as the Cluster Head. In case of a tie, the node with the lowest ID is chosen. Initially, each node broadcasts a Hello message to notify its presence to the neighbors. A Hello message contains its ID and residual energy and the nodes neighbor with their id and status. Each node builds its neighbor and adjacent cluster table based on the Hello messages received.

A node will be either in cluster member (*C\_MEMBER*), cluster head (*C\_HEAD*), or cluster undecided (*C\_UNDECIDED*) state. "*Undecided*" means that a node is still in search of its host cluster [25]. All nodes wake up in the Undecided state. We will refer to a node in the undecided state as an undecided node hereafter. A

state transition diagram of cluster formation is shown at Fig 5.3 which is taken and modified from [25].





As shown in Fig. 5.3, when a node enters the cluster undecided state, its  $u\_timer$  goes off in an undecided period ( $C\_UNDECIDED\_TD$ ) seconds and broadcast a *HELLO* message whenever it enters the cluster undecided ( $C\_UNDECIDED$ ) state. When a cluster head receives a *HELLO* message from an Undecided Node, it will send out a triggered *HELLO* message immediately. If an undecided node receives a *HELLO* message from a Cluster Head, it aborts its  $u\_timer$  and sets its own status to cluster member ( $C\_MEMBER$ ). When the  $u\_timer$  times out, then it re-enters the Undecided state; otherwise it elects itself as a Cluster Head. The new Cluster Head will change the first field in its subsequently broadcast *HELLO* messages from  $C\_UNDECIDED$  to cluster head ( $C\_HEAD$ ) thereafter.

Note that a member node may hear from several cluster heads and therefore have several host clusters; its host cluster heads are implicitly listed in the *HELLO* messages it broadcasts.

As clusters are identified by their respective cluster heads, for stable performance the cluster heads change need to be as infrequently as possible. The following rules for changing cluster head, as described in [26] and [28].

1. A non-cluster head never challenges the status of an existing cluster head, i.e. if X is a non-cluster head node with link to cluster head Y, X does not become a cluster head even if it has a higher energy than Y's.

2. When two cluster heads move next to each other over an extended period of time (for *CONTENTION\_PERIOD* seconds), then only will one of them lose its role of cluster head.

As a result, whenever a cluster head hears *HELLO* messages from another cluster head, it sets *c\_timer* to expire in *CONTENTION\_PERIOD* seconds. When *c\_timer* expires, it will check if it is still in contention with the other cluster head, by checking if the other cluster head is still in its neighbor table. If so, it compares its own energy with that of the other cluster head's. The one with a higher energy will continue to act as cluster head. The one with a lower energy gives up its role as cluster head and changes from *C\_HEAD* to *C\_MEMBER* in its subsequent *HELLO* messages. This might trigger reorganization of other clusters.

Whenever a member node's last cluster head entry times out, this node checks among its neighbors if it has the highest energy, if so it changes its state to *C\_HEAD* and sends out a triggered *HELLO*, otherwise it goes to *C\_UNDECIDED* state.

### 5.3.3 Adjacent Cluster Discovery [26]

The goal of Adjacent Cluster Discovery is for a cluster to discover all its adjacent clusters. For this purpose, each node keeps a Cluster Adjacency Table (CAT) that records information about all its neighboring cluster heads.

Note that for member nodes, neighboring cluster heads are always two hops away and can be discovered by checking the received *HELLO* messages.

For a cluster head, its neighboring cluster heads could be 2 or 3 hops away (see Fig 4.1). Using the *HELLO* messages alone, a cluster head is able to discover the cluster heads 2 hops away. However, it has to rely on its member nodes' Cluster Adjacency Table (CAT) to discover those neighboring cluster heads that are 3 hops away.

The format of CAT at each node is as follows in Fig 5.4:

```

+-----+ +-----+ +-----+
|Adj. Cluster1  |->|gateway1  |->|gateway2  |
+-----+ +-----+ +-----+
|Adj. Cluster2  |->|gateway1  |->|gateway2  |
+-----+ +-----+ +-----+
|...
+-----+

```

Fig 5.4 Format of CAT in each node

Gateway field (*gateway1/2*) contains the ID of the gateway node through which the neighboring cluster head (*Adj.Cluster1*) could be reached. Gateway node is the member node of the adjacent cluster. Note that there may be several gateway nodes leading towards a particular adjacent cluster.

This table is updated by the periodic HELLO messages a node hears. A member node finds out cluster heads that are exactly 2 hops away and records them in its CAT. In particular, whenever a node A receives a HELLO message from B, it scans through the message's list of entries. If there is a cluster head C (i.e. B is a member node of cluster C) and, furthermore, C is not already A's host cluster, A adds an entry in CAT for adjacent cluster C with gateway node B and link status specifying status of link between A and B.

In order for cluster heads to gain information on their adjacent clusters that are 3 hops away, each member node broadcasts its summarized CAT as a Cluster Adjacency Extension to the HELLO message as shown in Fig 5.5. (Only member node will send out this information, this is not the case with cluster heads.)

The format of the Cluster Adjacency Extension to HELLO message is shown below:

```

          0                1                2                3
          0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
          +-----+ +-----+ +-----+ +-----+
          |                                     | Length |
          +-----+ +-----+ +-----+ +-----+
          |           Adj. Cluster Head1  IP address           |
          +-----+ +-----+ +-----+ +-----+
          |           Adj. Cluster Head2  IP address           |
          +-----+ +-----+ +-----+ +-----+
          |                                     |
          +-----+ +-----+ +-----+ +-----+

```



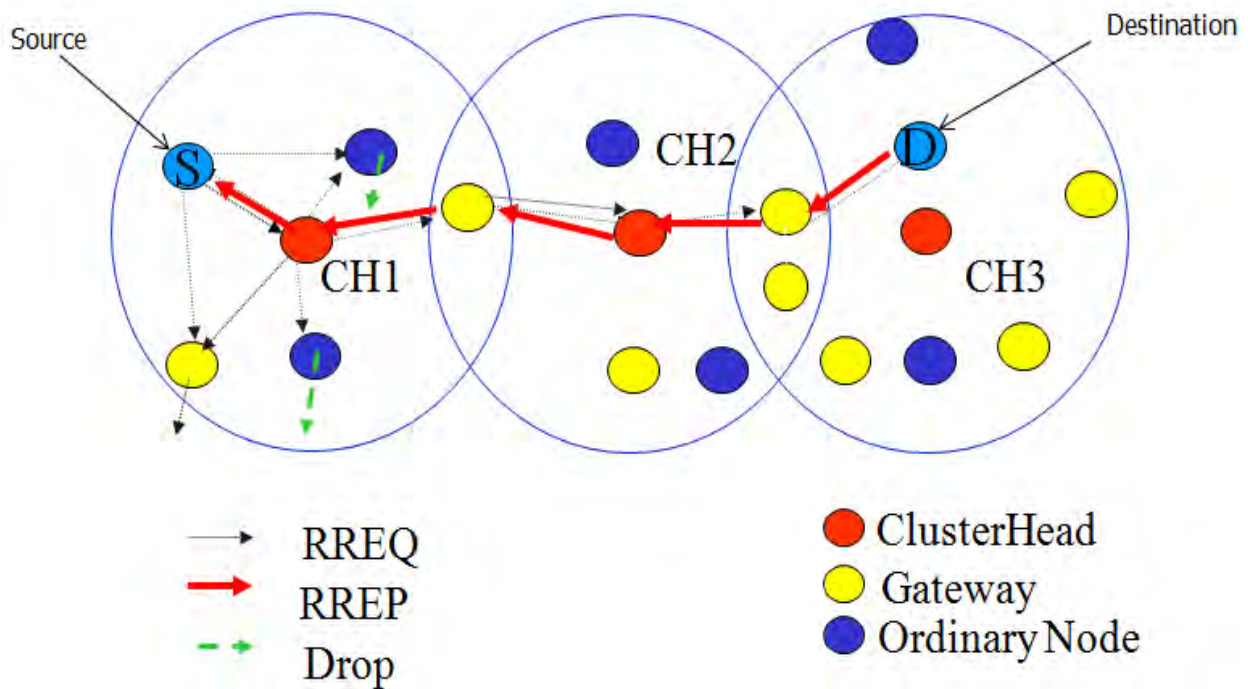


Fig. 5.6 CBAODV Route Establishment.

Route Discovery is the mechanism whereby a node S wishing to send a packet to a destination D obtains route to D. The way S finds a route (or multiple routes) to D is also done by flooding, however, because of the clustering approach the number of times nodes are disturbed are much less in general. As shown in Fig 5.6, in Route Discovery, only cluster heads are flooded with Route Request Packets (RREQ) in search for a route.

Fig 5.7 show a flow diagram of an intermediate node's operation when it receives RREQ.

To perform Route Discovery to D, the source node S sends out an RREQ, with the target node address field set to D, the operation is shown in the following piece of codes.

1. Whenever a member node M receives an RREQ,
 

```

      if D is its neighbour, RREQ is just uni-cast to D.
      else
        if M a Gateway Node to a Head[x],
      
```

```

    uni-cast the RREQ to Cluster Head[x].
else
    discard the RREQ.

```

2. Whenever a cluster head C receives an RREQ,

```

    if D is its neighbour or is 2-hop away
        uni-cast RREQ to D.
    else
        broadcast RREQ

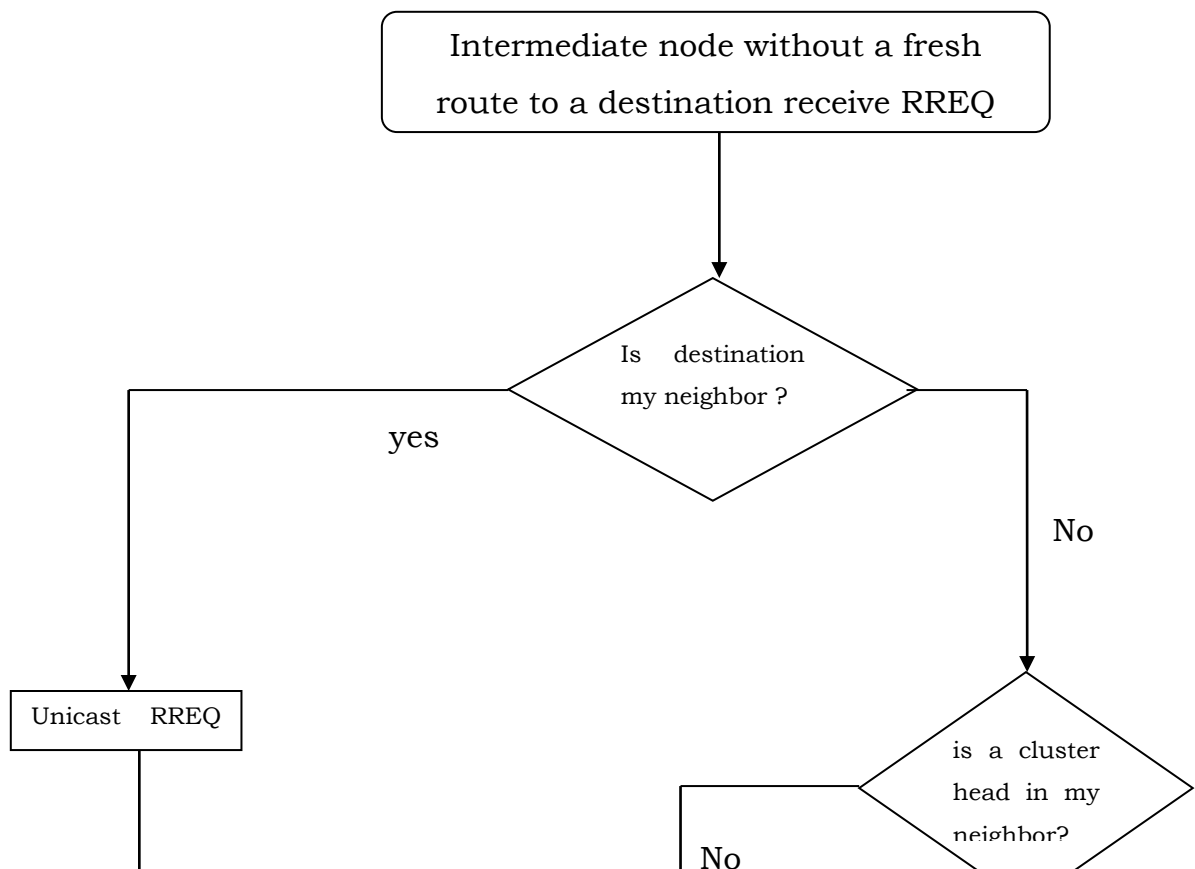
```

In CBAODV, the RREQ will always follow a route with the following pattern to reach destination D:

S,CH1,G1,CH2,G2,CH3 ..... D

When the target of the Request, node D, receives the RREQ, D sends out an RREP packet to S as a reply using the routing table which is the same as the flat AODV. If a source does not receive any RREP after sending out RREQ for certain period of time, it goes into exponential back off before re-sending RREQ.

When a node wishes to send a packet, it always examines its own Route table first before performing Route Discovery.



*Fig 5.7,  
Receive  
Request  
(recvRequest())  
operation at  
intermediate  
node*

## *5.4 Design of CAODV*

Clustered AODV (CAODV) routing protocol is a preliminary version of CBAODV in which the lowest ID clustering algorithm [23] is used for cluster formation. Adjacent Cluster Discovery and routing considerations are same as that of CBAODV as explained above.

The need for CAODV's performance comparison with AODV, CBAODV, DSR and CBRP is to show the advantage of energy based clustering algorithm and also AODV and DSR are flat routing protocols, in the meantime CAODV and CBRP are clustered version of AODV and DSR respectively which used lowest ID clustering algorithm for cluster formation.

## **Chapter Six: Simulation, Results and Result Analysis**

In this Chapter, we present the performance evaluation of CBAODV. First simulation scenarios and model will be discussed; the network topology and movement of nodes with traffic models will be discussed next. Then detail simulation results and analysis will be presented.

### ***6.1 Simulation Scenarios and Model***

The NS-2 simulator supports for simulating wireless networks consists of different network components including physical, data link, and medium access control (MAC) layer models. From channel type, a wireless channel model with a 250m-transmission range has been chosen. IEEE 802.11 for wireless networks is used as the MAC layer protocol. All packets (both data and routing) sent by the routing layer are queued at the interface queue until the MAC layer can transmit them. The interface queue has a maximum size of 50 packets and is worked as a priority queue. There are two priorities each served in First In First out (FIFO) manner, which means Routing packets have higher priority than data

packets. The routing protocols that have been implemented at the network layer are AODV, CAODV, CBAODV, DSR and CBRP.

## 6.2 The traffic and mobility models

The model of mobility used is Random Waypoint model [29]. In Random Waypoint, the direction and the speed of move are chosen according to a fixed interval. The mobility models were created for the simulations using 50, 100 and 150 nodes, and this model was set in such a way that first all the nodes were provided with initial location in the given rectangular topography field. The field configuration used is: 1000 m x 1000 m. Then all the nodes move within their boundary by setting their final destination and the speed that each node move with. The nodes move with maximum speed of 1m/s, 5m/s 10m/s, and 20m/s and 20 and 30 traffic connections have been used in the simulation. All the simulations are run for 300 simulated seconds. In this thesis work the flat AODV uses HELLO messages for determining local connectivity [35].

Number of node	50, 100, 150
Routing Protocol	AODV, CAODV, CBAODV, DSR, CBRP
Simulation time	300 sec
Number of connection	20, 30
Maximum Speed	1m/s, 5m/s, 10m/s, 20m/s
Network Area	1000mx1000m
Traffic Type	CBR
Agent	UDP
Packet Size	512bytes

Pause time	30 sec
Node initial Energy	1000J

Table 6.1: Simulation parameters value

### 6.3 Evaluation Metrics

The metrics used to evaluate the performance of the proposed routing protocols are average packet delivery ratio, average Normalized Routing Load and average end-to-end delay.

#### 6.3.1 Average Packet Delivery Ratio (APDR)

Packet Delivery Ratio (PDR) is the ratio of total number of packets successfully received by the destination nodes to the number of packets sent by the source nodes. Mathematically PDR expressed as:

$$PDR = \frac{\text{Number\_of\_Packets\_Correctly\_Received\_by\_Destination}}{\text{Number\_of\_Packets\_Sent\_By\_Source}} \quad (6.1)$$

And Average PDR (APDR) is the ratio of the sum of all (PDR) of the total number of simulation scenario (with fixed parameters) to the total number of scenarios. In this work for each metrics 5 runs were conducted each with different randomly deployed topology, so the APDR is calculated as:

$$APDR = \frac{\sum_{i=1}^5 (PDR_i)}{5} \quad (6.2)$$

This metric gives us an idea of how successful the protocols are in delivering packets to the application layer. A high value of PDR indicates that most of the packets are being delivered to the higher layers and is a good indicator of the protocol performance.

### 6.3.2 Average end-to-end Delay (AEED)

End-to-end (EED) delay is the average delay between sending the data packet by the source and its receipt at the corresponding receiver for a single scenario. This includes all the delays caused during route acquisition, buffering and processing at intermediate nodes, and retransmission delays at the MAC layer for a given simulation time. This can be expressed as:

$$EED = \frac{\sum_{i=0}^n Time\_Packet\_Correctly\_Received_i - Time\_Packet\_Sent_i}{Total\_Number\_ofPackets\_Received} \quad (6.3)$$

where n is number of nodes

Here also the AEED was taken from 5 runs with the formula:

$$AEED = \frac{\sum_{i=1}^5 (EEDi)}{5} \quad (6.4)$$

### 6.3.2 Average Normalized Routing Load (ANRL)

*Normalized Routing Load:* is the ratio of routing packets at the network layer to the sum of data packets received. NRL Indicates how many control packets has been applied to successfully deliver a data packet.

$$NRL = \frac{\text{Total\_number\_of\_Control\_Pakcets\_at\_Network\_Layer}}{\text{Total\_Number\_Data\_Packets\_Received}} \quad (6.5)$$

Here also the ANRL was taken from 5 runs with the formula:

$$ANRL = \frac{\sum_{i=1}^5 (NRL_i)}{5} \quad (6.6)$$

## **6.4 Simulation Results**

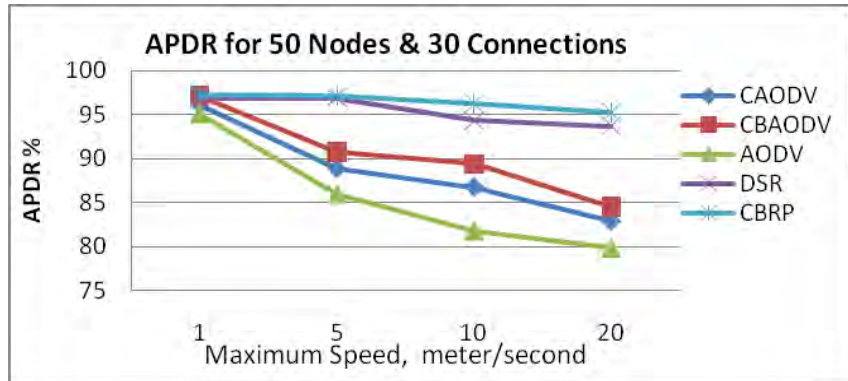
In this section the simulation results are shown with respect to the metrics discussed in the previous section. The section also discusses the results based on the graph.

Here are the graphs for the calculated results obtained from the simulation of five protocols for randomly deployed topology.

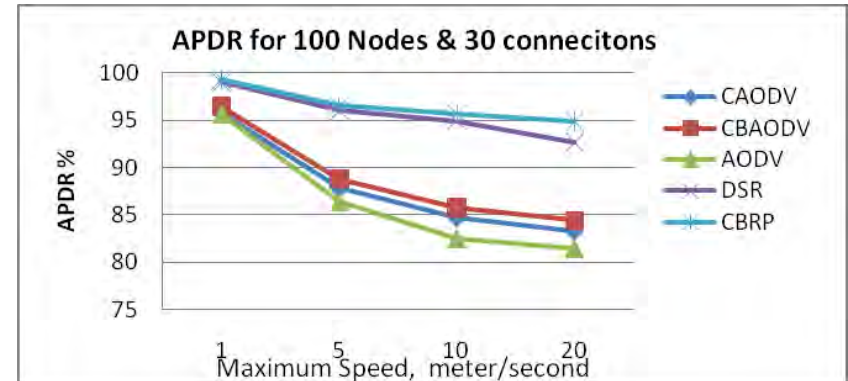
### **6.4.1 Scenario 1: Thirty Connections**

The first scenario is conducted for 50, 100 and 150 number of nodes with a maximum traffic connection of thirty for various mobility scenarios.

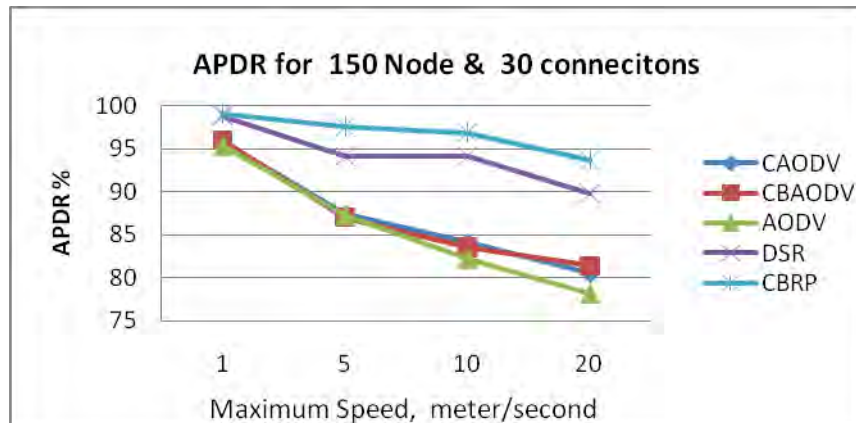
### 6.4.1.1 Average Packet Delivery Ratio (APDR)



(a)



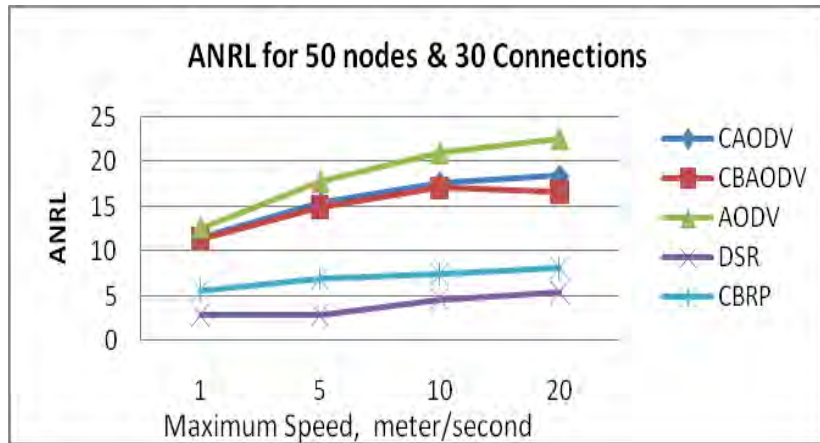
(b)



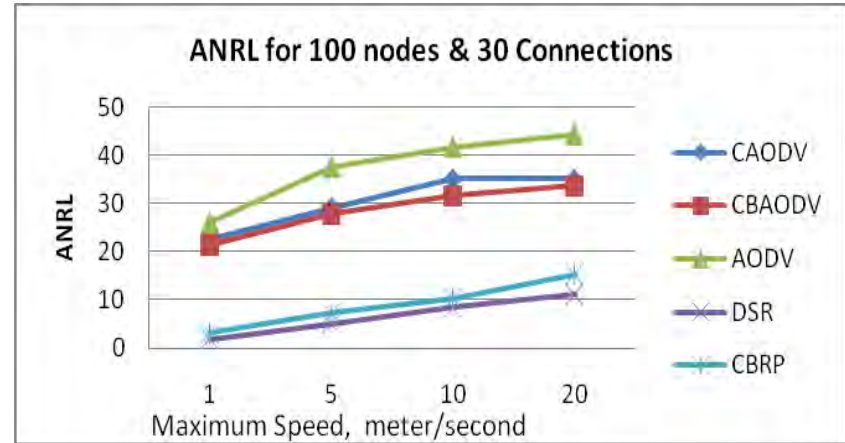
(c)

Fig: 6.1 Packet Delivery Fraction vs different nodes mobility , scenario1.

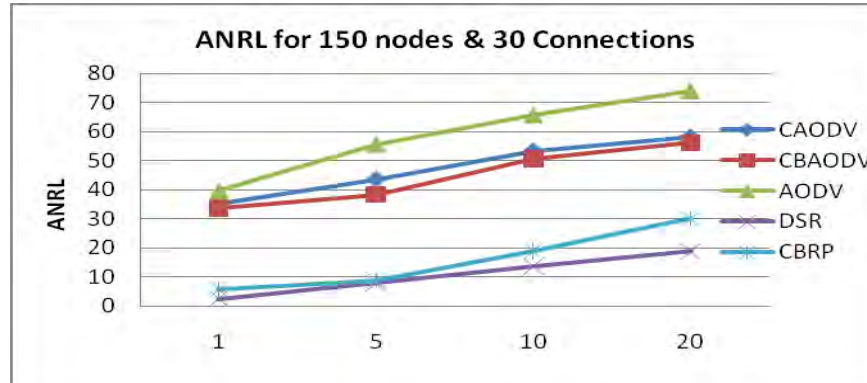
### 6.4.1.2 Average Normalized Routing Load (ANRL)



(a)



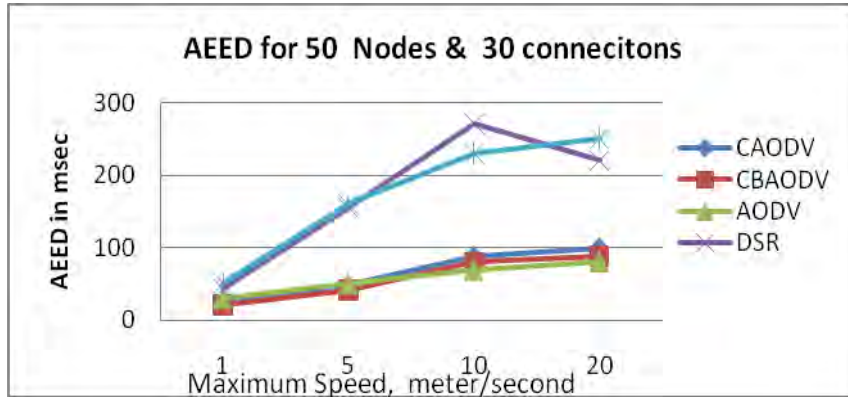
(b)



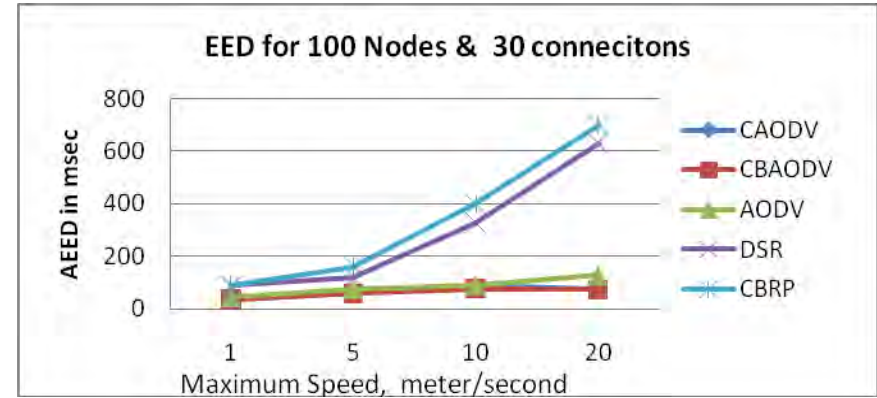
(c)

Fig: 6.2 Average Normalization Routing Load vs different nodes mobility, scenario1.

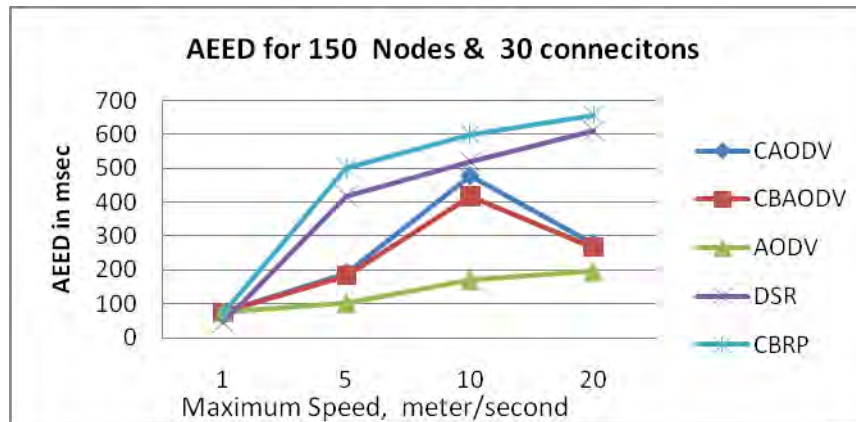
### 6.4.1.3 Average end-to-end Delay (AEED)



(a)



(b)



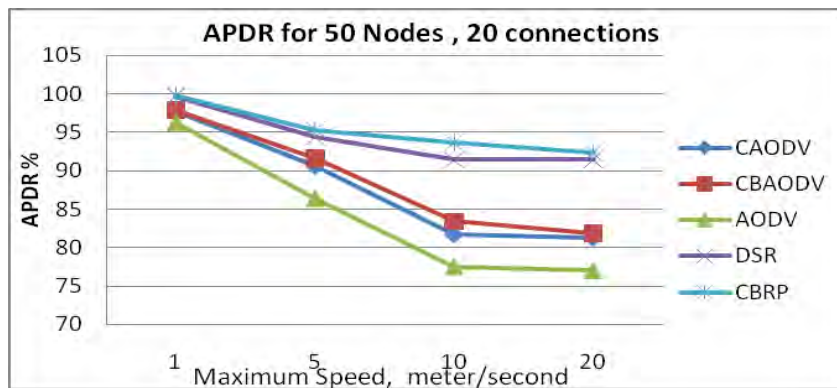
(c)

Fig: 6.3 Average End to End Delay vs different nodes mobility, scenario 1.

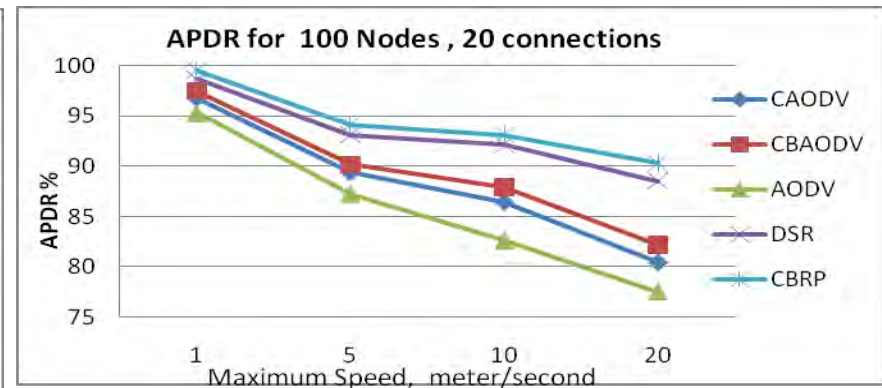
### 6.4.2 Scenario 2: Twenty Connections

The second scenario is conducted for 50, 100 and 150 number of nodes with maximum traffic connections of twenty.

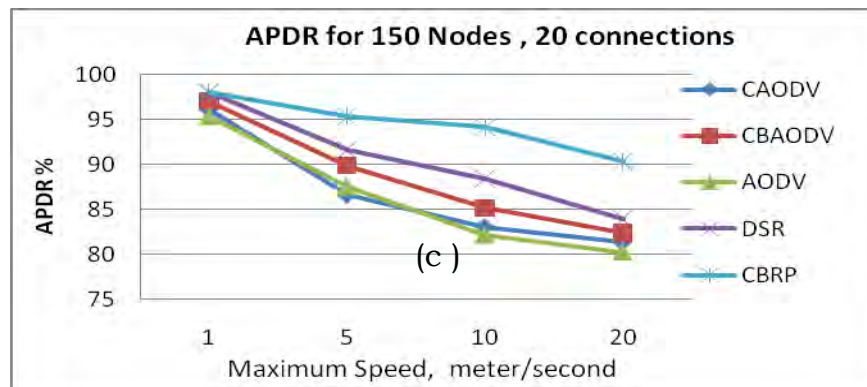
#### 6.4.2.1 Average Packet Delivery Ratio (APDR)



(a)



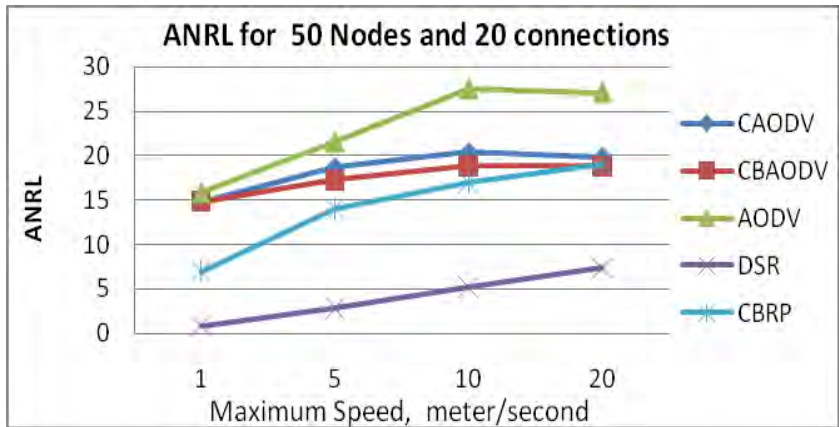
(b)



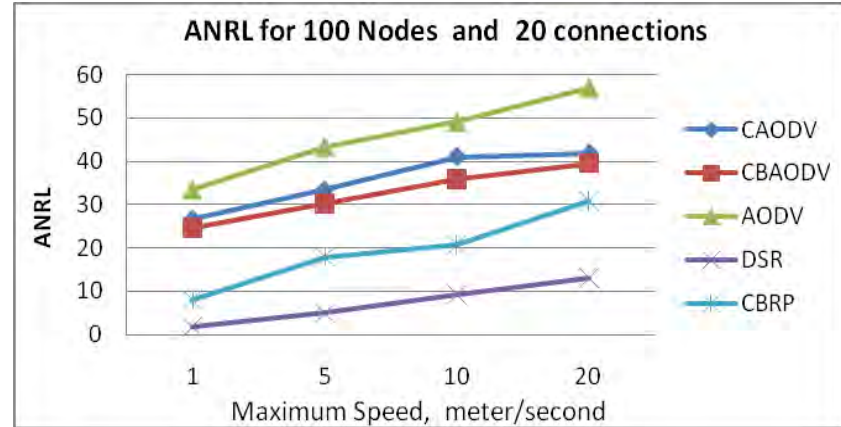
(c)

Fig: 6.4 Packet delivery fraction vs. different Node Mobility, scenario2.

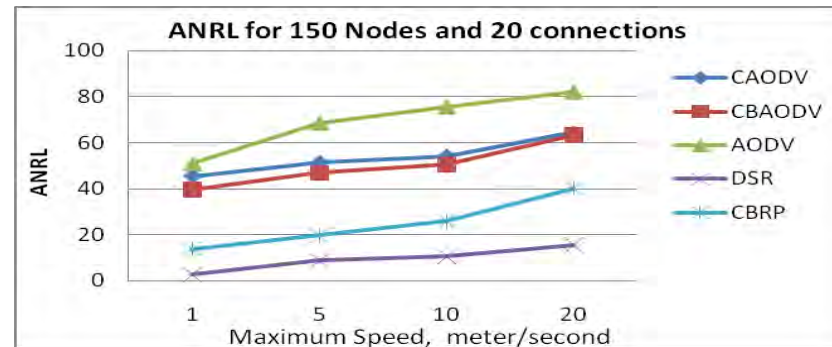
### 6.4.2.2 Average Normalized Routing Load (ANRL)



(a)



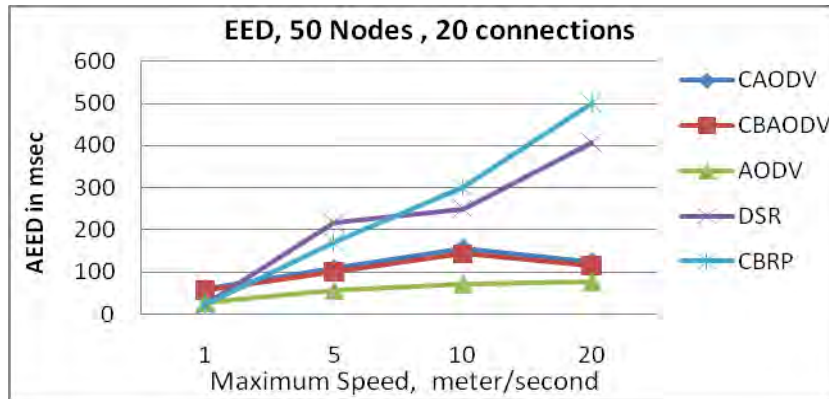
(b)



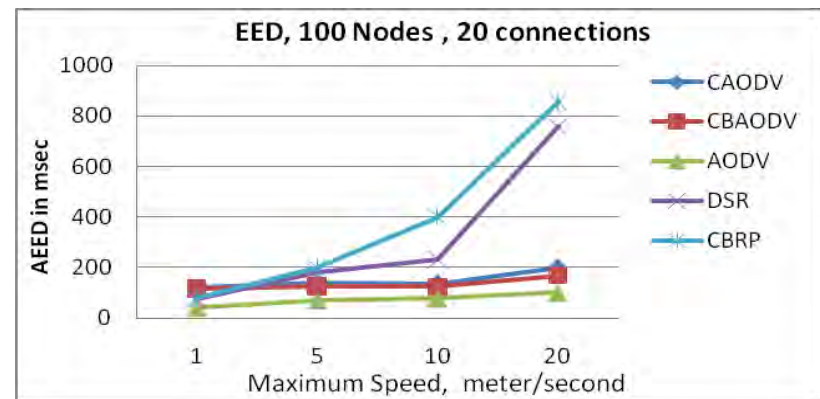
(c)

Fig: 6.5 Average Normalization Routing Load vs different nodes mobility, scenario2.

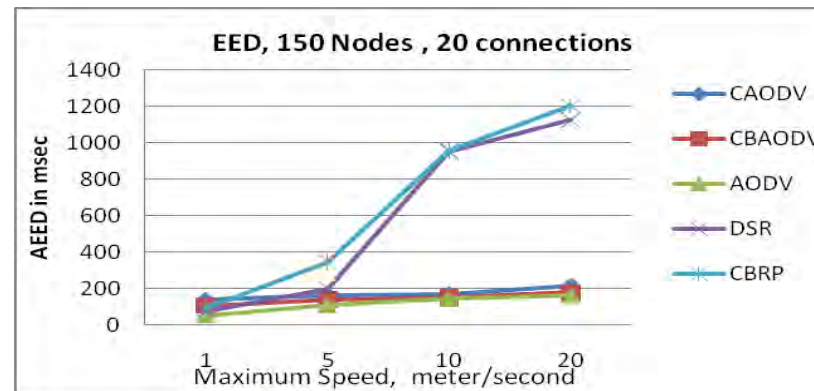
### 6.4.2.3 Average end-to-end Delay (AEED)



(a)



(b)



(c)

Fig: 6.6 Average End to End Delay vs different node mobility, scenario2

As shown in figures above, the performance of the five ad hoc routing protocols AODV, CAOD, CBAODV, CBRP and DSR have been compared with thirty and twenty maximum traffic loads or connections. One can see the similarity of the results from the graphs from scenario one and two, so the results and discussion presents together for both scenarios as follow.

As Fig 6.1 and Fig 6.4 show the CBAODV routing protocol has got up to 7.6% and 2.6% PDR improvement over AODV and CAODV respectively (Fig 6.1(a)). In the mean time the two source routing based protocols, DSR and CBRP, have very high PDR in all scenarios, for example each respectively performs 89.5% and 94.0% for highest mobility, traffic load and network density (Fig 6.1(c) ). When the density of the network increases from 50 to 150 a corresponding performance degradation in PDR happened for the five routing protocols, as shown in Fig 6.1(a, b &c) and Fig 6.4(a, b & c). Results also show that between DSR and CBRP, CBRP has a better PDR for a larger network size. This better scalability comes from its largely reduced flooding for route discovery.

Fig 6.2 and Fig 6.5 shows, without any periodic hello messages, DSR outperforms the other four protocols in terms of routing over load. In most cases, AODV found to have the worst routing over load due to its periodic use of hello messages for link connectivity and route discovery, which is network-wide flooding. CBAODV have got significant improvements over AODV, except in some conditions it has got almost equivalent over load with CAODV.

CBRP has a much smaller flooding range; the number of its route requests and replies is constantly half that of DSR. But its hello messages outweigh this gain.

Among the five protocols, AODV has the shortest end-to-end delay Fig 6.3 and Fig 6.5. The CBAODV and CAODV have got longer delay when compared with AODV, but CBAODV have got some improvements in some scenarios as shown

in Fig 6.3 (c). The source routing protocols, CBRP and DSR have a longer delay because their route discovery takes more time as every intermediate node tries to extract information before forwarding the reply. CBRP, CBAODV and CAODV are time-consuming because of the task of maintaining cluster structure takes time.

In all scenarios when the Energy aware clustering algorithm based clustered AODV (CBAODV) compared with that of the lowest ID clustering algorithm based clustered AODV (CAODV), the CBAODV perform well.

The two cluster based AODV protocols, CBAODV and CAODV, show from the whole scenarios an average of 3.12% and 1.9% PDR improvements over AODV respectively. This is due to the cluster nature of the protocols, only cluster Heads are flooded with Request packets and only they are allowed to Broadcast the packet. The CBAODV have an average of 1.15% PDR over the lowest ID cluster Based AODV. This is due to load balancing on the network, in CBAODV the cluster heads are selected based on available energy in the nodes. Every node initially given 1000 Joules which is enough for 300 second simulation time, but when a node serve the network too much its power decreased and in the next selection of cluster heads it will be free and a new and fresh CH is selected. In the case of CAODV most of the time the lowest ID nodes are selected as CH which creates a significant load on the nodes throughout the network life time. The CBAODV also have got a significant improvement on NRL, an average of 10.7 and 2.2 over AODV and CAODV respectively. From all scenarios results show AODV have an average of 33.7 and 47.47 msec advantages over CBAODV and CAODV respectively.

Meanwhile, the two source routing protocols DSR and its cluster version , CBRP show a very promising PDF and NRL over AODV, CAODV and CBAODV, but they suffer too much in terms of EED when the node mobilities increased this is due to source routing and salvaging of the packets in the intermediate nodes.

## Chapter Seven: Conclusions and future work

This thesis work presents a Cluster Based AODV routing scheme for MANETs to solve the scalability problem of AODV, which consider the residual energy of nodes for cluster formation. The performance of CBAODV is studied by simulations based on NS2. The result shows, the positive application of clustering to achieve better scalability in MANET.

An extensive simulation study conducted to compare the five on-demand ad hoc routing protocols; AODV, CAODV and CBAODV from the distance vector group and DSR and CBRP from source routing group, using a variety of workloads such as mobility, load, and size of the ad hoc networks.

In most scenarios CBAODV show a significant performance improvement over AODV and CAODV. The results also indicate that the two source routing-based protocols, DSR and CBRP, have very high throughputs while the distance-vector-based protocols, exhibits a very short end-to-end delay of data packets.

Furthermore, despite its improvement in reducing route request packets, CBRP has a higher routing overhead than DSR because of its periodic hello messages. DSR has much smaller routing overhead than others, and AODV has the largest overhead among the three protocols.

In this thesis work node's residual energy used for cluster selection criteria, but if a combination of energy, speed, processing capacity and other metrics used for cluster selection, a more scalable routing protocol can be achieved.

The cluster heads can be used for further functionalities in addition to routing management like key and power management

## References

- [1] Carlos de Morais Cordeiro and Dharma Prakash Agrawal “AD HOC AND SENSOR NETWORKS Theory and Applications”, World Scientific Publishing, 2006.
- [2] STEFANO BASAGNI, MARCO CONTI, SILVIA GIORDANO and IVAN STOJMENOVIC “MOBILE AD HOC NETWORKING”, IEEE Press, 2004.
- [3] E. M. Belding-Royer, “Multi-level hierarchies for scalable ad hoc routing,” Wireless Networks, Vol. 9, 2003, pp. 461-478.
- [4] ZHENG Kai and WANG Neng LIU Ai-fang “A new AODV based Clustering routing protocol” in IEEE, 2005
- [5] Sahar Adabi, Sam Jabbehdari, Amirmasoud Rahmani, and Sepideh Adabi “A Novel Distributed Clustering Algorithm for Mobile Ad-hoc Networks” in Journal of Computer Science 4 (2), 2008, pp 161-166
- [6] Wonchang Choi, Miae Woo “A Distributed Weighted Clustering Algorithm for Mobile Ad Hoc Networks” Proceedings of the Advanced International Conference on Telecommunications and International Conference on Internet and Web Applications and Services (AICT/ICIW 2006), IEEE, 2006
- [7] Mingliang Jiang, Jinyang Li and Y.C. Tay “INTERNET-DRAFT CBRP Functional Specification “July 1999
- [8]. Subir Kumar Sarkar, T G Basavaraju, C. Puttamadappa, “Ad hoc Mobile Wireless Networks,” Taylor & Francis Group, LLC.2008.
- [9] Tim Daniel Hollerung “The Cluster-Based Routing Protocol” Winter 2003/2004
- [10] Denko, M.K. and Lu, H. “An AODV-based clustering for mobile ad hoc networks” , IFIP International Conference on Ad hoc Networking (ADHOC-NET06), held with the 19th IFIP World Computer Congress, Santiago, Chile, published by Springer, 20-25 August 2006.
- [11] Ya-qin Fan, Lin-zhu Wang and Li Zhou “The Comparison and Simulation of Ad Hoc Network Routing Protocol Based on NS2” JOURNAL OF NETWORKS, VOL. 5, NO. 4, APRIL 2010.

- [12] Rashim “MANET (Mobile Adhoc Network)”, <http://www.saching.com> , 2008.
- [13] C. Perkins, E. Royer, and S. Das. “Ad hoc on demand distance vector (AODV)” Internet Draft, Internet Engineering Task Force, February 2003.
- [14] Ian D. Chakeres and Elizabeth M. Belding-Royer “AODV Routing Protocol Implementation Design” Proceedings of the International Workshop on Wireless Ad hoc Networking (WWAN), Tokyo, Japan, March 2004, pp. 698-703.
- [15] Ratish Agarwal and DR. Mahesh Motwani “Survey of clustering algorithms for MANET”, International Journal on Computer Science and Engineering VII.1 (2), 2009, 98-104
- [16] JANE Y. YU AND PETER H. J. CHONG “A SURVEY OF CLUSTERING SCHEMES FOR MOBILE AD HOC NETWORKS” IEEE Communications Surveys & Tutorials, First Quarter 2005.
- [17] Sahar Adabi, Sam Jabbehdari, Amirmasoud Rahmani and Sepideh Adabi “A Novel Distributed Clustering Algorithm for Mobile Ad-hoc Networks” Journal of Computer Science 4 (2): 161-166, 2008
- [18] M. BHEEMALINGAIAH, M. M. NAIDU, D. Sreenivasa RAO “Energy Aware Clustered Based Multipath Routing in Mobile Ad Hoc Networks”, I. J. Communications, Network and System Sciences, 2009, 2, 91-168
- [19] Sanlin Xu “Mobility Metrics for Routing in MANETs” A thesis submitted for the degree of Doctor of Philosophy of The Australian National University, June 2007
- [20] M.Chatterjee, S.K. Das, and D.Turgut, “An On-Demand Weighted Clustering Algorithm (WCA) for Ad hoc Networks,” in proceedings of IEEE Globecom’00, pp. 1697-701, 2000.
- [21] I. Er and W. Seah. “Mobility-based d-hop clustering algorithm for mobile ad hoc networks”, IEEE Wireless Communications and Networking Conference Vol. 4. Pp. 2359-2364, 2004
- [22] A.D. Amis and R. Parkash, “Load-Balancing Clusters in Wireless Ad Hoc Networks,” in proceedings of 3<sup>rd</sup> IEEE ASSET’00, pp. 25-32 Mar.2000
- [23] M.Geral and J.T.Tsai, “Multiuser, Mobile, Multimedia Radio Network,” Wireless Networks, vol.1.pp.255-65, Oct. 1995.

- [24] NS-2 Network simulator. <http://www.isi.edu/nsnam/ns/>.
- [25] Kevin Fall and Kevin Fall "The ns Manual" the VINT Project, January, 2010.
- [26] Mingliang Jiang, Jinyang Li and Y.C. Tay "INTERNET-DRAFT CBRP Functional Specification " July 1999
- [27] Tim Daniel Hollerung "The Cluster-Based Routing Protocol", winter semester 2003/2004
- [28] Chiang, C.C., Wu, H.K., Liu, W., Gerla, M., "Routing in Clustered Multihop, Mobile Wireless Networks with Fading Channel", The Next Millennium, the IEEE SICON, 1997.
- [29] T. Camp, J. Boleng, and V. Davies, "A Survey of Mobility Models for Ad Hoc Network Research," Wireless Communication & Mobile Computing (WCMC): Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications, vol. 2, no. 5, pp 483-502, 2002. Copyright (C) 2004 Toilers Research Group -- Colorado School of Mines.
- [29] Azzedine Boukerche "A Performance Comparison of Routing Protocols for Ad Hoc Networks" Parallel Simulations and Distributed Systems (PARADISE) Research Laboratory IEEE, 2001
- [30] Cao Jianglian, Zhang Zhaoxiao, Zeng Wenli, Pei Tingrui "Energy-aware AODV Routing for Ad Hoc Networks" International Conference on Networks Security, Wireless Communications and Trusted Computing, 2009
- [31] Xinsheng Wang, Qing Liu and Nan Xu "The Energy-Saving Routing Protocol Based on AODV" Fourth International Conference on Natural Computation , IEEE 2008.
- [32] Yogesh Chaba, Yudhvir Singh and Manish Joon "Performance Evaluation and Analysis of Cluster Based Routing Protocols in MANETs" International Conference on Advances in Computing, Control, and Telecommunication Technologies, IEEE 2009.
- [33] Sze-Yao Ni, Yu-Chee Tseng, Yuh-Shyan Chen, Jang-Ping Sheu, "The broadcast storm problem in a mobile ad hoc network", MobiCom, 1999.
- [34] Jinyang Li, Charles Blake, Douglas S.J., De Couto, Hu Imm Lee, Robert Morris, "Capacity of Ad Hoc Wireless Networks", Mobicom, 2001.
- [35] Ian D. Chakeres and Elizabeth M. Belding-Royer "The Utility of Hello Messages for Determining Link Connectivity." Proceedings of the 5th

International Symposium on Wireless Personal Multimedia Communications (WPMC) 2002.

[36] C. E. Perkins and P. Bhagwat , “Highly Dynamic Destination-Sequenced Distance Vector (DSDV) for Mobile Computers,” Proc. of the SIGCOMM 1994 Conference on Communications Architectures, Protocols and Applications, Aug 1994, pp 234-244.

[37] 21. J. Strater and B. Wollman, “OSPF Modeling and Test Results and Recommendations,” Mitre technical report 96W0000017, Xerox Office Products Division, March 1996.

[38] T. Clausen, P. Jacquet, A. Laouiti, P. Minet, P. Mulethaler, A. Qayyum, and L. Viennot, “Optimized Link State Routing Protocol,” IETF DRAFT, draft-ietf-manet-olsr-02.txt, <http://hipercom.inria.fr/olsr/>, July 2002.

[39] Zygmunt J. Haas , Marc R. Pearlman and Prince Samar, “The Zone Routing Protocol (ZRP) for Ad Hoc Networks,” Internet Draft, <http://www.ietf.org/proceedings/02nov/I-D/draft-ietf-manet-zone-zrp-4.txt>, work in progress, July 2002.

[40] Y.-B. Ko and V. N. H., ”Location-Aided Routing in mobile Ad hoc networks,” ACM/IEEE Mobicom, pages 66-75, October 1998.

[41] David Johnson, David Maltz and Yih-Chun Hu, “The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks,” Internet Draft, draft-ietf-manet-dsr-10.txt, work in progress, July 2004

[42] C. E. Perkins, E. M. Belding-Royer, and S. Das, “Ad hoc on-demand distance vector (AODV) Routing,” RFC 3561, July 2003,

## Appendix A: AWK SCRIPTS FOR METRICS CALCULATIONS

```
BEGIN {
    sends=0; flag=0; recvs=0; routing_packets=0.0;
droppedBytes=0; droppedBytes7=0; check=0; droppedPackets=0;
highest_packet_id =0; sum=0; summean=0.0; count=0; avghop=0;
recvnum=0; noerrors=0; nodied=0; firsttime=0.0; EnergyConsume=0.0
variance=0.0;
    }

    {
    time = $2;
    packet_id = $6;
    NodeId=$3;

    # CALCULATE PACKET DELIVERY FRACTION

    if (( $1 == "s")&& ( $4=="AGT" ) && ( $7 == "cbr" ) ) {
        sends++; }
    if (( $1 == "s")&& ( $4=="MAC" ) && ( $7 == "cbr" ) ) {
        avghop++; }
    if (( $1 == "r")&&( $4=="AGT" ) && ( $7 == "cbr" ) ) {
        recvs++; }

    # CALCULATE DELAY

    if ( start_time[packet_id] == 0 ) start_time[packet_id] = time;

    if (( $1 == "r")&&( $4=="AGT" ) && ( $7 == "cbr" ) ) {
end_time[packet_id] = time; }

        else { end_time[packet_id] = -1; }

    # CALCULATE TOTAL AODV

    if (($1 == "s") &&( $4== "RTR") && ($7 == "AODV") )
        routing_packets++;

    # DROPPED PACKETS

    if (($1 == "D" ) && ( $7 == "cbr" ) && ($2 > 0) )

        {
            droppedBytes=droppedBytes+$8;

            droppedPackets=droppedPackets+1;

        }

    #find the number of packets in the simulation
```



## Appendix B: TCL SCRIPTS

### *AODV , CBAODV AND CAODV SIMULATION TCL SCRIPTS*

```
=====
# Define options
# =====

set val(chan)          Channel/WirelessChannel
set val(prop)          Propagation/TwoRayGround
set val(netif)         Phy/WirelessPhy
set val(mac)           Mac/802_11
set val(ifq)           Queue/DropTail/PriQueue
#set val(ifq)          CMUPriQueue      ;# interface queue type
set val(ll)            LL
set val(ant)           Antenna/OmniAntenna
set val(TxPwr)         0.281838
set val(x)             1000      ;# X dimension of the topography
set val(y)             1000      ;# Y dimension of the topography
set val(ifqlen)        50          ;# max packet in ifq
set val(seed)          0.0
set val(adhocRouting)  AODV
set val(nn)            150          ;# how many nodes are simulated
set val(cp)            "conal50"
set val(sc)            "tthmobb150"
set val(stop)          300.0       ;# simulation time
set val(energyMdl)     EnergyModel ;
set val(initenergy)    1000        ;# Initial energy in Joules
# =====
# Main Program
# =====
# Initialize Global Variables
# create simulator instance
set ns_ [new Simulator]
# setup topography object
set topo [new Topography]
# create trace object for ns and nam
set tracefd [open tr150ene.tr w]
set namtrace [open nam150ene.nam w]
$ns_ trace-all $tracefd
$ns_ namtrace-all-wireless $namtrace $val(x) $val(y)
#$ns_ use-newtrace
# define topology
$topo load_flatgrid $val(x) $val(y)
# Create God
set god_ [create-god $val(nn)]
# define how node should be created
#global node setting

$ns_ node-config -adhocRouting $val(adhocRouting) \
                -llType $val(ll) \
```

```

        -macType $val(mac) \
        -ifqType $val(ifq) \
        -ifqLen $val(ifqlen) \
        -antType $val(ant) \
        -propType $val(prop) \
        -phyType $val(netif) \
        -channelType $val(chan) \
        -topoInstance $topo \
        -agentTrace ON \
        -routerTrace ON \
        -energyModel $val(energyMdl) \
        -initialEnergy $val(initenergy) \
        -macTrace ON
# Create the specified number of nodes [$val(nn)] and "attach" them
# to the channel.

for {set i 0} {$i < $val(nn)} {incr i} {
    set node_($i) [$ns_ node
        $node_($i) random-motion 0           ;# disable random motion
    }
#
# Define node movement model
#
puts "Loading connection pattern..."
source $val(cp)
#
# Define traffic model
#
puts "Loading scenario file..."
source $val(sc)

# Define node initial position in nam

for {set i 0} {$i < $val(nn)} {incr i} {

    # 20 defines the node size in nam, must adjust it according to your
    scenario
    # The function must be called after mobility model is defined

    $ns_ initial_node_pos $node_($i) 20
}

#
# Tell nodes when the simulation ends
#
for {set i 0} {$i < $val(nn)} {incr i} {
    $ns_ at $val(stop).0 "$node_($i) reset";
}

$ns_ at $val(stop).0002 "puts \"NS EXITING...\" ; $ns_ halt"

puts $tracefd "M 0.0 nn $val(nn) x $val(x) y $val(y) rp $val(adhocRouting)"
puts $tracefd "M 0.0 sc $val(sc) cp $val(cp) seed $val(seed)"
puts $tracefd "M 0.0 prop $val(prop) ant $val(ant)"

```

```
puts "Starting Simulation..."  
$ns_ run
```