1. INTRODUCTION

Ad-hoc Network in a wireless can be supposed as a group of mobile nodes on wireless medium sharing data and communication with each other which don't have any infrastructure or an administrator in the network. Ad-hoc network in build for the required purpose of the nodes after the purpose is obtained the network no longer exists. As there is a shift in technology from wired devices to wireless devices so a need for the protocol which can be useful in handling communication between the mobile nodes in Ad-hoc network will be raised.

Many protocol as defined previously to deal with this need of change in trend in technology but most of them don't consider an important factor which is the backbone of the wireless devices i.e. the energy in the nodes. The wireless devices run on the batteries they don't have the constant power supply. So the routing protocol should consider the energy as the metric in selecting the routes in routing so as to increase the network lifetime by preserving the energy of the nodes.

Several protocols are already developed for the communication between the mobile devices of Ad-hoc network devices e.g. DSDV [14], DSR [13], AODV [12], [16], [15], GSR [18], OLSR [19], and ZPR [17]. In this dissertation an improvement over AODV [12]. [15], [16] and DSDV [14] are performed so the routing protocol takes into account energy as a metric for the route selection.

2. SOURCE OF INSPIRATION

In the last few years the wired device are shifting to the wireless medium and we have noticed a huge increase in demand of the wireless devices. So to manage communication between the wireless devices some protocol needed to be developed. Some protocol exists to handle the communication between the wireless devices which are mobile in nature e.g. DSDV [14], DSR [13], AODV [12], [16], [15], GSR [18], OLSR [19] and ZPR [17]. But the problem in these protocol exits is to how to manage the energy so that the life of the network increases.

To manage the energy of the node in the way that leads to the maximize the network lifetime some alternative path should be chosen which utilizes the nodes which have more remaining energy than other nodes. But the selection of other routes should not lead to decrease in communication speed which is totally acceptable since the links transfer speed are quite high and that result in difference of few milliseconds if we prefer the alternate routes. This fact inspires us in the direction of modifying the existing protocols so that the end result should be increased network lifetime.

3. PROPOSED MODEL

The proposed model to preserve energy is to calculate the *Energy Preserving Value* (EPV) which is based upon the remaining energy of the node and the number of neighbours connected to the node. The energy model is the standard energy model available in the *Network Simulator* (Ns-2) [4], [5].

When the EPV of the node is calculated its forward this value to its neighbour the receiving the EPV adds up its own EPV which is termed as *Energy Preserving Value Total* (EPVT). Now the reverse route stored at each intermediate node is based upon the EPVT not on the hop count. The route request reaches the destination node it replies backs to the nodes from which it received the request. The previously saved reverse route is used to generate the path between the source and the destination node.

The update of the route is done only when the new sequence numbered route is received or the same sequence number but a higher value of the EPVT at the source node. This change in route selection policy leads to choose the path in which nodes with higher energies as compared to the nodes in the other available routes.

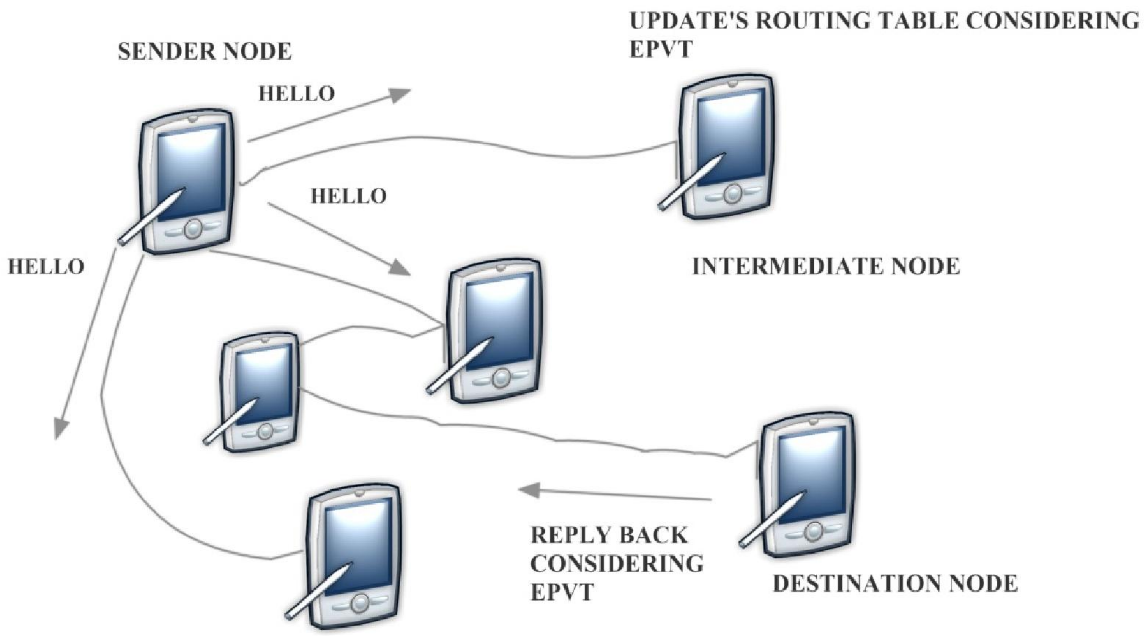

Fig 3.1 Proposed Model

The above figure depicts the general functioning of the proposed model. In general the source node broadcast the Hello packets to acquire the knowledge about its neighbours. The nodes connecting to the nodes updates the routing table based upon the EPVT calculated which don't follow the least hop count or minimum distance concept for finding the route. When the route request reaches the destination node its reply's to the node with reply packet. The reverse route is created using the value stored in routing table which is updated using EPVT so the generated route is more energy efficient than the previous route which was selected by the minimum hop count concept.

4. NETWORK SIMULATOR

4.1 NS-2

The simulation of the above proposed model is done under the NS-2 with using the standard energy model available in the NS-2.NS (version 2) is a discrete event driven simulator for network it's based on the concepts of object orientation and written in C++ [6] and OTcl developed at UC Berkeley. NS is generally used to simulate the local and wide area networks. It facilitates its uses with the protocols like TCP, UDP, router queue management, Telnet, FTP and routing algorithms like Dijkstra [20]. The NS is currently under VINT[21] project for simulation.

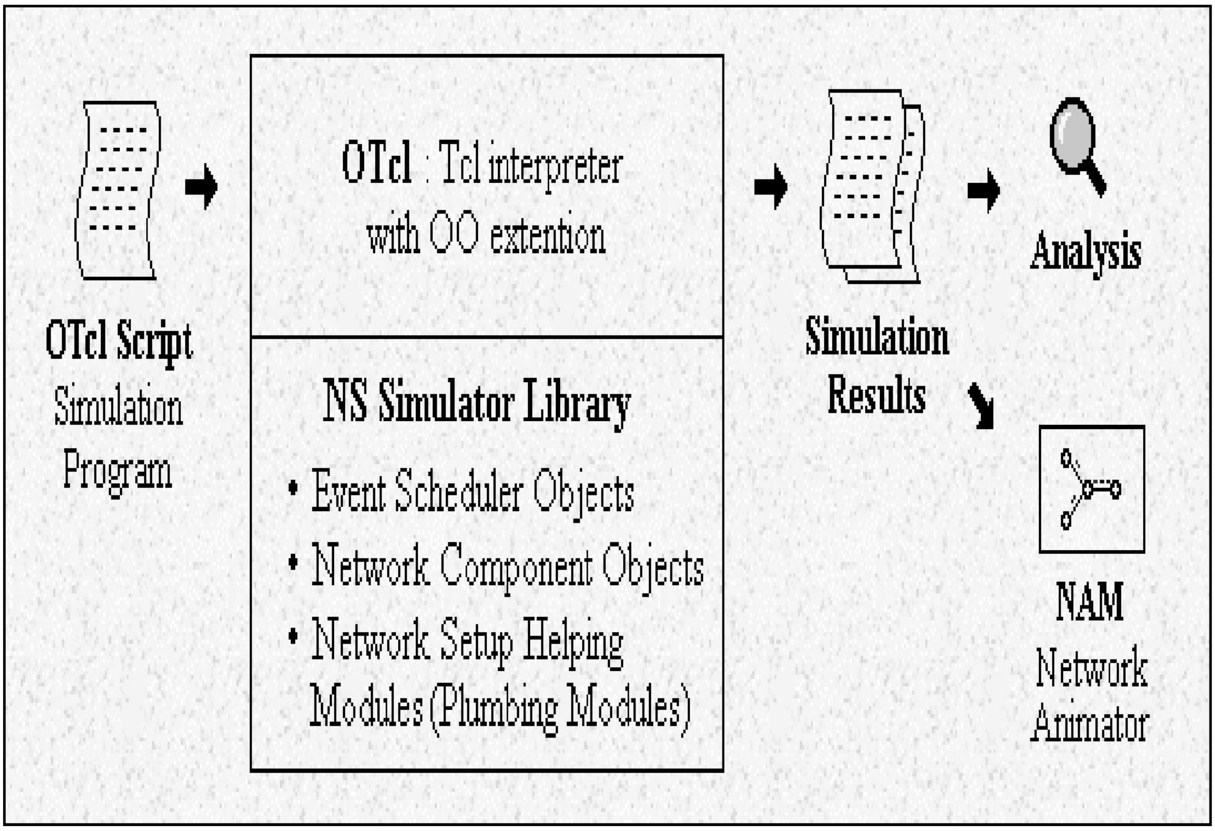

Fig. 4.1 User view of NS-2

4.2 NAM

Network Animator (NAM) is a Tcl based tool available with NS for viewing the simulation of network. It supports animation at packet level, topological view and various data observing tools. The NAM development is currently under the VINT [21] project.

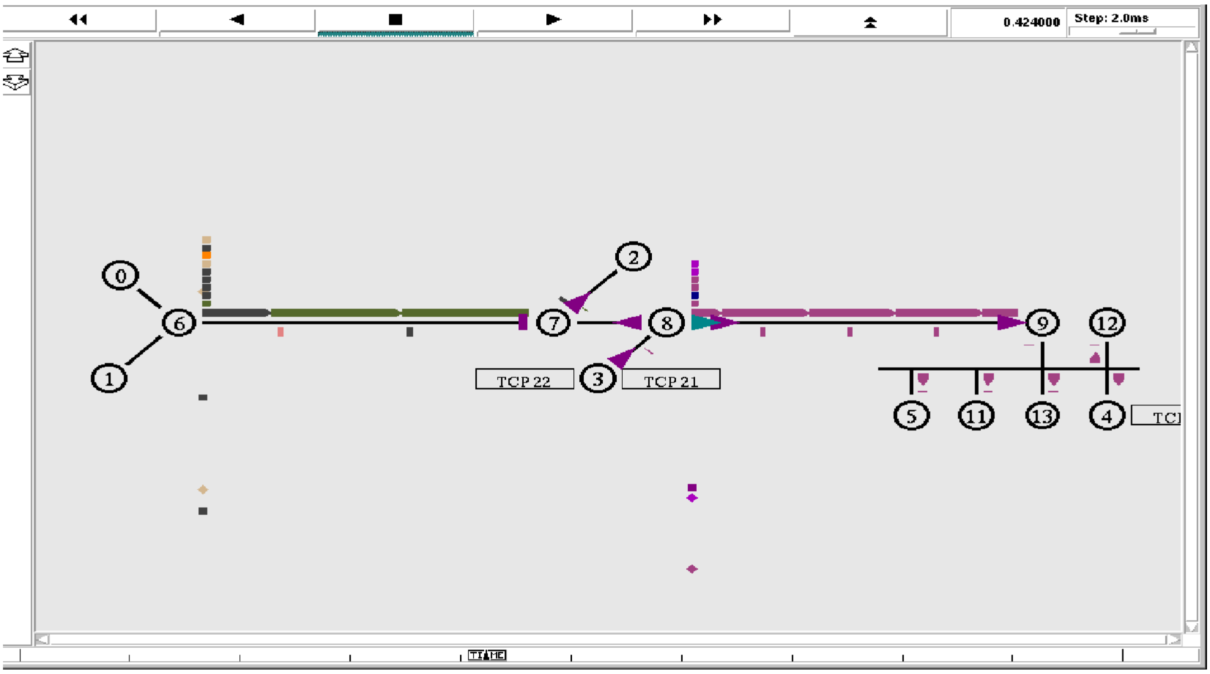

Fig. 4.2 A view of NAM

4.3 Node in NS-2

A node is a blend of node entry object and classifiers. A unicast node has an address classifier that performs the unicast routing and port classifier. While a multicast node, has a classifier that classifies between the multicast and unicast packets and in addition a multicast classifier for multicast routing. The default node in NS-2 is unicast node.

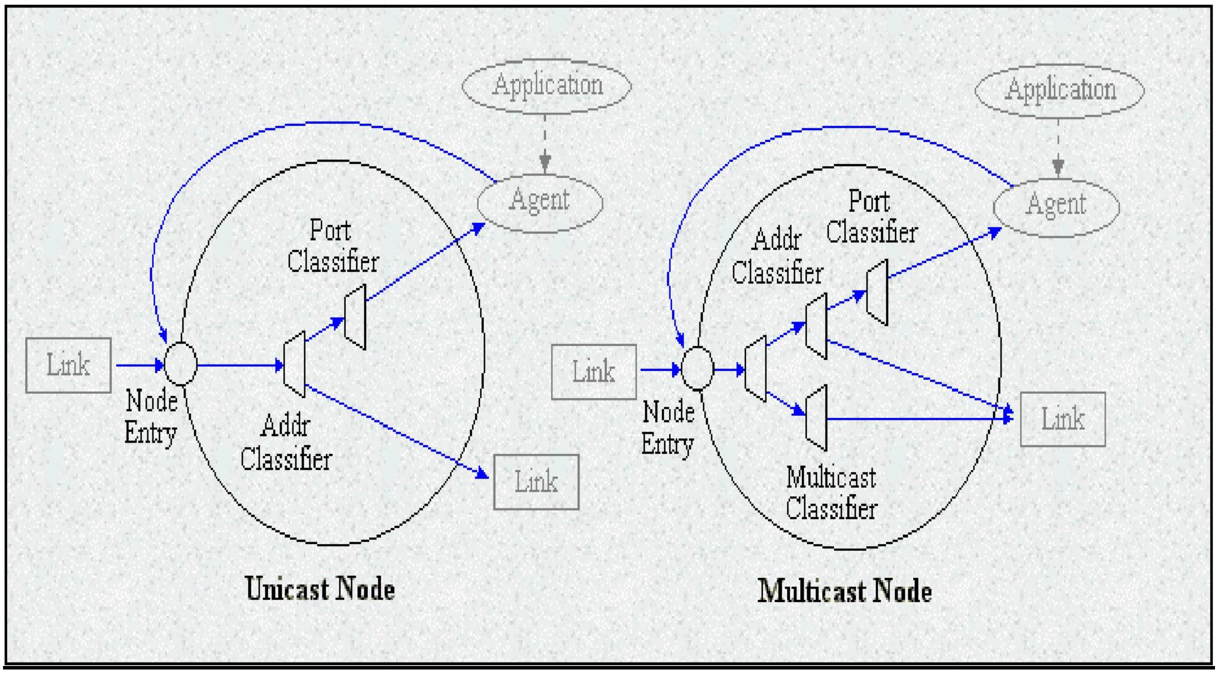

Fig. 4.3 Types of node in NS-2

4.4 Post Simulation Tracing

Trace files is the features provided by the NS-2 to trace down the events while the simulation is running. It traces down the time of every event, source node, destination node, type of packet etc. The traces stored in the trace file are required to conclude about the simulation performed. Below is the sample trace file of NS-2.

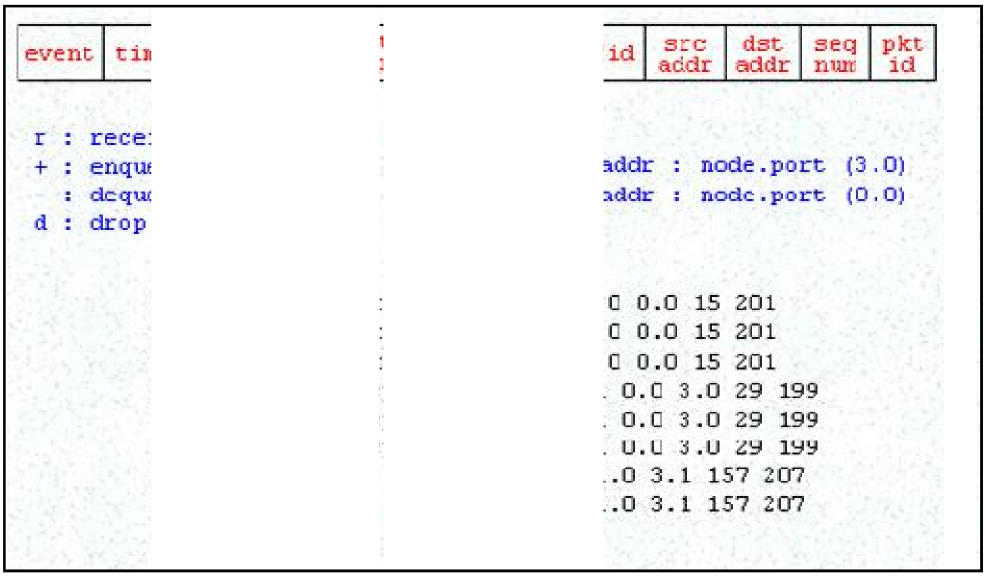

Fig. 4.4 A Sample trace file of NS-2

5. TYPES OF ROUTING

5.1 Link State Routing

In link state routing each node maintains its view of the network with taking into account the cost of each link. Each node broadcasts its view of the network periodically in the network which maintains the consistency of the network using flooding. As any node receives the view of the network from the other nodes it updates its view and applies the shortest path algorithm to update its next hop for each destination.

Some of the cost of the links could be incorrect due to larger propagation delay in the network which could lead to the looping in the network. But these loops are eliminated easily as the information is updated.

5.2 Distance Vector Routing

In distance vector algorithm every node i maintain for each destination x a set of distances $\{d^x ij\}$ where j ranges over the neighbors of i. Node i treats neighbor k as next-hop for a packet destined for x if $d^x ij$ is $\min(d^x ij)$. The above distance vector algorithm is termed as classical Distributed Bellman Ford algorithm.

Distance vector routing doesn't consider the cost of link it just works upon the hop count required to reach the destination. Hence it is computationally advantageous to use distance vector in comparison to the above link state routing. But using Distance vector routing could lead up to the looping which could be even worse when it turns into the count to infinity problem. This problem is eliminated using the sequence number to verify the freshness of the packet received.

6 TYPES OF PROTOCOLS

6.1 Proactive Protocols

Each node in the network has routing table for the broadcast of the data packets and want to establish connection to other nodes in the network. These nodes record for all the presented destinations, number of hops required to arrive at each destination in the routing table. The routing entry is tagged with a sequence number which is created by the destination node. To retain the stability, each station broadcasts and modifies its routing table from time to time. How many hops are required to arrive that particular node and which stations are accessible is result of broadcasting of packets between nodes. Each node that broadcasts data will contain its new sequence number and for each new route, node contains the following information:

- How many hops are required to arrive that particular destination node
- Generation of new sequence number marked by the destination
- The destination address

The proactive protocols are appropriate for less number of nodes in networks, as they need to update node entries for each and every node in the routing table of every node. It results more Routing overhead problem. There is consumption of more bandwidth in routing table. Example of Proactive Routing Protocol is *Destination Sequenced Distance Vector* (DSDV).

6.2 Reactive Routing

Reactive Protocol has lower overhead since routes are determined on demand. It employs flooding (global search) concept. Constantly updating of route tables with the latest route topology is not required in on demand concept. Reactive protocol searches for the route in an on-demand manner and set the link in order to send out and accept the packet from a source node to destination node. Route discovery process is used in on demand routing by flooding the route request (RREQ) packets throughout the network. Examples of reactive routing protocols are the dynamic source Routing (DSR), ad hoc on-demand distance vector routing (AODV).

7 AODV PROTOCOL

7.1 AODV

AODV [15], [12], [16] (Adhoc on Demand Distance Vector) is a routing protocol which was developed for use by mobile nodes in adhoc network which is usually termed as the MANET [8].The used in a dynamically changing network environment i.e. the topology of the network is changing frequently, the protocol use less overhead in route discovery and maintenance. It uses destination sequence numbers to ensure the loop freedom routing in the routing which problem "like count to infinity" [11] which usually suffered by classical distance vector protocols.

AODV allows mobile node to discover routes for the destination quickly, without maintaining the routes for the nodes which are currently inactive in the network. The feature of AODV which separates it from the other protocols that is uses destination sequence number created by the destination which is transferred along with route information to the requesting nodes. Using destination sequence number ensures freedom from the loops. As if there are two routes for the desired destination the route with the largest destination sequence number is accepted the other route is discarded.

Route Requests (RREQ), Route Replies (RREP), and Route Errors (RERR) are the messages types defined by the AODV [12], [16], [15]. AODV requires certain messages to be distributed widely in the entire adhoc network. As the two ends of communication i.e. source node and destination node have the valid route between them AODV don't play any role between them.

Initially all nodes in the network creates the Hello packet which consist of information about the node and exchange the Hello packets. After Hello packet reception each node adds their neighbors into its routing table creating separate entry for each neighbor. When source wants to communicate with the destination which is not its neighbor. Source broadcast the route request (RREQ) message in the network each intermediate node receiving the RREQ message creates a reverse route to the source which is used when the destination reply to the source request. When the RREQ message reaches the destination it creates the route reply (RREP) for the source. Following the reverse route created during forwarding of the packet

RREP reaches the source. There can be multiple packets reaching to the source which are discarded based upon the sequence number of the packet. Shorter route is updated whenever an packet with the unexpired sequence number and have less hop count than the current route is reached to the source.

Many relevant information other than source and destination sequence number is also stored in the routing table of the node is commonly known as soft state related to the node. In each routing table entry, the addresses of active neighbors from which packets for the given destination are received are also stored. A route entry is said to be active if it is in used by active nodes. The path from a source to the destination which is followed by packets along the active entries is called an active path.

When the link between nodes breaks this event is announced with the route error (RERR) message. When a node found that the link between its previously connected node is temporarily/permanently down the node generates the RERR message for the connected node so notify it that the link id no more available. A node maintains the precursor list of node connected node precursor list contains the list node from which the current node is connected. After node confirms the link broken it removes it from the precursor list. RERR message can be broadcasted, unicast or multiple unicasted. When the message is unicasted it is termed as the control message.

AODV [12], [15], [16] stores the following information in the route table entry:

— Destination IP Address.
— Network Interface.
— Hop Count (number of hops needed to reach destination).
— Next Hop.
— Destination Sequence Number.
— List of Precursors (Nodes which likely to use the current node as route node to communicate with the destination node).
— Lifetime (expiration or deletion time of the route).
— Valid Destination Sequence Number flag. Other state and routing flags (e.g., valid, invalid, repairable, being repaired).

7.2 MESSAGE FORMATS

7.2.1 Route Request (RREQ) Message Format

<table>
<tr><td colspan="10">0</td><td colspan="10">1</td><td colspan="10">2</td><td colspan="2">3</td></tr>
<tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td></tr>
<tr><td colspan="8">Type</td><td>J</td><td>R</td><td>G</td><td>D</td><td>U</td><td colspan="11">Reserved</td><td colspan="8">Hop Count</td></tr>
<tr><td colspan="32">RREQ ID</td></tr>
<tr><td colspan="32">Destination IP Address</td></tr>
<tr><td colspan="32">Destination Sequence Number</td></tr>
<tr><td colspan="32">Originator IP Address</td></tr>
<tr><td colspan="32">Originator Sequence Number</td></tr>
</table>

Type: 1.

J: Join flag (reserved for multicasting).

R: Repair flag (reserved for multicasting).

G: Gratuitous RREP flag.

D: Destination only flag.

U: Unknown Sequence Number.

Reserved: Sent as 0.

Hop Count: The required number of hops from originating node to destination.

RREQ ID: Used for identifying the RREQ uniquely.

Destination IP Address: The IP Address of destination.

Destination Sequence Number: The latest sequence received by the originator of destination.

Originator IP Address: IP Address of the node which originated RREQ.

Originator Sequence Number: The sequence number which leads to the originator.

<u>7.2.2 Route Reply (RREP) Message Format</u>

<table>
<tr><td colspan="10">0</td><td colspan="10">1</td><td colspan="10">2</td><td colspan="2">3</td></tr>
<tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td></tr>
<tr><td colspan="8">Type</td><td>R</td><td>A</td><td colspan="9">Reserved</td><td colspan="5">Prefix Size</td><td colspan="8">Hop Count</td></tr>
<tr><td colspan="32">Destination IP Address</td></tr>
<tr><td colspan="32">Destination Sequence Number</td></tr>
<tr><td colspan="32">Originator IP Address</td></tr>
<tr><td colspan="32">Lifetime</td></tr>
</table>

Type:	2.
R:	Repair flag (used for multicast).
A:	Acknowledgment required.
Reserved:	Sent as 0.
Prefix Size:	If nonzero, it specifies that the next hop may be used for any node with the same prefix size.
Hop Count:	The number of hops from destination to originator.
Destination IP Address:	IP Address of destination.
Destination Sequence Number:	The destination sequence number when RREP generated.
Originator IP Address:	The IP address of the node from where RREQ is broadcasted.
Lifetime:	The time (ms) for which destination considers the route valid.

7.2.3 Route Error (RERR) Message Format

<table>
<tr><td colspan="10">0</td><td colspan="10">1</td><td colspan="10">2</td><td colspan="2">3</td></tr>
<tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td></tr>
<tr><td colspan="8">Type</td><td>N</td><td colspan="16">Reserved</td><td colspan="7">DestCount</td></tr>
<tr><td colspan="32">Unreachable Destination IP Address</td></tr>
<tr><td colspan="32">Unreachable Destination Sequence Number</td></tr>
<tr><td colspan="32">Additional Unreachable Destination IP Addresses</td></tr>
<tr><td colspan="32">Additional Unreachable Destination Sequence Numbers</td></tr>
</table>

Type: 3.

N: No delete flag, when a node performed local repair of links, the upstream node should not delete flags.

DestCount: The number of unreachable destinations.

Unreachable Destination IP Address: No route for the specified destination.

Unreachable Destination Sequence Number: The sequence number of the unreachable destination IP address.

7.3 WORKING OF AODV

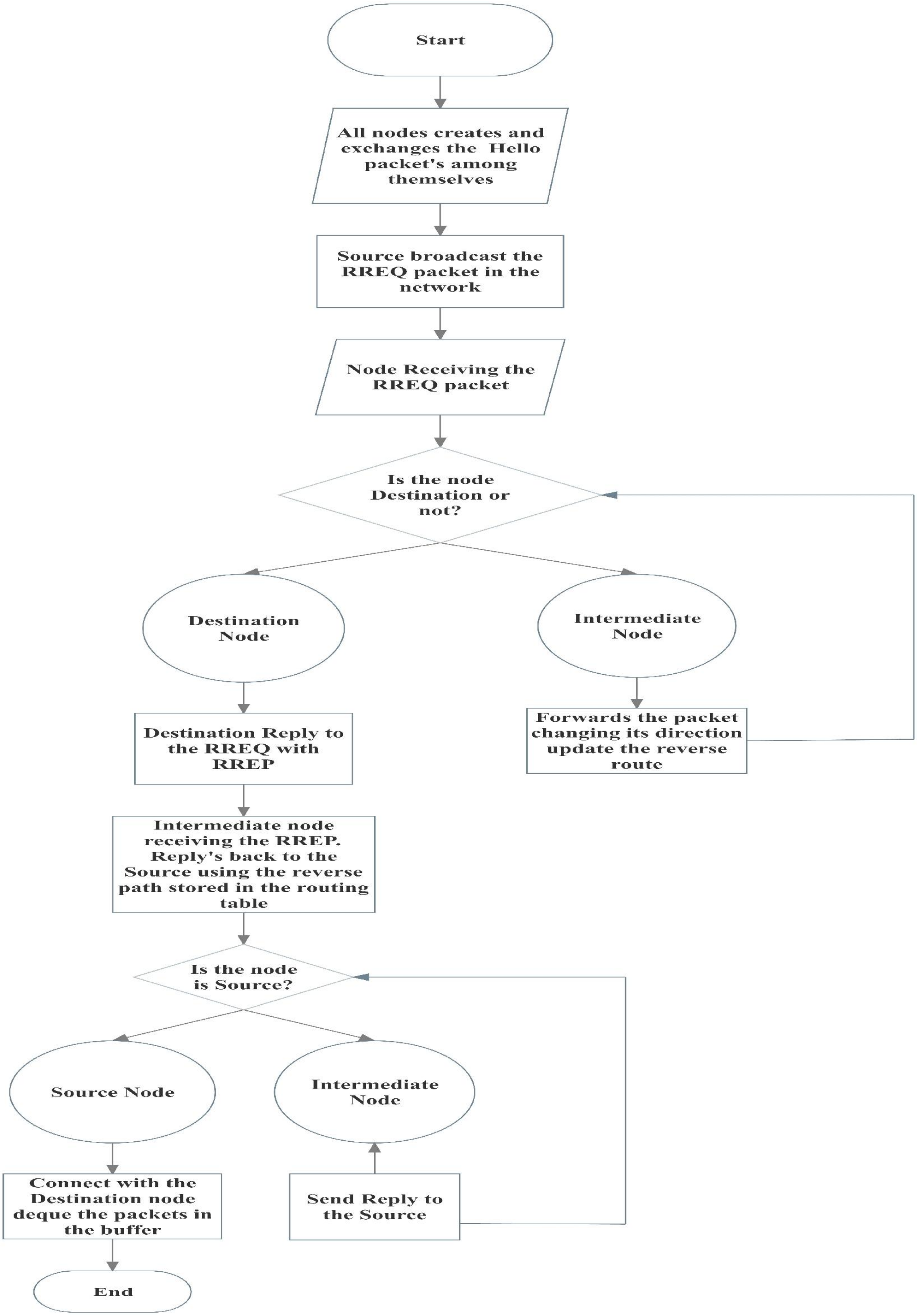

Fig. 7.3.1 AODV Working

8. PROPOSED EPAODV PROTOCOL

8.1 EPAODV

Energy Preserving Adhoc On-Demand Distance Vector (EPAODV) is an enhanced/improved routing protocol based on the AODV [12], [15], [16]. EPAODV improves on the total network life time by preserving the energies of nodes which is preserved by the selecting an optimized route for communication between the nodes.

The difference between EPAODV and AODV is that the route selection in AODV is based upon the hop count while in EPAODV the route selection is based upon the *Energy preserving value* (EPV). So the routing in EPAODV is based upon the EPV which is defined on remaining energy of nodes in the network that result in higher utilization of the nodes which have more energy as compared to the other nodes due to which the lower energy nodes are not utilized. The effect of this result in greater life time of network as compared to the AODV [15], [12], [16].

EPAODV is a loop free routing protocol as its uses sequence number for routing in the network. Whenever there are two routes for pair of source and destination node the route with the greater sequence number is selected and the rest is discarded. Network failure and error handling is similar to the AODV protocol. Route Requests (RREQ), Route Replies (RREP), and Route Errors (RERR) are the messages types defined by the EPAODV. EPAODV requires certain messages to be distributed widely in the entire adhoc network. As the two ends of communication i.e. source node and destination node have the valid route between them EPAODV don’t play any role between them.

In EPAODV Hello packets are enabled .The Hello packets are exchanged between the nodes in initial phase of the network setting up. These Hello packets carry the default values of AODV [16] [12] [15] as well the *Energy Preserving Value* (EPV) of each node in it. This EPV is saved in the reserved field of the AODV packet.

During the propagation of the packet in the network each node, receives RREQ destined for the node, forwards the packet again in the network and update the reverse route to the source based upon the *Energy Preserving value total* (EPVT) not on the hop count as in the AODV protocol[16] [12] [3].

Every node calculates its EPV value and stores it in its reserved field of the packets which it has to forward. The EPV is calculated as below.

Er = Remaining energy of node.

Emax = Maximum energy/initial energy of node.

Nbt = Total numbers of neighbors of a node.

EPV (node) = .8(Er/Emax) + .2*(Nbt)*

The above EPV value is based upon the probabilistic reasoning to define the fitness of the node based upon its remaining energy and the number of connected nodes.

EPV has the responsibility of choosing the route between the source and destination. The route selected by considering the EPV is energy efficient than the route discovered by the AODV protocol. Every node adds it EPV with the incoming EPV.

EPVT =EPV TOTAL

EPVT = EPV (incoming packet) + EPV (node).

EPVT is store in the remaining reserved field of the packet then forward the packet. The destination node receiving the RREQ sends RREP to the node. Following the strategy all the way creates a path from the destination to source which is based upon the EPVT.

The energy model used in this EPAODV is standard energy model available in the NS-2 [5] [4].The network starts identical to the AODV [16] [12] protocol by creating the Hello packets and sharing in the network. When the source node wants to transfer to destination it broadcast the RREQ messages in the network. The intermediate nodes receiving the RREQ update/create the reverse path to the source node based upon the EPVT not based upon the hop count. The updating of reverse route is done when a packet with greater sequence number or greater EPVT is obtained.

When the RREQ is reached the destination, it is replied with the RREP messages to the nodes from which it obtained the RREQ message. The intermediate node receiving the RREP is supposed to take the reverse route creating during the forwarding of the packets. The source getting the RREP from the destination flushes all data packet waiting in the queue to the destination. The updating of route from the source to destination is also based upon the sequence number, when a greater sequence number or greater EPVT is obtained the route is updated. Using EPVT for updating the route rather than the hop count is responsible for selecting the energy efficient route.

8.2 MESSAGE FORMATS

8.2.1 Route Request (RREQ) Message Format

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
Type								J	R	G	D	U	EPV & EPVT											Hop Count							
RREQ ID																															
Destination IP Address																															
Destination Sequence Number																															
Originator IP Address																															
Originator Sequence Number																															

Type: 1.

J: Join flag (reserved for multicasting).

R: Repair flag (reserved for multicasting).

G: Gratuitous RREP flag.

D: Destination only flag.

U: Unknown Sequence Number.

Reserved: Reserved field is divided into two parts. Part I stores the EPV and Part II stores the EPVT.

Hop Count: The required number of hops from originating node to destination.

RREQ ID: Used for identifying the RREQ uniquely.

Destination IP Address: The IP Address of destination.

Destination Sequence Number: The latest sequence received by the originator of destination.

Originator IP Address: IP Address of the node which originated RREQ.

Originator Sequence Number: The sequence number which leads to the originator.

8.2.2 Route Reply (RREP) Message Format

<table>
<tr><td colspan="10">0</td><td colspan="10">1</td><td colspan="10">2</td><td colspan="2">3</td></tr>
<tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td></tr>
<tr><td colspan="8">Type</td><td>R</td><td>A</td><td colspan="9">EPV & EPVT</td><td colspan="5">Prefix Size</td><td colspan="8">Hop Count</td></tr>
<tr><td colspan="32">Destination IP Address</td></tr>
<tr><td colspan="32">Destination Sequence Number</td></tr>
<tr><td colspan="32">Originator IP Address</td></tr>
<tr><td colspan="32">Lifetime</td></tr>
</table>

2.

Type:	
R:	Repair flag (used for multicast).
A:	Acknowledgment required.
Reserved:	Reserved field is divided into two parts. Part I stores the EPV and Part II stores the EPVT.
Prefix Size:	If nonzero, it specifies that the next hop may be used for any node with the same prefix size.
Hop Count:	The number of hops from destination to originator.
Destination IP Address:	IP Address of destination.
Destination Sequence Number:	The destination sequence number when RREP generated.
Originator IP Address:	The IP address of the node from where RREQ is broadcasted.
Lifetime:	The time (ms) for which destination considers the route valid.

8.2.3 Route Error (RERR) Message Format

<table>
<tr><td colspan="10">0</td><td colspan="10">1</td><td colspan="10">2</td><td colspan="2">3</td></tr>
<tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td></tr>
<tr><td colspan="8">Type</td><td>N</td><td colspan="16">Reserved</td><td colspan="7">DestCount</td></tr>
<tr><td colspan="32">Unreachable Destination IP Address</td></tr>
<tr><td colspan="32">Unreachable Destination Sequence Number</td></tr>
<tr><td colspan="32">Additional Unreachable Destination IP Addresses</td></tr>
<tr><td colspan="32">Additional Unreachable Destination Sequence Numbers</td></tr>
</table>

Type: 3.

N: No delete flag, when a node performed local repair of links, the upstream node should not delete flags.

DestCount: The number of unreachable destinations.

Unreachable Destination IP Address: No route for the specified destination.

Unreachable Destination Sequence Number: The sequence number of the unreachable destination IP address.

8.3 WORKING OF EPAODV

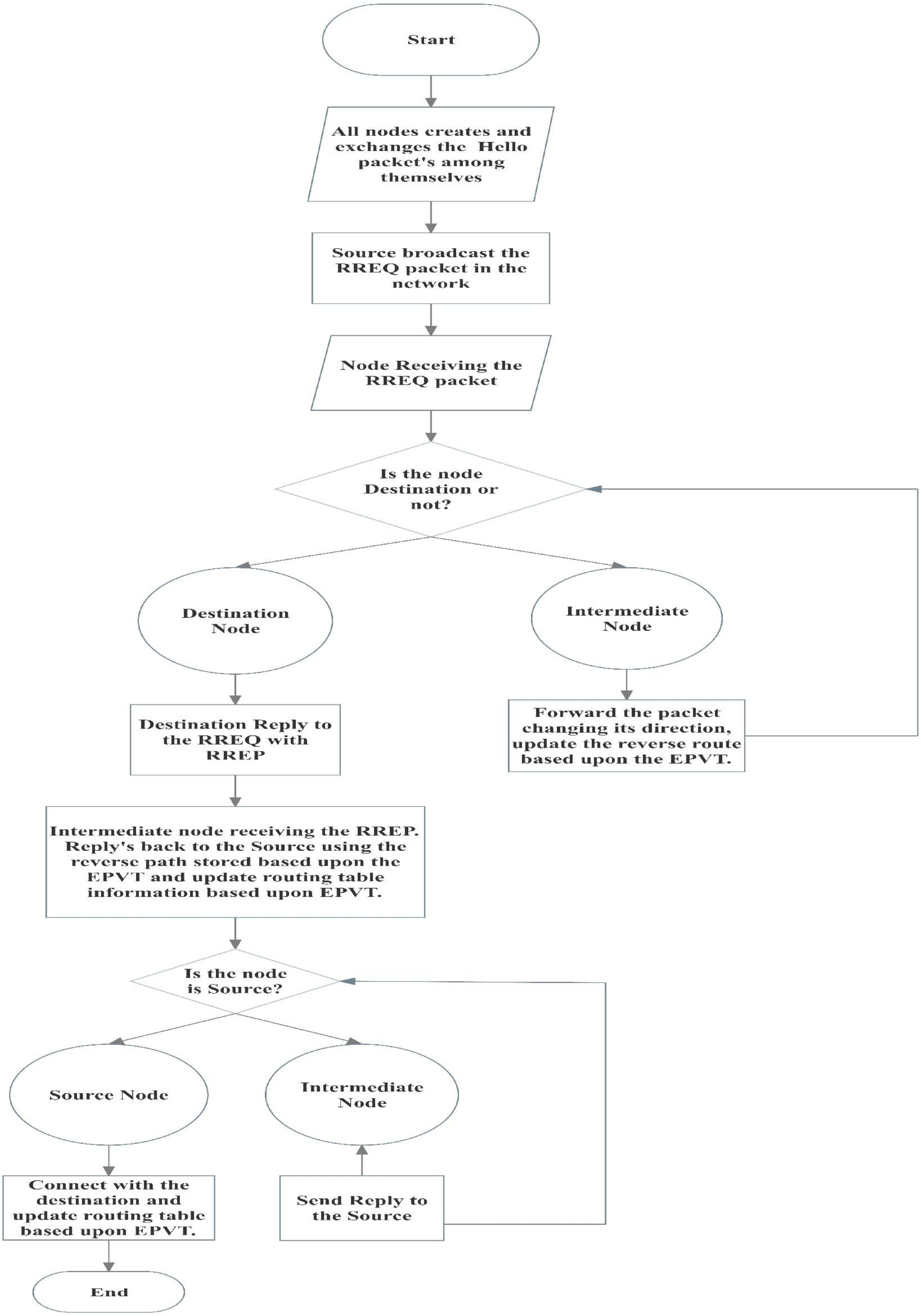

Fig. 8.3.1 Working EPAODV

8.4 SIMULATION OF EPAODV

In our simulation model, a network with 15 mobile nodes is placed in a rectangular region size of 500 x 500 units where 15 nodes are randomly and uniformly distributed in the region. To emulate the dynamic mobile environment, all of the nodes move around in their entire region. The initial energy of all the nodes is 100 joules. As in the energy model of NS-2 is in joules. The total running time of simulation of both EPAODV and AODV is around 250 milliseconds.

The following figure depicts the behaviour of both protocols over the total remaining energy in the network. X axis of the graph is for time in milliseconds and y axis is for the total remaining energy in the network.

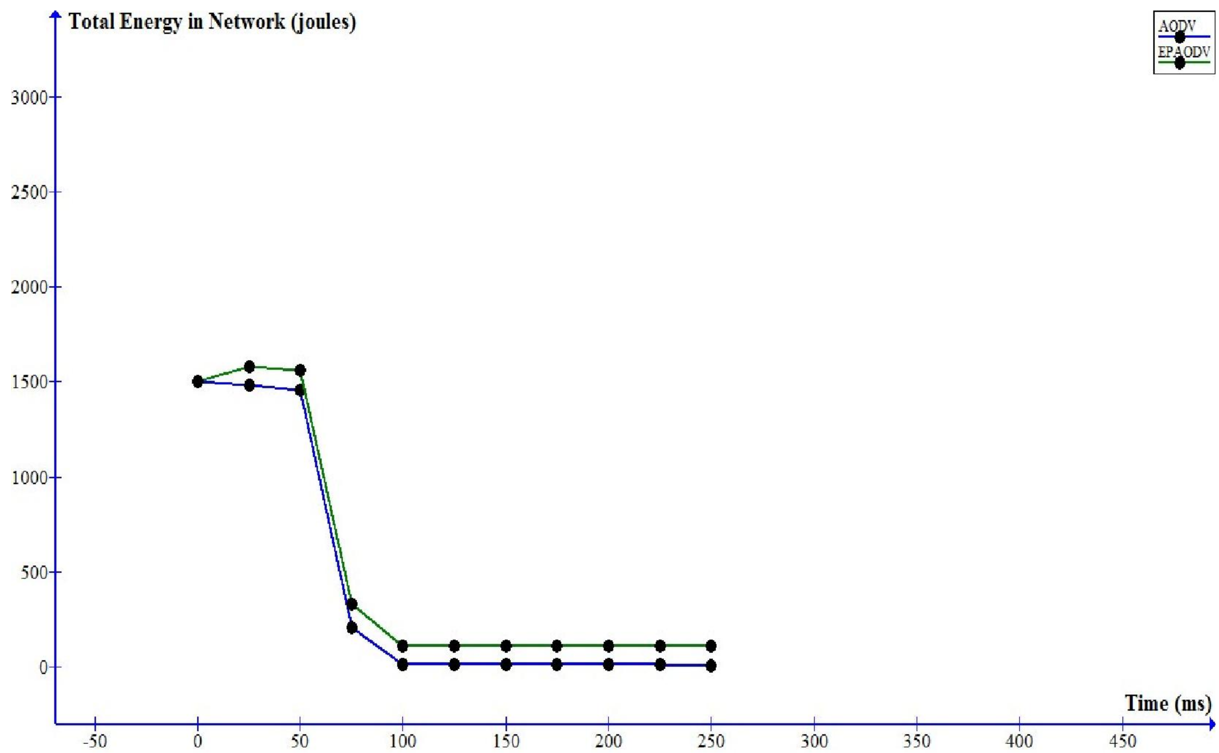

Fig. 8.4.1 AODV VS EPAODV (Total energy in network)

The following figure depicts the behaviour of both protocols about the data transferred i.e. when the source deque the packets in its buffer. X axis of the graph is for time in milliseconds and y axis is for the data transferred in the network.

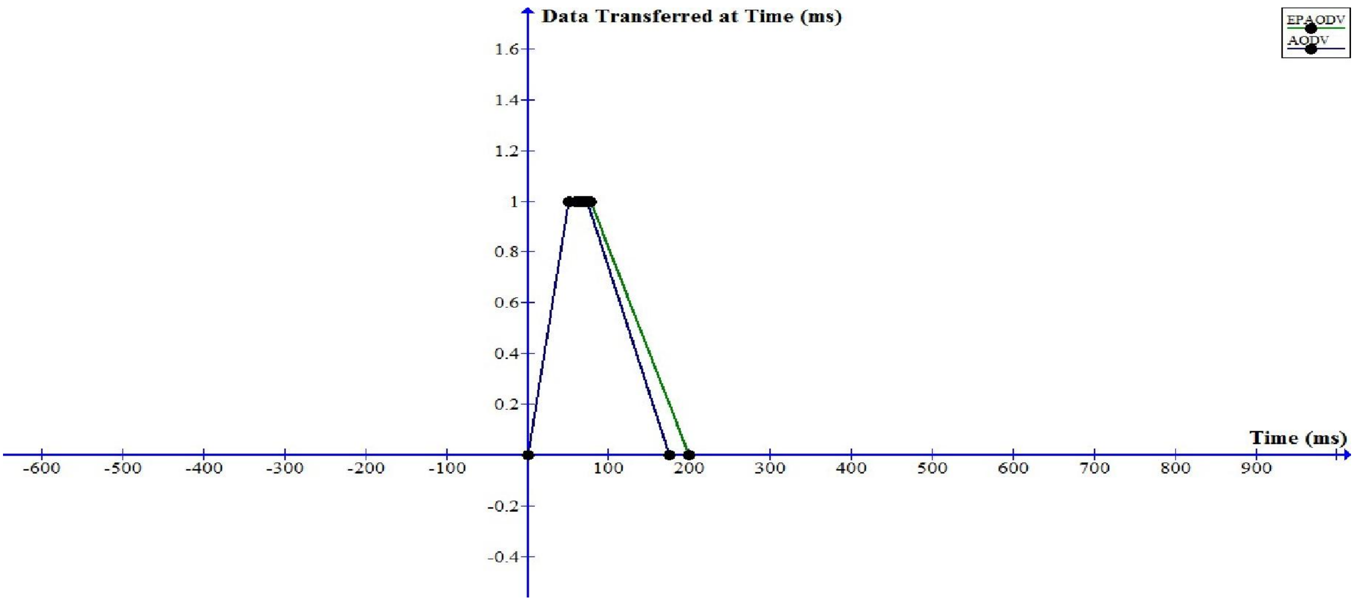

Fig. 8.4.2 EPAODV VS AODV (Data transferred in the network)

The following figure depicts the behaviour of both protocols about the data transferred i.e. when the source deque the packets in its buffer with the energy of the node which sends the data packet. Y axis of the graph is for energy of node in joules and X axis is for the data transferred in the network.

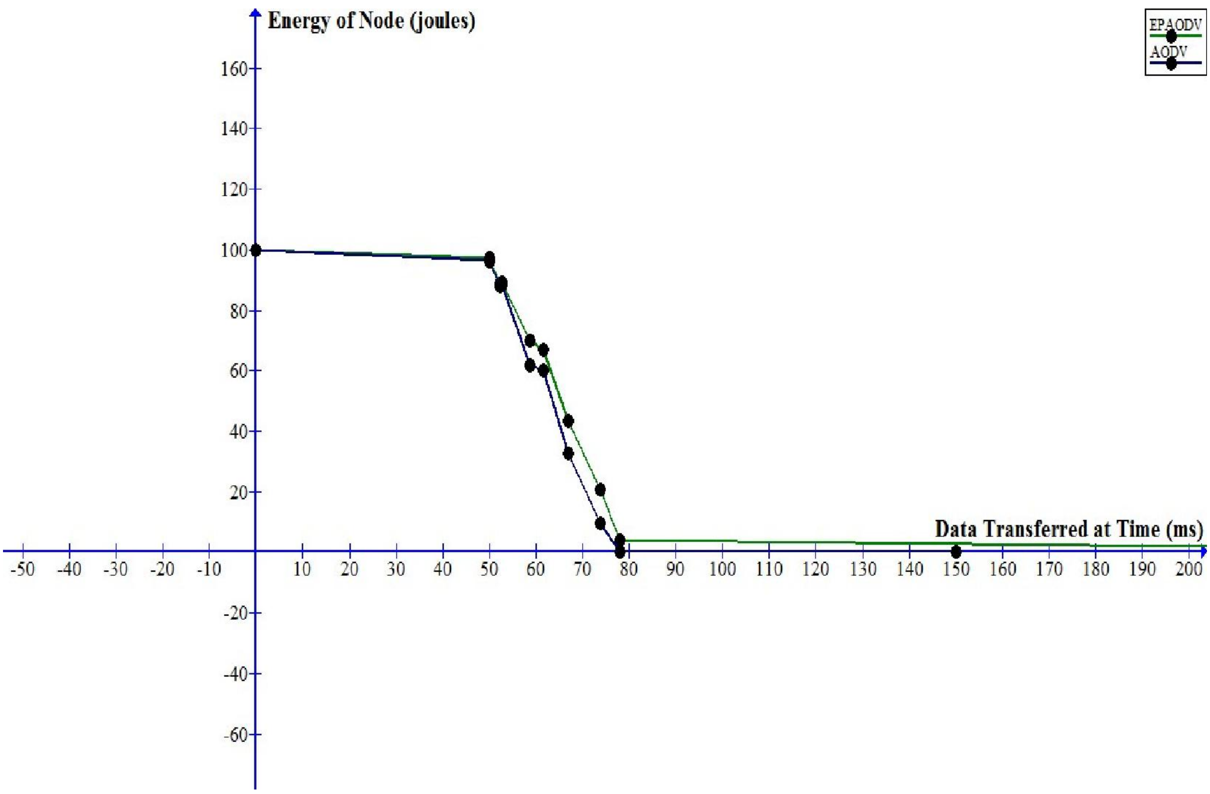

Fig. 8.4.3 EPAODV VS AODV (Energy with Data transferred)

The following figure depicts the behaviour of both protocols about the number of nodes alive in the network. X axis of the graph is for time in milliseconds and y axis is for the number of nodes alive in the network.

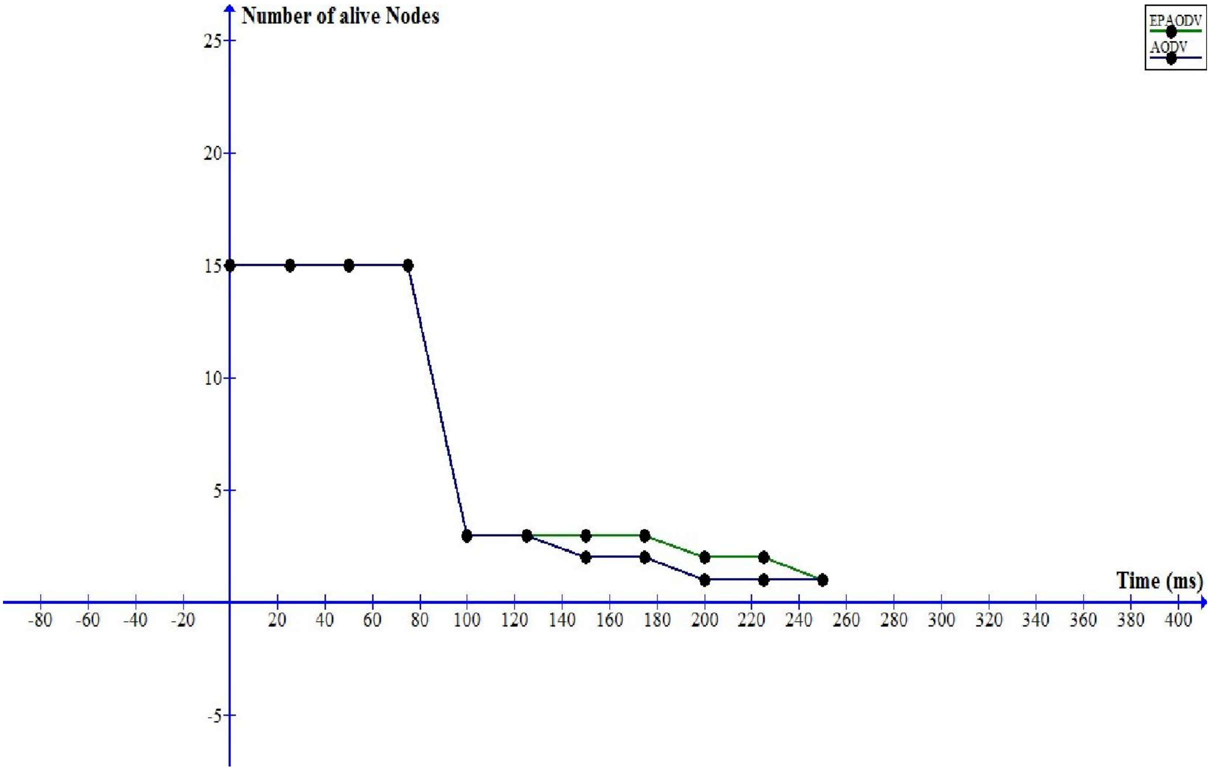

Fig. 8.4.5 EPAODV VS AODV (Number of alive nodes)

9. DSDV PROTOCOL

9.1 DSDV

Distance Sequenced Distance Vector (DSDV) [14] is a routing protocol developed for wireless adhoc network. It is an improvement over the distribute Bellman ford [7] by eliminating its looping drawbacks. The DSDV routing protocol works on the adhoc (for this purpose) network which has no wires and administrator. Adhoc network differ from the existing network by the way topologies changes frequently in adhoc network, secondly most user don't wish to get administrative permission to setup connection between themselves.

Each node in the DSDV maintains the designated to be used for the transfer to the particular node. Each node packet contains destination in the packet header which it forwards to the designated node which forwarded in the network until it reaches the destination node. The routing is based upon the distance vector [10].

In distance vector algorithm every node i maintain for each destination x a set of distances $\{d^x ij\}$ where j ranges over the neighbors of i. Node i treats neighbor k as next-hop for a packet destined for x if $d^x ij$ is $\min(d^x ij)$. The above distance vector algorithm is termed as classical Distributed Bellman Ford algorithm. However this can lead to the looping in the network because node uses the next hop for transfer who information could be stale or incorrect this problem is removed in the DSDV by using sequence number in the routing.

Packets are transmitted between the nodes of network using the routing table which are stored at each node. Each table entry is marked with unique sequence which is generated by the destination node. To maintain the consistency of the table each periodically transfers the updates within the network. Routes received by the receiver are also broadcasted by the receiver within the network in forward direction. Upon forwarding the packet it also add its own hop by incrementing the hop count by one.

A broken link in the network is defined as infinite value in the routing table which indicates that the node is unreachable and also every other which uses this node as a link to the broken node also sets the entry infinite in routing table. While generating the broken link information a new sequence number is added to the packet so to update as the link wakeup again.

In other to regulate the information transferred in the packet DSDV defines two types of packets one of which is “full dump” and the other is “incremental”. Full dump carry all the information available in the routing table while the other carry the changes from the last full dump packet. The incremental packet can be fitted in a single NPDU (Network protocol Data Unit) while full dump NPDU can use many NPDU for transferring the information of routing table.

The information broadcasted by each node contains the following data in them

- Destination Address.
- The required number of hops to reach the destination.
- The sequence number of the information received about the destination

9.2 DSDV PACKET

There are four C++ classes for maintenance of packets and packet headers in NS-2 which are Packet, packet_info, PacketHeader, and PacketHeaderManager. The Packet contains the definition of packets used in the simulation it s the sub class of event so that the packet could be scheduled. The packet_info contains the text representing the packet name. The PacketHeader facilitates the packet header creation and locating the packet in the network. The PacketHeaderManager defines the class which is helpful in managing and configuring the packet header.

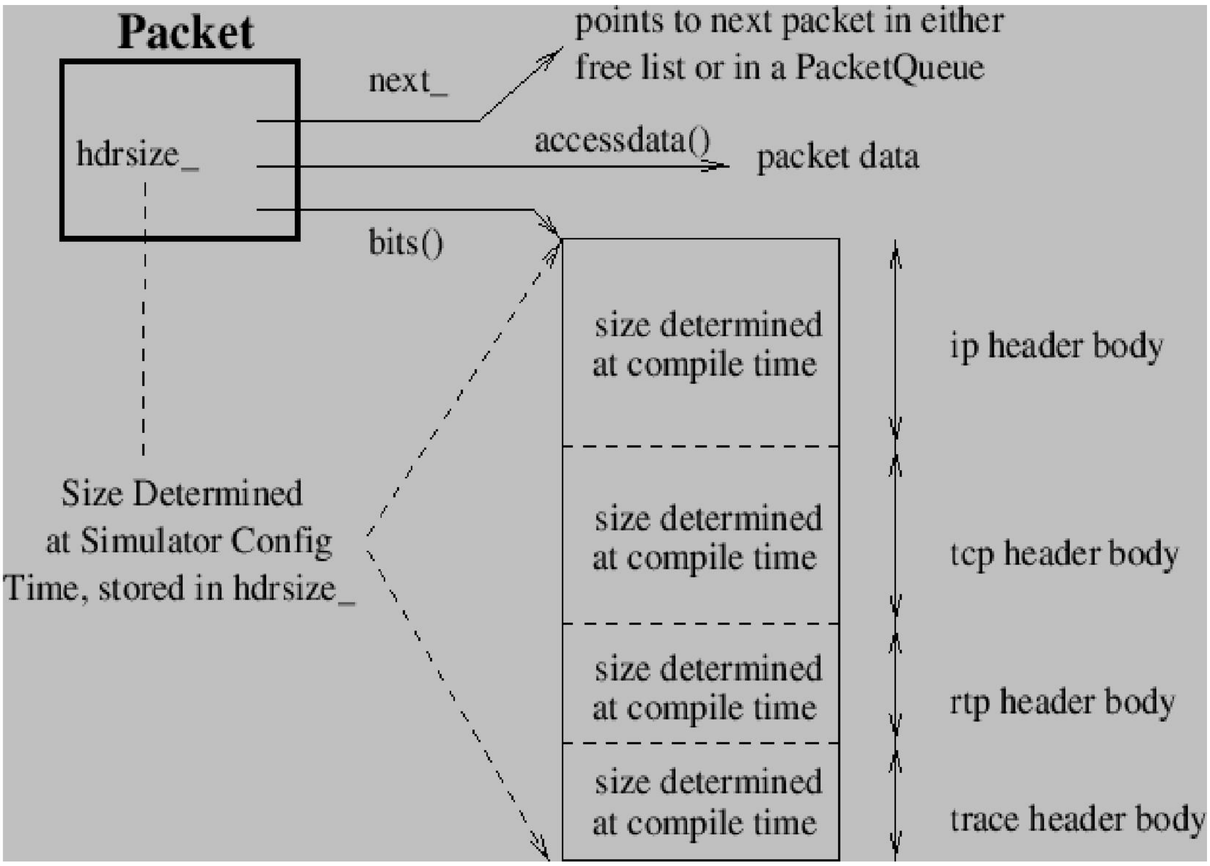

Fig. 9.2.1 A Packet class object

9.3 WORKING OF DSDV

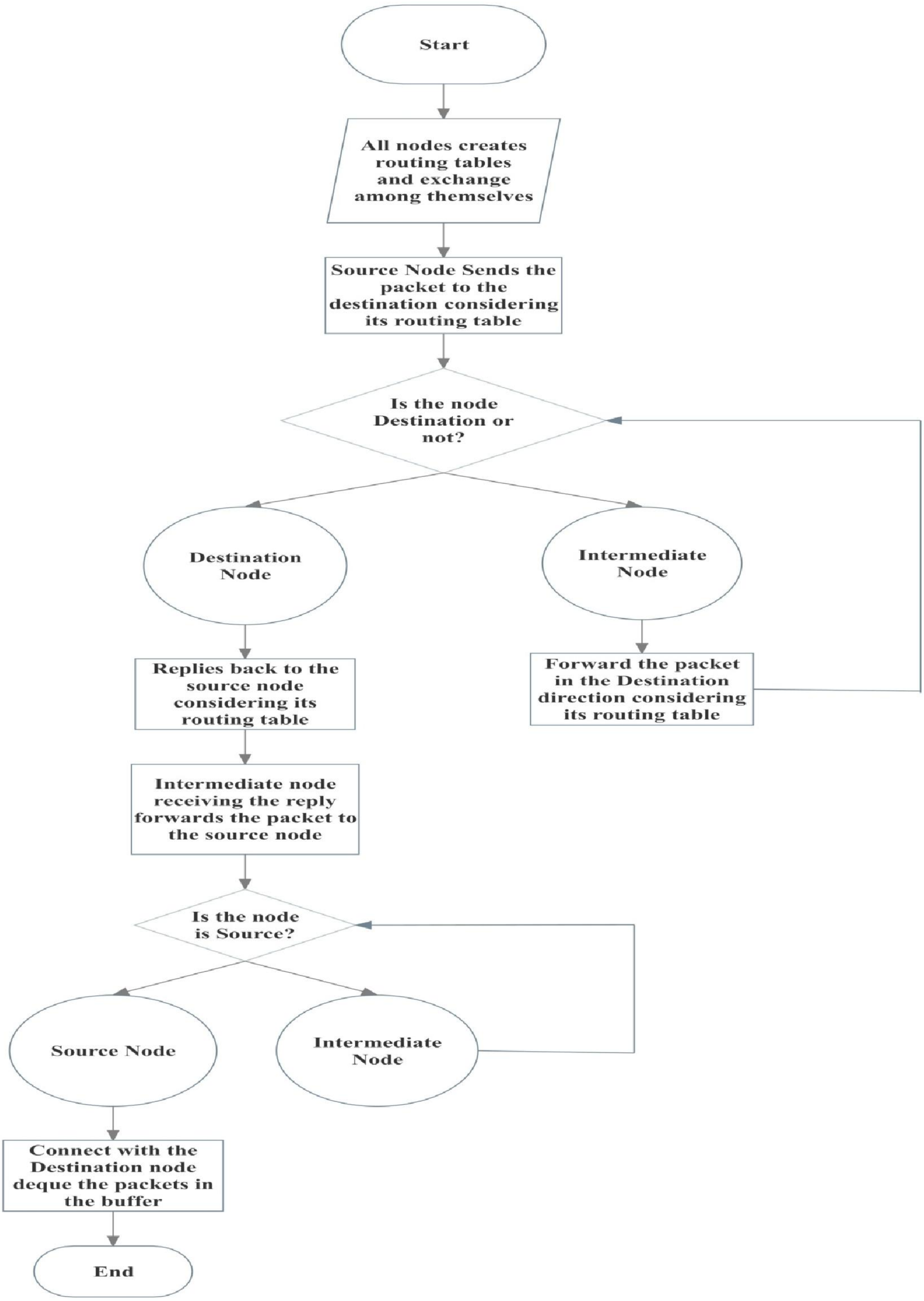

Fig. 9.3.1 Working of DSDV

10. PROPOSED EPDSDV PROTOCOL

10.1 EPDSDV

Energy Preserving Distance Sequenced Distance Vector (EPDSDV) is an enhanced/improved routing protocol based on the DSDV [14]. EPDSDV improves on the total network life time by preserving the energies of nodes which is preserved by the selecting an optimized route for communication between the nodes.

The difference between EPDSDV and DSDV is that the route selection in DSDV is based upon the hop count while in EPDSDV the route selection is based upon the *Energy preserving value* (EPV). So the routing in EPDSDV is based upon the EPV which is defined on remaining energy of nodes in the network that result in higher utilization of the nodes which have more energy as compared to the other nodes due to which the lower energy nodes are not utilized. The effect of this result in greater life time of network as compared to the DSDV [14].

EPDSDV is a loop free routing protocol as its uses sequence number for routing in the network. Whenever there are two routes for pair of source and destination node the route with the greater sequence number is selected and the rest is discarded. Network failure and error handling is similar to the DSDV protocol. EPDSDV requires certain messages to be distributed widely in the entire adhoc network.

In EPDSDV the nodes exchange the information with each as in DSDV but the metric chosen for routing is EPV rather than hop count which was used in DSDV. This EPV value is stored in the packet transferred in the network.

During the propagation of the packet in the network each node, receives full dump packet destined for the node, forwards the packet again in the network and update the routing table for the source based upon the *Energy Preserving value total* (EPVT) not on the hop count as in the DSDV protocol.

Every node calculates its EPV value and stores it in the packets which it has to forward. The EPV is calculated as below.

Er = Remaining energy of node.

Emax = Maximum energy/initial energy of node.

Nbt = Total numbers of neighbors of a node.

EPV (node) = .8(Er/Emax) + .2*(Nbt)*

The above EPV value is based upon the probabilistic reasoning to define the fitness of the node based upon its remaining energy and the number of connected nodes.

EPV has the responsibility of choosing the route between the source and destination. The route selected by considering the EPV is energy efficient than the route discovered by the DSDV protocol. Every node adds it EPV with the incoming EPV.

EPVT =EPV TOTAL

EPVT = EPV (incoming packet) + EPV (node).

EPVT is store in the packet then forward the packet. The destination node receiving the request sends the reply to the node from which it has received the request for the connection. Following the strategy all the way creates a path from the destination to source which is based upon the EPVT.

The energy model used in this EPAODV is standard energy model available in the NS-2 [5] [4].The network starts identical to the DSDV [14] protocol by creating the full dump packets and sharing in the network. When the source node wants to transfer to destination it sends the request packet for the destination considering its routing table message. The intermediate nodes receiving the request update/create the routing table to the source node based upon the

EPVT not based upon the hop count. The updating of routing table is done when a packet with greater sequence number or greater EPVT is obtained.

When the request is reached the destination, it is replied to the node from which it obtained the request message. The intermediate node receiving the reply is supposed to take the routing table entry creating during the forwarding of the packets. The source getting the reply from the destination flushes all data packet waiting in the queue to the destination. The updating of route from the source to destination is also based upon the sequence number, when a greater sequence number or greater EPVT with same sequence number is obtained the route is updated. Using EPVT for updating the route rather than the hop count is responsible for selecting the energy efficient route.

10.2 EPDSDV PACKET

There are four C++ classes for maintenance of packets and packet headers in NS-2 which are Packet, packet_info, PacketHeader, and PacketHeaderManager. The Packet contains the definition of packets used in the simulation it s the sub class of event so that the packet could be scheduled. The packet_info contains the text representing the packet name. The PacketHeader facilitates the packet header creation and locating the packet in the network. The PacketHeaderManager defines the class which is helpful in managing and configuring the packet header.

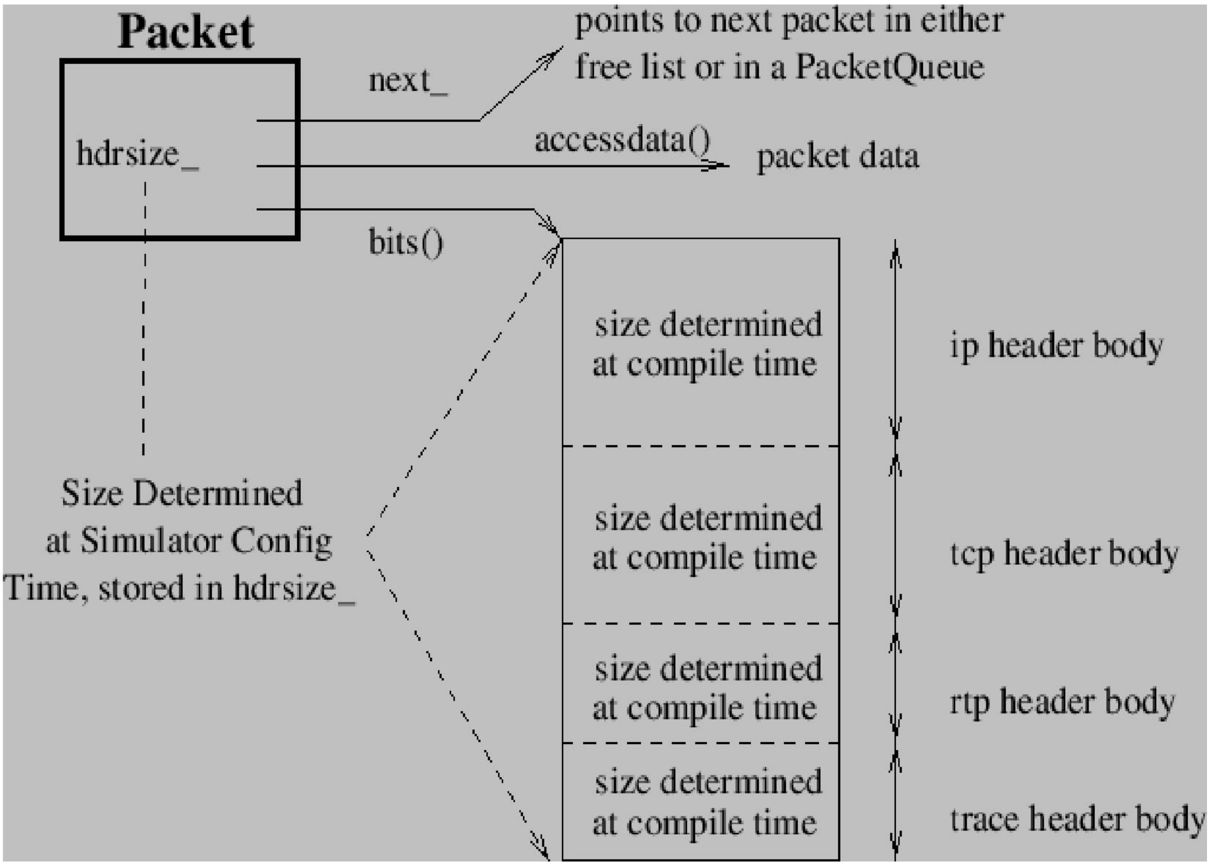

Fig. 10.2.1 A Packet class object

10.3 WORKING OF EPDSDV

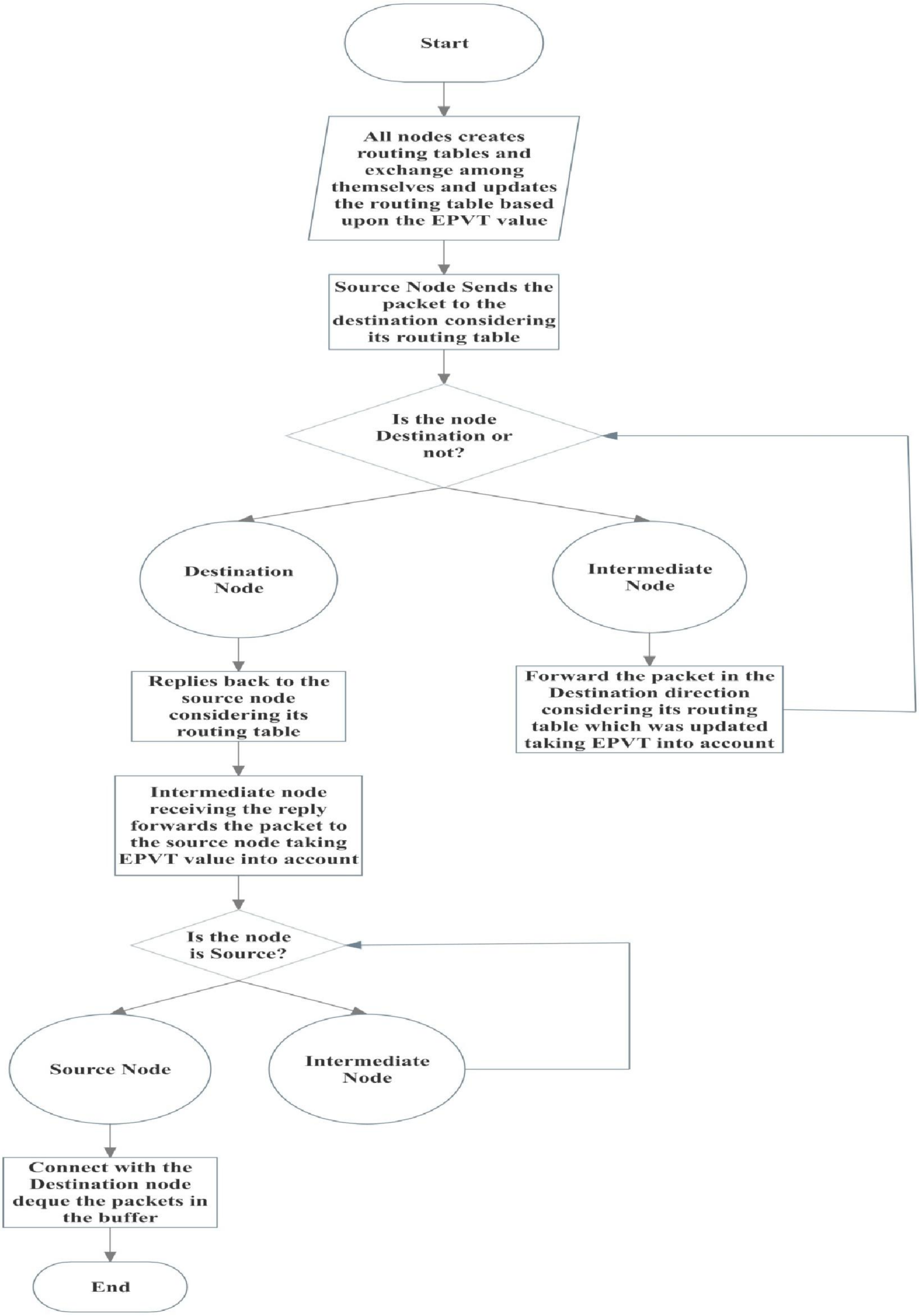

Fig. 10.3.1 Working of EPDSDV

10.4 SIMULATION OF EPDSDV

In our simulation model, a network with 15 mobile nodes is placed in a rectangular region size of 500 x 500 units where 15 nodes are randomly and uniformly distributed in the region. To emulate the dynamic mobile environment, all of the nodes move around in their entire region. The initial energy of all the nodes is 100 joules. As in the energy model of NS-2 is in joules. The total running time of simulation of both EPDSDV and DSDV is around 250 milliseconds.

The following figure depicts the behaviour of both protocols over the total remaining energy in the network. X axis of the graph is for time in milliseconds and y axis is for the total remaining energy in the network.

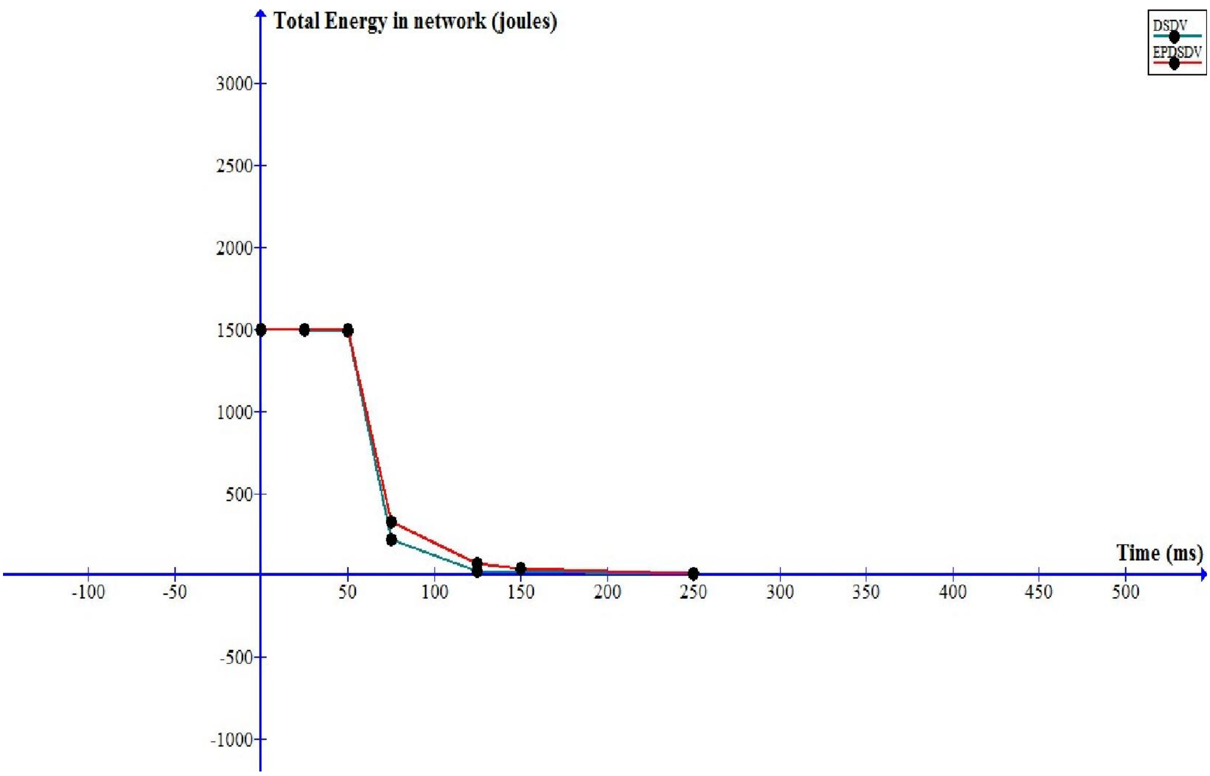

Fig. 10.4.1 DSDV VS EPDSDV (Total energy in network)

The following figure depicts the behaviour of both protocols about the data transferred i.e. when the source deque the packets in its buffer. X axis of the graph is for time in milliseconds and y axis is for the data transferred in the network.

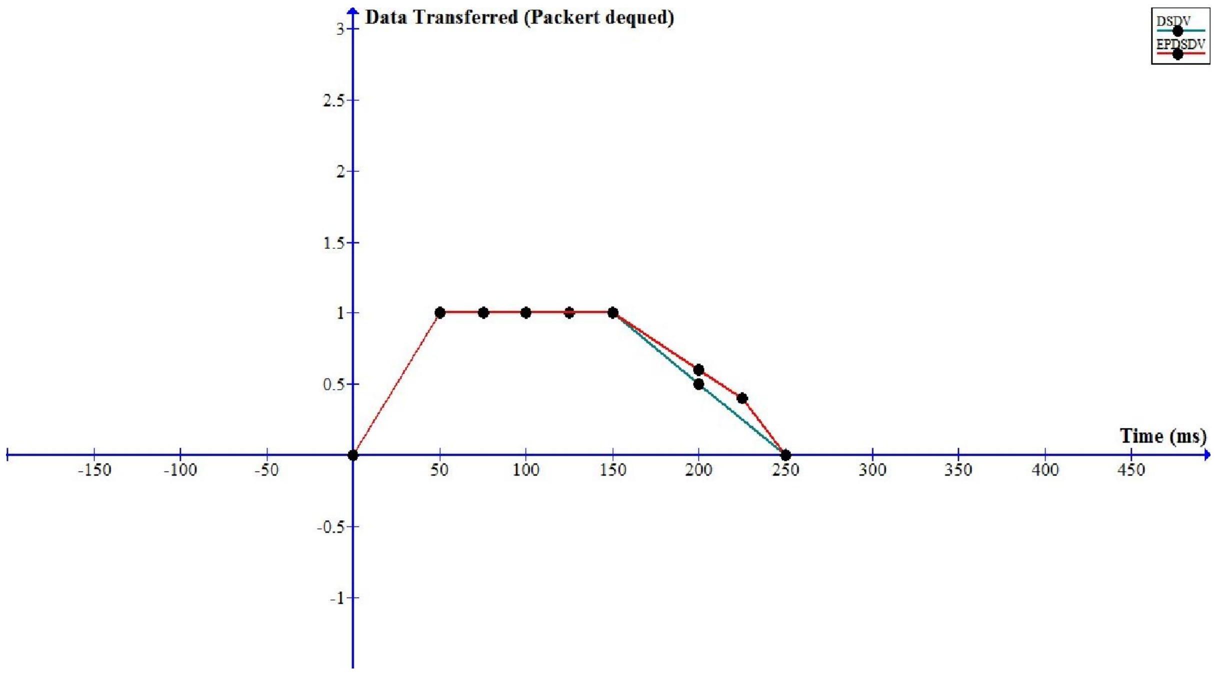

Fig. 10.4.2 EPDSDV VS DSDV (Data transferred in the network)

The following figure depicts the behaviour of both protocols about the data transferred i.e. when the source deque the packets in its buffer with the energy of the node which sends the data packet. Y axis of the graph is for energy of node in joules and X axis is for the data transferred in the network.

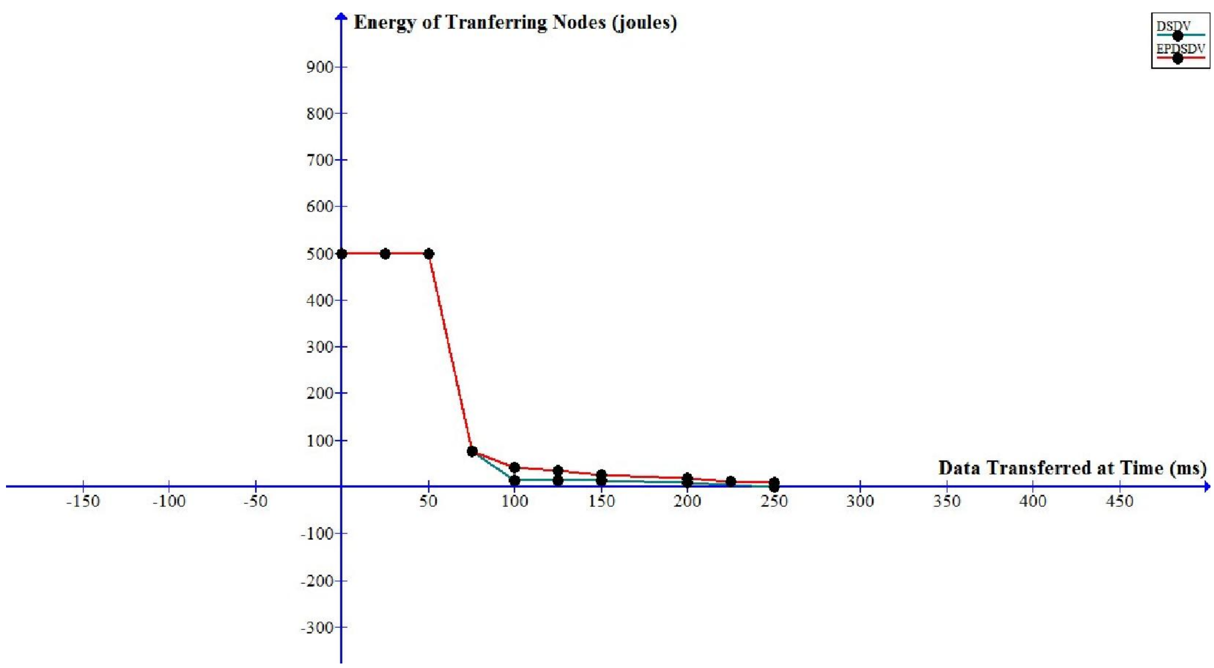

Fig. 10.4.3 EPDSDV VS DSDV (Energy with Data transferred)

The following figure depicts the behaviour of both protocols about the number of nodes alive in the network. X axis of the graph is for time in milliseconds and y axis is for the number of nodes alive in the network.

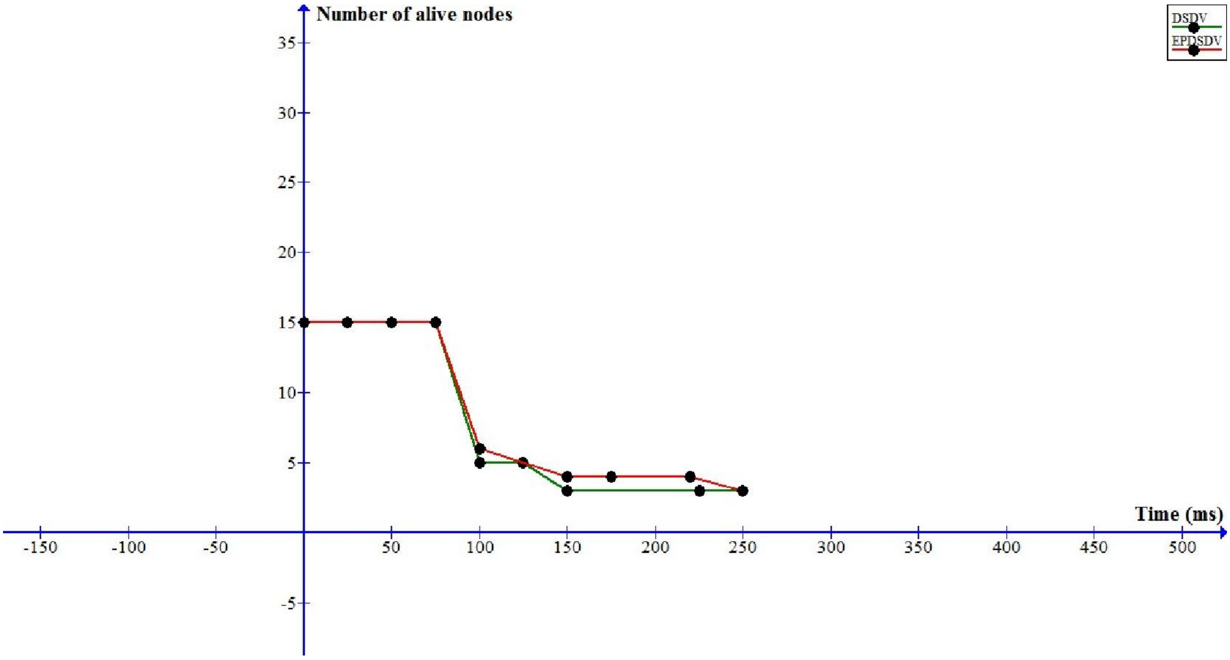

Fig. 10.4.4 EPDSDV VS DSDV (Number of alive nodes)

11. CONCLUSION

In this it was found that the using the EPVT for routing in the EPAODV and EPDSDV produces better network lifetime [1], [2], [3] than their respective parents routing protocols i.e. AODV and DSDV. The number of alive node over the time is increased and the total remaining energy of the nodes is also increased. Throughput shows a minute decrement as compared to their parent protocols which was expected before the simulation as the alternative route are preferred than the minimum hop count which forces them to take longer routes that result to take more time to perform specified data transfer but the decremented is too minute that it can be ignored as the link speed is high.

12. FUTURE WORK

In further enhancements of the proposed EPAODV and EPDSDV link transmission speed and the link breaking error could also be considered as the metric to improve the route selection criteria which could provide better results. Considering the link breaking error improves the reliability of the protocol. Taking link transmission speed as metric fused with EPVT will provide the better throughput of the proposed protocol. Further the EPVT is based upon the probabilistic reasoning this can be improved by generating the EPVT values based upon the previously stored in a database which should be consistent and provides the improved EPVT. A centralized node can be used by making the energy efficient node as a central node which transfers the packets in the network; the central node is updated periodically based upon the energy.

REFERENCES

[1] S. Corson, J. Macker, “Mobile Ad hoc Networking(MANET),” Internet Draft, Oct. 1998.

[2] Tseng, Yu-Chee ; Hsieh, Ten-Yueng, “Fully power-aware and location-aware protocols for wireless multi-hop ad hoc networks,” Computer Communi-cations and Networks.Proceedings. Eleventh International, 2002.

[3] Jin-Man Kim, Jong-Wook Jang, “Proceedings of the Advanced InternationalConference on Telecommunications and International Conference on Internet and Web Applications and Services ”,2006.

[4] WIKIPEDIA “Network Simulator”

[5] ISI "The Network Simulator : ns-2 http://www. isi.edu/nanam/ns/, University of Southern Califor-nia

[6] “The C++ Programming Language”, B. Stroustrup.

[7] Bellman, Richard. "On a routing problem". *Quarterly of Applied Mathematics* 16: 87–90. MR 0102435.1998

[8] S. Corson, J. Macker, “Mobile Ad hoc Networking(MANET),” Internet Draft, Oct. 1998.

[9] Volkan Rodoplu and Teresa H. Meng, “Minimum energy mobile wireless networks,” in Proceedings of the 1998 IEEE International Conference on Communications, ICC’98, Atlanta, GA, June 1998.

[10] "RFC1058 - Routing Information Protocol", C. Hedrick, Internet Engineering Task Force, June 1988.

[11] http://www.mathcs.emory.edu/~cheung/Courses/455/Syllabus/5a-routing/distance-vec3.html

[12] Dr.A.A.Gurjar, ,Mr.A.A.Dande "International Journal Of Engineering And Science Issn: 2278-4721, Vol. 2, Issue 3 February 2013.

[13] D. B. Johnson and D. A. Maltz. Dynamic source routing in ad hoc wireless networks. In T. Imielinski and H. Korth, editors,Mobile Computing,volume 353, chapter 5, pages 153–181. Kluwer Academic Publishers, Boston, 1996.

[14] C. Perkins and P. Bhagwat. Highly DynamicDestination-Sequenced Distance-Vector Routing(DSDV) for Mobile Computers. In ACM SIGCOMM'94 Conference on Communications Architectures, Protocols and Applications, volume 24,pages 234–244, New York, NY, USA, 1994.

[15] C. K. Toh. Ad Hoc Mobile Wireless Networks:Protocols and Systems. Prentice Hall, December 2001.

[16] C. Perkins and E. M. Royer. Ad–hoc on-demand distance vector routing. InProceedings of 2nd IEEE Workshop on Mobile Computing Systemsand Applications, pages 90–100, New Orleans,LA, February 1999.

[17] Z. J. Haas. A new Routing Protocol for Reconfigurable Wireless Networks. InProceedings of IEEE International Conference on Universal Personal Communication, pages 1–11, Atlanta, GA, October 1997.

[18] T. Chen and M. Gerla. Global state routing: A new routing scheme for ad-hoc wireless networks. InProceedings of the IEEE international conference on communications (ICC), pages 171–175, Atlanta, GA, June 1998.

[19] RFC 3626.

[20] Dijkstra, E. W. "A note on two problems in connexion with graphs" 1959.

[21] http://www.isi.edu/nsnam/vint/