Energy Efficient Clustering-Based Multipath Routing in Mobile Ad Hoc Networks

Khaja Mahabubullah[1], Dr. Syed Abdul Sattar[2]
[1]*Ph.D Scholar (PT), Dept. of Computer Science, Rayalaseema University, Kurnool (A.P)*
[2]*Professor and Principal, Nawab Shah Alam Khan College of Engineering and Technology, Hyderabad. (T.S.)*

Abstract— Clustering is an important research topic for mobile ad hoc networks (MANETs) because clustering makes it possible to guarantee basic levels of system performance, such as throughput and delay, in the presence of both mobility and a large number of mobile terminals. The paper presents Energy Efficient Clustering-Based Multipath Routing (EECMR) in Mobile Ad Hoc Networks. The proposed algorithm uses optimization technique for the minimization of Cluster-Heads (CH) based on four parameter such as Degree-difference, sum of distance to all neighbors, mobility of node and battery power of node. In this algorithm each node is calculated its weight taking into account all the above parameters. The node with the smallest weight is selected as the clusterhead and transmits the data packets of its cluster numbers using well-known multipath routing protocol Ad hoc On-demand Distance Vector Multipath Routing (AODVM). We compared the performance of our algorithm with that of Energy Aware Clustered-Based Multipath Routing (EACMR) terms of the number of clusterhead change, the number of re-affiliations, the control overhead, throughput, energy consumption, and packet delivery ratio, end-to-end average delay of packets and lifetime of network. The results have proved the superior performance of the proposed algorithm over the existing clustering algorithm EACMR.

Keywords— *Clustering, MANET, Multipath, AODVM, EACMR, EECMR (key words)*

I. INTRODUCTION

With the advance of wireless communication technologies, small-size and high-performance computing and communication devices are increasingly used in daily life. After the success of second generation mobile system, more interest was started in wireless communications. A Mobile Ad hoc Network (MANET) is a wireless network without any fixed infrastructure or centralized control, it contains mobile nodes that are connected dynamically in an arbitrary manner. The Mobile Ad hoc Networks are essentially suitable when infrastructure is not present or difficult or costly to setup or when network setup is to be done quickly within a short period. They are very attractive for tactical communication in the military and rescue missions. They are also expected to play an important role in the civilian for as convention centers, conferences, and electronic classrooms. They provide network service even where it is difficult to deploy a fixed infrastructure at a very low cost. It also cases the task of relocation and other notifications to the network structure. The clustering is an important research area in mobile ad hoc networks because it improves the performance of flexibility and scalability when network size is huge with high mobility. All mobile nodes operate on battery power, hence, the power consumption becomes an important issue in Mobile Ad hoc Networks[1][2]

A fundamental problem in the MANET is how to deliver data packets among nodes efficiently without predetermined topology or centralized control, which is the main objective of ad hoc routing protocols. Because of the dynamic nature of the network, ad hoc routing faces many unique problems not present in wired networks. Particularly in MANETs where routes become obsolete frequently because of mobility and poor wireless link quality. The Multipath routing addresses these problems by providing more than one route to a destination node. Multipath routing appears to be a promising technique for ad hoc routing protocols. The multiple paths can be useful in improving the effective bandwidth of communication, responding to congestion and heavy traffic, increasing delivery reliability and security. The traffic can be distributed among multiple routes to enhance transmission reliability, provide load balancing, and secure data transmission [2].

Most existing routing protocols for MANET build and utilize only one single route for each pair of source and destination nodes. Due to node mobility, node failures, and the dynamic characteristics of the radio channel, links in a route may become temporarily unavailable and making the route invalid. The overhead of finding alternative routes may be high and extra delay in packet delivery may be introduced. Multipath routing addresses this problem by providing more than one route to a destination node. Source and intermediate nodes can use these routes as primary and backup routes. Alternatively, source node can distribute traffic among multiple routes to enhance transmission reliability, provide load balancing, and secure data transmission. Hence combination of clustering and multipath routing is efficient technique in MANETs[2]

2. Related Work

Proactive protocols have large overhead and less latency as it is vice versa in case of reactive. Hybrid routing protocol is introduced to bridge the pitfalls. Hybrid routing protocol owns the advantages of Proactive Routing as well as Reactive Routing protocols. It has the table maintenance mechanism of proactive protocol and route discovery mechanism of reactive protocol to overcome latency and overhead problems in the network. Hybrid routing protocol is widely used protocol in Networks consisting of huge number of nodes. Large networks are divided as different zones where routing inside each zone is performed by proactive approach and outside the zone routing is done using reactive approach[1]

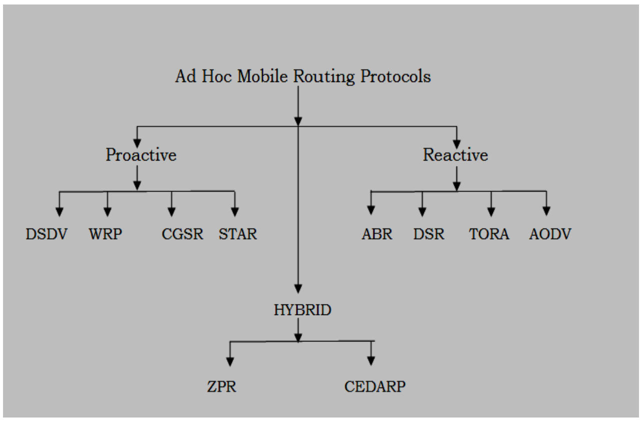


Fig 1. Classification of Routing Protocols

Clustering is an important research topic for mobile ad hoc networks (MANETs) because clustering makes it possible to guarantee basic levels of system performance, such as throughput and delay, in the presence of both mobility and a large number of mobile terminals. A large variety of approaches for ad hoc clustering have been presented, whereby different approaches typically focus on different performance metrics. It provides descriptions of the mechanisms, evaluations of their performance and cost, and discussions of advantages and disadvantages of each clustering scheme [1]. 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. 2. 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[3][4]

2.1 Advantages of Clustering

Clustering in Ad Hoc networks has many advantages compared to the traditional networks. They are as follows[4]
1) It allows the better performance of the protocol for the Medium Access Control (MAC) layer by improving the spatial reuse, throughput, scalability and power consumption.
2) It helps to improve routing at the network layer by reducing the size of the routing tables.
3) It decreases transmission overhead by updating the routing tables after topological changes occur.
4) It helps to aggregate topology information as the nodes of a cluster are smaller when compared to the nodes of entire network. Here each node stores only a fraction of the total network routing information.
5) It saves energy and communication bandwidth in ad-hoc networks

2.2 Issues of Clustering

The highly dynamic and unstable nature of MANET's makes it difficult for the Cluster based routing protocol to divide a mobile network into clusters and determination of cluster heads for each cluster. Clustering reduces communication and control overheads due to pre-determined paths of communication through cluster heads. It is vital for scalability of media access protocols, routing protocols and the security infrastructure [3][4][5]

Routing protocols which considers only bidirectional links may have link asymmetry due inefficient or abnormal routing. Untapped network capacity is represented by the undiscovered unidirectional links, which reduces the network connectivity [1][4].A large numbers of mobile terminals are managed by a MANET using a cluster topology. The construction and maintenance of a cluster structure requires additional cost compared with a topology control without cluster.

2.3 Clustering has some side effects and drawbacks

1) The maintenance cost for a large and dynamic mobile network requires explicit message exchange between mobile node pairs. As the network topology changes quickly and concerns many mobile nodes, the number of information message exchange grows to reach a critical point. This information exchange consumes a lot of network bandwidth and energy in mobile nodes.
2) A ripple effect of re-clustering occurs if any local events take place like the movement or the death of a mobile node, as a result it may lead to the re-election of a new cluster-head. When a new cluster-head is re-elected it may cause re-elections in the whole of the cluster structure. Thus, the performance of upper-layer protocols is affected by the ripple effect of re-clustering.
3) One of the major drawbacks of clustering in MANETs is that some nodes consume more power when compared to others nodes of the same cluster, as special node like a cluster-head or a cluster-gateway manage and forward all messages of the local cluster their power consumption.

2.4 Classification of Clustering Algorithms

The clustering algorithms are classified in to seven categories [3]

- DS based Clustering
- Low Maintenance Clustering

- Mobility Aware Clustering
- Energy-efficient clustering
- Load-balancing clustering
- Combined-metrics-based Clustering(Weighted Clustering)
- Secure Clustering

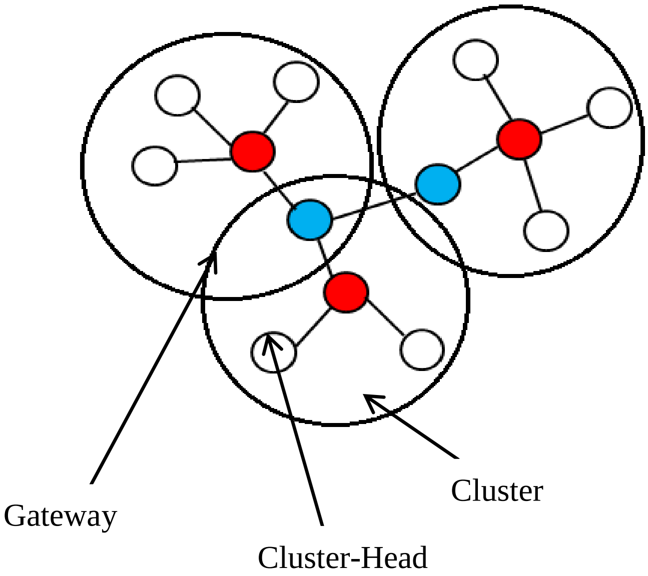


Fig. 2. Cluster structure illustration.

Different types of weighted based clustering algorithms have been surveyed [6][7] some of them are An Efficient Weight-based clustering algorithm, Stable and Flexible Weight based Clustering Algorithm, Hierarchical Weighted Clustering Algorithm optimized, Robust Clustering Algorithm, Load-Balancing and Weighted Clustering Algorithm, Weight Based Clustering Algorithm, Adaptive Weighted Cluster Based Routing Protocol (AWCBRP)[8] and WCA with Mobility Prediction, An On-Demand Weighted Clustering Algorithm (WCA)[9].

Several load balancing clustering algorithms have been proposed , Energy Aware Load Balancing Clustering[10] , Load-Balancing Clusters [11], Energy-efficient clustering in mobile ad-hoc networks using multi-objective particle swarm optimization [12].

The Energy efficient clustering avoids unnecessary energy consumption or balancing energy consumption for mobile nodes in order to prolong the lifetime of mobile nodes and network.

The on-demand routing is the most popular approach in the MANET. Instead of periodically exchanging route messages to maintain a permanent route table of the full topology, the on-demand routing protocols build routes only when a node needs to send the data packets to a destination. The standard protocols of this type are Dynamic Source Routing (DSR)[13] and the Ad hoc On-demand Distance Vector(AODV) routing[14]. However, these protocols do not support multipath.

The several multipath on-demand routing protocols were proposed, some of the standard protocols are, the Ad hoc On-demand Multipath Distance Vector (AOMDV)[15] is an extension to the AODV protocol for computing multiple loop-free and link-disjoint paths. The Split Multipath Routing (SMR) [16] is an on-demand Multipath source routing protocol can find an alternative route that is maximally disjoint from the source to the destination. The Multipath Source Routing (MSR) [17] is an extension of the DSR protocol to distribute traffic among multiple routes in a network. The Ad hoc On-demand Distance Vector Multipath Routing (AODVM)[18] is an extension to the AODV for finding multiple node disjoint paths. These protocols build multiple routes based on demand but they did not consider energy.

Several energy aware multipath on-demand routing protocols have been proposed [19]- [25], but these protocols are not based on clustering. The Cluster Based Multipath Dynamic Source Routing CMDSR)[15] was designed to be adaptive according to network dynamics. It uses the hierarchy to perform route discovery and distributes traffic among diverse multiple paths, but it does not consider energy in the route selection.

The Grid-Based Energy Aware Node-Disjoint Multipath Routing Algorithm (GEANDMRA)[26] considers energy aware and node-disjoint multipath**,** it uses grid head election algorithm to select the grid-head which is responsible for forwarding routing information and transmitting data packets. The routing is performed in a grid-by-grid manner, the network area is partitioned into non-overlapping square zones with the same size, each zone is named with a unique ID(x,y) as the conventional coordinate. At any time each node can obtain its location information from Global Position System (GPS) to know in which zone it is located. But the disadvantages of GPS exist.

Energy Aware Clustered Based Multipath Routing, it is based on the Combined Higher Connectivity Lower ID (CONID)clustering algorithm [2] is used generates the clusters in the network and uses the Ad hoc On-demand Distance Vector Multipath Routing (AODVM)[18] for transmission of data packets. Main drawback this protocol doesn't consider the optimization clusters based network's size. It considers only one-hop clusters. If network size increases number of clusters also increases. Due this routing overhead increases, routing table size increases, bandwidth and energy are wasted. To overcome this drawback EECMR is proposed.

3. Energy Efficient Clustering-Based Multipath Routing

This protocol combination power efficient clustering algorithm and Ad hoc On-demand Distance Vector Multipath Routing (AODVM). It forms optimization of clusters and uses energy aware node-disjoint multipath routing for transmission of data packets.

The HELLO messages are already in usage in on demand routing, each node maintains only the status of its neighbors. Periodically, each node broadcasts a hello message to all its neighbors to indicate its active status. Each node receives the hello messages from its neighbors and updates its neighbor information table**.** Inthe EEACMR, the clusters are formed by using hello messages, this result in less overhead. The EEACMR finds node-disjoint multiple routes from a source to destination and increases the network life time by using optimal routes.

3.1 Cluster formation in EEACMR

Before going into detail of the steps of the algorithm we describe the major parameters involved in the working of the algorithm. We consider a communication network which is represented by an undirected graph G=(V,E) where V is the set of nodes and E is the set of bi-directional communication links. The clustering process divides V into a collection of subsets {V_1, V_2.......,V,}

Where $V = \bigcup_{i=1}^{k} V_i$

Such that each subset V_i induces a connected sub graph of G. It should be noted that these sub graphs can overlap and each sub set is a cluster. EEACM uses four parameters mobility, Transmission Power, Degree difference and Transmission rate are described as follows

3.1.1 Parameters for computing weight factor of node v

i. Degree Difference: In cluster-based structure a performance parameter for load balancing is *degree difference* (Δv) [1], for each node *v* which is defined as the difference of ideal node degree (δ) and actual degree (connectivity) of that node. Degree of node (*dv*) is the number of neighbors of node *v* that are in the transmission range. Ideal degree is the number of neighbors that a clusterhead can handle effectively.

Degree Difference $\Delta_v = |d_v - \delta|$ (1)

ii. Energy Consumption: Clusterhead has to perform extra task for routing and forwarding the packets, so it is more prone to energy drainage. More power is needed for communicating long distant neighbors. In mobile ad hoc network nodes communicate with each other through the wireless channel. There are two path loss modes the Friis model and two-ray models. As Friis model equation is as follows

$$\frac{P_r}{P_t} = G_t G_r \left(\frac{\lambda}{4\pi R} \right)^2 \quad (2)$$

Where, P_t is transmitted power and P_r is power at the receiving antenna, G_t and G_r are the antenna gains, λ is the wavelength $\lambda = \frac{c}{f}$, f is frequency and *R* is the distance between the nodes. Energy-consumption of a node is directly proportional to the distance of that node with its neighbors. Sum of distance to all neighbors (Sd_v) is found as

$$sd_v = \frac{1}{d_v} \sum_k \sqrt{(x_v - x_k)^2 + (y_v - y_k)^2} \quad (3)$$

Where (x_v, y_v) and (x_k, y_k) are the coordinates of the node *v* and node *k* respectively. Summation is done for all neighbors *k* of node *v*. The parameter “Sum of distances” is used for energy consumption at the time of clusterhead selection [1].

Another path loss model is two-ray model or two-path model or free space model in which signal reaches to receiver through paths one is line-of-sight path and other is path which reflected(or refracted or scattered**)** wave is received. It is described as follows[ ]

$$\frac{P_r}{P_t} = G_t G_r \left(\frac{h_t h_r}{d^2} \right)^2 \quad (4)$$

Where h_t and h_r are heights of transmitter and receiver receptively.

iii. Mobility: Mobility or stability is an important factor in deciding the clusterheads. In order to avoid frequent clusterhead changes, it is desirable to elect a clusterhead that does not move very quickly. When the clusterhead moves fast, the nodes may be detached from the clusterhead and as a result, a re-affiliation occurs. Re-affiliation can increase computation and processing, which is not a desirable feature [1]. The running average of the speed for every node till current time *T* gives a measure of mobility and is denoted by M_v, as

$$M_v = \frac{1}{T} \sum_{t=1}^{T} \sqrt{(x_t - x_{t-1})^2 + (y_t - y_{t-1})^2} \quad (5)$$

Where (x_t, y_t) and (x_{t-1}, y_{t-1}) are the coordinates of the node *v* at time *t* and (*t-1*) respectively.

iv. Power: A clusterhead consumes more battery than an ordinary node because it has extra responsibilities. It can estimate the remaining battery power by the amount of time spent by the node as a clusterhead. The parameter $P_v$is the cumulative time of a node being a clusterhead. $P_v$is used to measure how much battery power has been consumed by the node [1]. Higher the value of $P_v$lower the remaining battery power

3.1.2 Computation of weight factor of node v

In this distributed clustering protocol each node is assigned a weight that indicates its suitability for its selection as clusterhead. This weight is decided by a generalized formula that takes into account all the above parameters. All four parameters (Δ_v ,Sd_v, M_v ,and P_v,) explained above can be used as a performance matrix for selection of a node as a clusterhead. Weight of these parameters can change according to requirement. Weighing factors are chosen in such a way that

$$W_1 + W_2 + W_3 + W_4 = 1 \text{ or } \sum_{i=1}^{4} W_i = 1 \quad (6)$$

Combined weight of a node $W_v$is calculated as follows

$$W_v = (W_1 \times \Delta_v) + (W_2 \times Sd_v) + (W_3 \times M_v) + (W_4 \times P_v) \quad (7)$$

Each node calculates its weight and broadcasts it periodically in a hello packet to all nodes in its transmission range. When a node receives the weights of its 1-hop neighbors, it inserts them in the possible CH set, which includes all potential cluster-heads.

3.1.3 Cluster formation

At the system initiation each node assumed to be holding a status "*undecided*". Periodic broadcast of hello message enables a node to gather useful information about its neighborhood. Based upon the information obtained from the hello messages each node computes the combined weight value. This information is further exchanged by neighbors and nodes store this neighborhood information including combined weight in neighbor tables. If a node determines that it has the lowest value of combined weight among its neighbors, it changes its status as "*CH*" and sends "*join_cluster*" messages to its neighbors. Each neighboring node receiving this join request check its own status and if it is still "*undecided*" , it respond with "*accept_join*" and become a member of that clusterhead and change its status as "*member*". This process runs in parallel and continued until all the nodes of the network either become clusterheads or members.

Algorithm 1: Cluster formation algorithm

1. Calculate the combined weight W_v of each node *v*
2. Broadcast $W_v$to all immediate neighbors through HELLO Packets
3. Process the broadcast received from immediate neighbors and records the W_x
4. If {$W_v < W_x$}
5. Node v selects itself as clusterhead Start a timer *Tclusterhead and* Send "*join_cluster"* message to all nodes *x* in the neighbor table
6. If { Node *x* is neither a *CH* nor a member of any cluster}
7. Node *x* accept "*Join_cluster"*, reply with "*accept_join"* to *v*
   else
8. does not respond to join request
   else
9. node *v* is not a *CH*, wait for *join_cluster* from other *CH*
10. Go to step 4 if there is any node with status "*undecided"*

The membership of leaving cluster and can join new cluster this is called re-affiliation. In ad hoc environment there may exist some situations where some regions of the network become overcrowded (for example location near the speaker in a conference) and the clusterheads lay in this regions may be overloaded due to movement of some nodes with in this area. A clusterhead is called overloaded if it is serving more than threshold members. The overloaded clusterheads can adversely affect the performance of the network.

Table 1. Neighbor Table

ID	Role	Cluster	Load	Flag

Table 2. Cluster Adjacency Table (CAT)

ID of adjacent CH	Gateway Node ID	Load

This algorithm further allows no two clusterheads to be one-hop neighbor of each other. Nodes falling in overlapped clusters are treated as Gateway (nodes connecting two clusterheads. The clustering algorithm is as described below. The proposed algorithm of the following seven steps:

Step 1. ON POWER-ON or on receipt of a RECLUSTER packet a node N initializes its status to known (i.e., set status =UN) and set weight equal to Zero (i.e., W_N =0).If nodes received a RECLUSTER packed, it also broadcast it.

Step 2. Periodically each node broadcasts HELLO packets containing its ID.

Step 3. On receiving the HELLO packets from nodes lying within its transmission range, each node broadcasts its neighbor table.

Step 4. If a node does not receive any HELLO packet, it declares itself a clusterhead. Otherwise the node calculates its weight and sends it to all neighbors.

Step 5. If for a node there is no neighbor node having a smaller weight than its weight then the node becomes a clusterhead. Otherwise the node sends a joint request (JOIN_REQ) message to the neighboring node with the smallest weight, considering that node to be its clusterhead.

Step 6. The clusterhead acknowledges a JOIN_REQ message by sending a joint acknowledge (JOINT_ACK) message. The clusterhead may refuse to respond if its degree is already equal to the threshold value.

In such a case the node sends JOINT_REQ message to the next clusterhead which has the weight just higher than the clusterhead which had just refused sending the JOIN_ACK message. In the process the node ultimately either finds the clusterhead or finds none. If the node finds clusterheads responding to its JOIN_REQ message to the neighboring clusterheads repeating the process.

Step 7. If the node does not receive a JOINT_ACK message after a given number of attempts, then this node generates a RECLUSTER packet and broadcasts it.

The Figure 3 presents the pseudo code for the cluster formation process carried out at a node. The procedure is followed at all the nodes in the network for the complete clustering process.

1. On POWER-ON or receipt of RECLUSTER packet
   [Set status=UN; Set weight W_N = 0]
2. If (received RECLUSTER packet)
   Broadcast RECLUSTER packet.
3. Broadcast HELLO packet containing its node ID
4. Receive HELLO packets from other nodes.
5. If (Node does not receive HELLO packet from any node)
   Node declares itself as clusterhead (i.e., set status =CH)

6. Add IDs of the neighbor nodes in one-hop neighbor table

7. Broadcast its one-hop neighbor table

8. Receive one-hop neighbor table from the neighboring nodes and construct two-hop neighbor table

9. Calculate weight according to equation (7).

10. Broadcast its weight.

11. Receive weights from neighbor nodes.

12. If (own-weight< all weights of neighbor nodes)
    Declare itself as clusterhead and set status =CH
    Else
    Declare itself as ordinary node and set status =O
13. If the node receives packets from two or more clusterheads
    Declare itself as Gateway and set status = G

14. If (node is O or G)
    Send JOIN_REQ message to the neighboring CH with smallest weight.
    Repeat
    {If (node receives a JOIN_ACK message)
    {The node becomes a member of the cluster.
    Marks the JOIN_ACK message sender node as its CH
    }
    Else
    Send JOIN_REQ message to the next Neighbor CH with smallest weight.
    Endif
    }
    Until (JOIN_REQ message has not been sent to all neighbor CHs or CH selected)

15. If (CH not selected)
    { Wait for a random time.
    If (Number of attempts allowed is over)
    Broadcast RECLUSTER packet.
    Else
    { Increment number of attempts.
    Repeat step 12.
    }
    }
16. If (node is CH)
    Receive JOIN_REQ message.
    If (Degree (CH) == Threshold value)
    Discard the message.
    Else
    Acknowledge with JOIN_ACK message.
17. Stop

Fig.3. Pseudo Code for Cluster Formation

An Illustrative Example, we demonstrate our algorithm with the help of a suitable example. The system topology in Figure 4, which consist nodes and present the initial configuration. Nodes in the network space with their individual IDS.

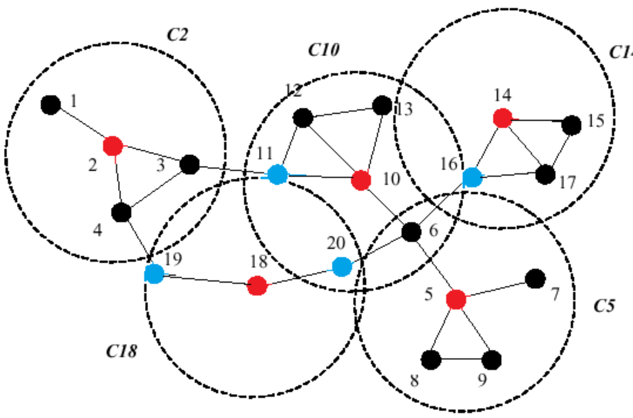


Fig.4 Cluster Formation and Cluster Maintenance

In ad hoc networks nodes are free to move within the clusters, and of course move in and out of a cluster. If member nodes move from one cluster to another than they can resign from cluster. Cluster Head and Gateway nodes moves from one cluster to another cluster. When the clusterhead moves fast, the nodes may be detached from the clusterhead and as a result, a re-affiliation occurs. Re-affiliation can increase computation and processing.

3.1.4 Clustered change is event driven and is invoked as:

1. When two cluster heads move into the range of each other each head waits a certain period of time before resigning its head role. The time period is due the mobility of node as the node may not remain in its range for longer time.

2. When a clusterhead leaves its own cluster and moves away then it will check its id with other clusterhead and here in our case a node with lowest id becomes new clusterhead. If the id is higher than the corresponding clusterhead, then it simply joins it as clustermember, otherwise it becomes new clusterhead and all other nodes will join it as cluster member.

3. Now the members of the clusterhead that has left its cluster will again start clustering procedure without affecting remaining clusters.

3.1.5 Cluster member or Gateway Change

1. If a clustermember or gateway node moves away from its containing cluster, it periodically sends join messages and if some existing member hears it, it replies by becoming its member by choosing that member as access point and sets its own status to clusterguest, and this change is reflected in clusterhead table.

2. When a member node or gateway node moves away and does not find any new cluster or other member, it repeatedly broadcasts hello message and changes its status to standalone.

4. Multipath Routing among clusterheads

EECMR uses the two types of routing Intra_cluster routing and Inter_cluster routing

4.1 Intra_cluster routing:

If the destination node *B* is in the same cluster as source node *A*, then the clusterhead of this cluster directly sends a positive acknowledgement to node *A* that, it has a direct route to node *B*.

4.2 Inter_cluster routing:

If the sender A and receiver B are not in the same cluster then node *A* sends a route request message containing node *B* as a destination to its clusterhead CH. *A*'s clusterhead propagates the message to its neighboring clusterheads

Let P_j^t be the path from source node S to destination node D at time t is defined as alternative sequence of clusterheads and gateways. It denoted by

$$P_j^t = S - CH_1 - GW_1 - CH_2 - GW_2 - CH_3 \ldots\ldots\ldots\ldots - CH_n - D$$

The weight of this is defined as sum of weights of clusterheads

$$W(P_j^t) = \sum_{i=1}^{n} W(CH_i) \qquad (8)$$

Optimal path selection:

Let k be number of node disjoint paths from source S to destination D, an optimal path is a path whose weight is least among k paths. It is defined as follows

$$OP_j^t = \{P_j^t \; / \; \min\{\, W(P_j^t)\} \text{ and } 1 \le j \le k\} \quad (9)$$

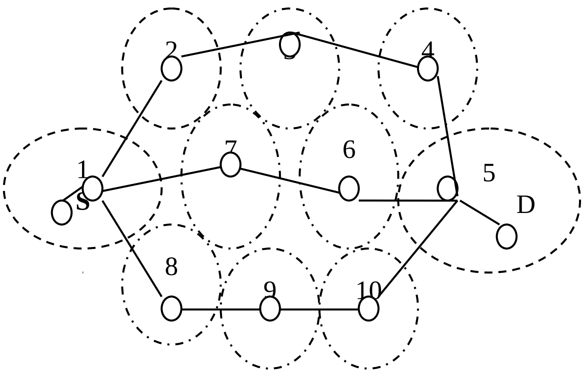


Fig .5. Node-disjoint paths through Clusterheads

The optimal path is also called primary path. Initially it is used for data transmission, if it fails, then the secondary path (whose cost is least among k-1 paths) is used for data transmission. For example in figure 5, there are three node disjoint paths from source S to destination D via clusterheads; the paths are 1-2-3-4-5, 1-7-6-5 and 1-8-9-10-5.

If source node S belongs to ith cluster whose clusterhead is CH_i then it is called as source clusterhead. Similarly, If destination D belongs to jth cluster whose clusterhead is CH_j then it is called as destination clusterhead in figure 5. clusterhead 1 is called as source clusterhead and clusterhead 5 is called as destination clusterhead. If two clusterheads are neighbors then they are called as 1-hop neighbor clusterheads. In figure 5, the clusterheads 3, 7, 8 are neighbors of the clusterhead 1.

4.3 Route Discovery and Route Selection

The route selection is based on the route discovery. In order to facilitate the computation of multiple node disjoint paths from the source to destination, We choose the Ad hoc On-Demand Distance Vector Multipath (AODVM)[9] protocol as a candidate protocol and make modifications to it to enable the discovery of node disjoint paths via clusterheads. The AODVM is the extension of AODV [6] to provide multiple nodes disjoint paths.

The Proposed Modifications

The proposed modifications are explained briefly as follows. Only clusterheads maintain routing tables and run this protocol to find node-disjoint paths. The other nodes (Gateway-nodes and Cluster-members) don't maintain the routing tables, simply they forward the packets according to specified path.

Modification of Control packets

The RREQ packet of AODVM is same as RREQ packet of EEACMR**.** The RREP packet of AODVM is extended as RREP packet of EEACMR by adding with weight filed**,** this field carries cumulative weight of clusterheads through which it passes. The initial value of this field is zero as shown in figure 6

RREP of AODVM	Weight

Fig.6 RREP Packet in EEACMR

Modification of Tables

In the AODVM, each node maintains two tables, the routing table is used to forward the data packets from source to destination whereas the RREQ table is used to form the route from source to destination, the fields of both tables are shown in figure 7 and figure 8.

Destination ID
Source ID
Lasthop ID
Nexthop ID
Cumulative Weight

Fig.7. Fields of Routing table

Destination ID	Source ID	Neighbor ID who transmitted the RREQ	Hop-count	Expiration time

Fig.8. Fields of RREQ Table

In the EEACMR, the routing table is extended to include the weight field. But RREQ table is same. These two tables are maintained by clusterheads only.

The proposed modifications in node functioning

In the AODVM, when the source node wants to send packets to a destination, it looks up its route cache to determine if it already contains a route to the destination. If it finds that an unexpired route to the destination exists, then it uses this route to send the packet. But if the node does not have such a route, then it initiates the route discovery process by broadcasting a RREQ packet.

It is modified that when the source node wants to send packets to a destination, it sends the RREQ to its clusterhead; clusterhead looks up its route cache (routing table) to determine if it already contains a route to the destination. If it finds that an unexpired route exists to the destination, then it sends the reply to the source and uses this route to forward the packets from the source node, else then it initiates the route discovery process by forwarding a RREQ packet to its all neighbor clusterheads.

In the EEACMR, the propagation of RREQ packet at intermediate clusterhead follows the same rules as at intermediate node in AODVM. When RREQ packet arrives at intermediate clusterhead, The RREQ is inspected and the following two cases it is dropped.

i. If this RREQ packet was already processed by the intermediate clusterhead or

ii. If TTL value of the RREQ packet becomes zero.

Otherwise, the intermediate clusterhead broadcasts the RREQ to all its neighbor clusterheads by recoding the information about the clusterhead through which RREQ was received and corresponding hop-count back to the source clusterhead into its RREQ table, in this way the RREQ is forwarded by an intermediate clusterheads. Finally several RREQs will reach to destination clusterhead through different paths from the source clusterhead.

When a RREQ packet is received to a destination clusterhead, it generates a RREP packet and sent back to its neighbor clusterhead (last hop) from which the RREQ packet has been received. When an intermediate clusterhead receives the RREP packet, it checks its RREQ table to find its neighbor clusterhead (next hop in reverse path) through which shortest path back to the source clusterhead and sends the RREP packet to it by deleting corresponding entry in the RREQ table.

Before forwarding RREP , clusterhead adds its weight to it. In order to ensure that a clusterhead does not participate in multiple paths, If it receives another RREP packet from different neighbor clusterhead, it is dropped (i.e., RREQ table is already empty), it generates a Route Discovery Error (RDER) packet and sends it back to the neighbor clusterhead through which another the RREP packet has been received. The neighbor clustered upon receiving the Route Discovery

Table 3: Simulation Set up.

Sno	Parameter and Values
1	Network size and Node's Placement 1000 x 1000 Sq meters (Size of NAM),
2	Number of Nodes: 25, 50, 75, 100
3	Total Simulation time : 100s
4	Node's Mobility : Random Way Point Mobility Model(RWPMM)
5	Node's Speed: 10 m/s, 20m/s, 30m/s, 40m/s
6	Node's Pause Time: 2s, 4s, 6s, 8s
7	Application Layer : Constant Bit Rate(CBR)
	Packet Payload size: 512 bytes
	Data rate :10Kbps, 20Kbps,30Kbps ,50Kbps
	Number of source and destination pairs : 1, 2, 4,8
8	Transport Layer: User Datagram Protocol(UDP) ,CBR
9	Network Layer: EAOMR and EEAOMR Routing Protocols
10	Data Link Layer:Logical Link Control Layer (LLC)
	Medium Access Layer(MAC):IEEE 802.11 DCF
	Interface Queue Type: CMU Priority Queue with size 50 packets
11	Physical Layer:
	Antenna Model : Omni Directional
	Wave Propagation Model: Two-way Ground Model
	Channel Type: Wireless Channel Physical layer Channel Bandwidth: 2Mbps Lucent Wave LAN card with frequency 915MHz Transmission range of node : 250 meters Interference range of node: 500 meters
12	Energy Model:
	Initial energy of each node: 25Joules
	Transmits Power: 0.075 Watts
	Receive Power: 0.065 Watts
	Idle Power: 0 .1 Watts
	Sleep Power: 0.005 watts
	Transition Power :0.2watts
	Transition Time: 0.005 sec

Error (RDER) packet will try to forward the RREP packet to another neighbor clusterhead through which shortest path back to the source clusterhead and sends the RREP packet to it by deleting corresponding entries in the RREQ table.

Finally several RREP packets will be received by the source clusterhead, this information is recorded into its cache based on arriving order and it sends the reply to the source .The path with minimum weight is selected as an optimal path for transmission of source's data. If clusterhead wants to transmit the data, it follows above procedure to find node-disjoint paths. Source clusterhead selects paths according to their descending order of weights.

5. Route Maintenance

The EACMR handles route maintenance in a manner similar to the AODVM. Whenever a link breakage happens in a route due to a node moving away, the previous hop node of the moved away node is responsible for sending a Route Error (RERR) message back to the source clusterhead to inform the breakage. It chooses alternative routes to maintain the connection. If there are no more redundant routes left, then it will start a new route discovery.

A node can listen to the signals transmitted the nodes within its transmission range. An edge two nodes as shown in figure 2 signifies that are neighbors a node has is clear from the Once the neighbors have been identifies, easy calculates its weight W according to equation weighing factors considered for the correspond system parameters defined, are w1= 0.4, w2 = 0.2, w4 = 0.05, and w5 = 0.05. Note the weighing parameters are chosen such that with w1+w2+w3+w4+w5 = 1. The contributions of the system parameters can be tuned by choose appropriate combination of the weighing factor.

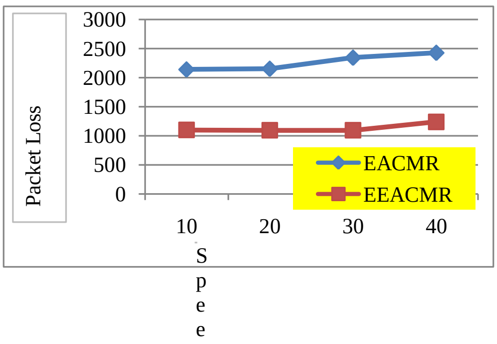
6. Simulation Study

The simulation experiments conducted for the performance evaluation were implemented in the NS2 [27][ ] . We evaluate the performance of the different clustering algorithms on a mobile network of N = 50 to 100 nodes covering a grid of 2000m x 2000m. The simulation time considered is 50 seconds. We assume that each clusterhead can handle at most 5 nodes (i.e. a = 5) in its cluster. As mobility is an important parameter in an ad hoc network, the weight w1 associated with Δ_v , is chosen to be high. The next higher weight is given to w2 associated with Sd_v , weights W4 and W5 were assigned lower weights with M_v and P_v respectively. The values of the weights used in our simulation are w1 = 0.4, w2 = 0.3, w3 = 0.2, w4 = 0.05, and w5 = 0.05. It should be noted here that, these values are arbitrary at this time and may be adjusted according to the system requirements. Simulation setup is shown in table 3

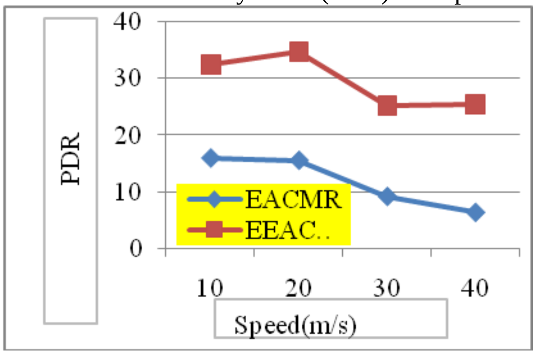
The following graphs as shown that EECAMR yields better performance as compared to ECAMR

Results and graphs

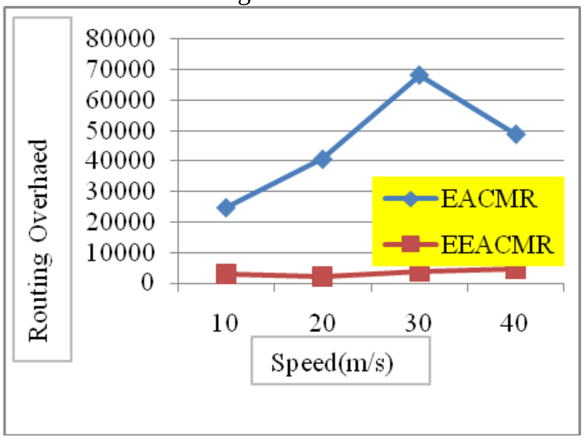
A. Number of Data Packets Loss Vs Speed



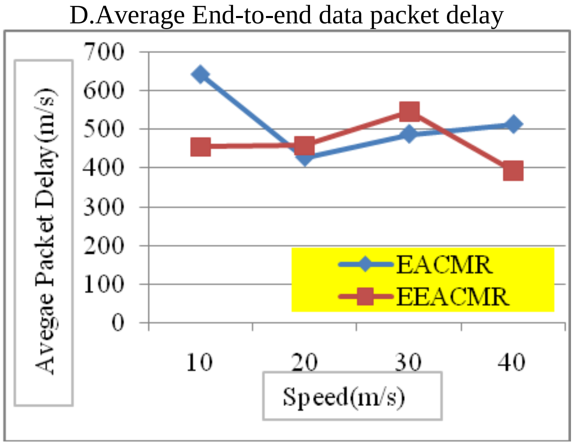
B. Packet Delivery Ratio (PDR) Vs Speed



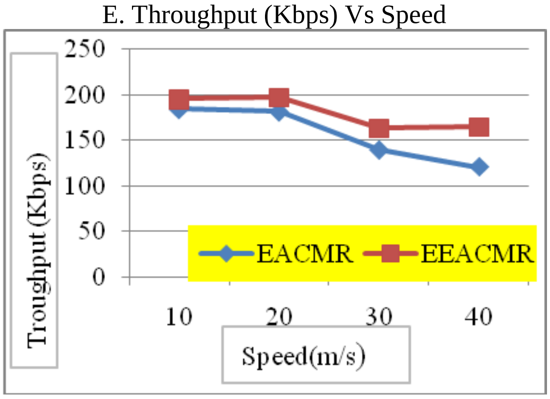
C. Routing Control Overhead



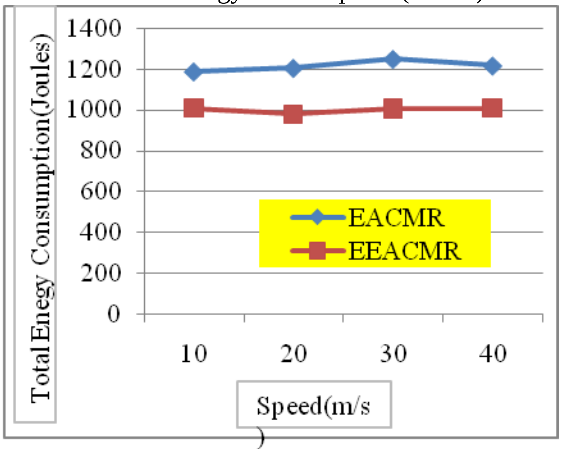
D.Average End-to-end data packet delay



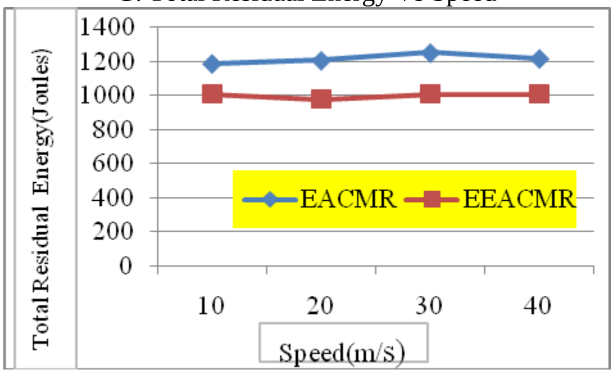
E. Throughput (Kbps) Vs Speed



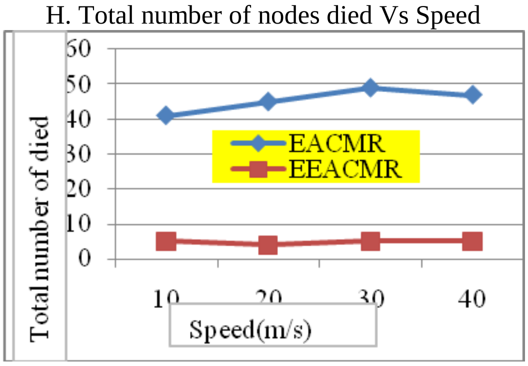
F. Total Energy Consumption (Joules)



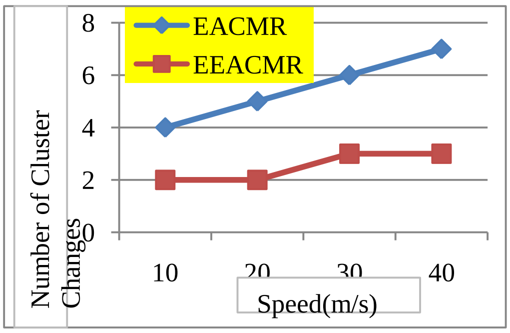
G. Total Residual Energy Vs Speed



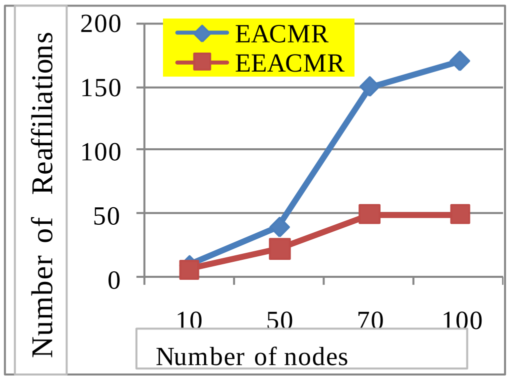
H. Total number of nodes died Vs Speed



I. Number of cluster changes Vs Speed



J. Number of Reaffiliations Vs Number of nodes



.

Figure J represents the reaffiliation count of the three clustering algorithms as a function of the number of nodes. Reaffiliation count is the number of different clusters a node joins sharing the stimulation time. The more is the number of clusters that a node joins greater would be the power consumption by the node. Here also our algorithm is found to perform well. Lowest has the largest reaffiliation count.

CONCLUSION

. In this paper EECAMR protocol is proposed to optimize number of clusters based four parameter such as (1) Degree-difference (2) Sum of distance to all neighbors, (3)Mobility of node and (4) Battery power of node.

Optimization problems have great importance in scientific, engineering design and decision-making applications. When an optimization problem has only one objective then the task of finding optimal solution is called single-objective problem. Normally in a single-objective problem we are interested in finding only a single solution except multimodal functions. When an optimization problem has more than one objective functions then the optimization problem is known as multi-objective optimization problem. In multi-objective problems multiple criteria are considered. Since there is no single solution that can be termed as optimal solution, i.e., having multiple conflicting objectives, therefore, we are interested in finding a number of optimal solutions.

Finally the performance of EEACMR is compared with performance of EACMR by using different quantitative performance metrics like Number of Data Packets Loss, packet deliver ratio, throughput, routing control overhead, end-to-end average packet delay, energy consumption, number of nodes died, number of clusters changes and number of reaffiliations. EEACMR enhances the lifetime of network by reducing number of nodes died 66% as compared to EACMR.

It is proved that EEACMR yields better performance as compared to performance of EACMR.

REFEEERENCES

[1] C. S. R. Murthy and B. S. Manoj, "Ad hoc Wireless Networks: Architectures and Protocols", *Prentice Hall*, 2004.

[2] M. Bheemalingaiah, M. M. Naidu and D. Sreenivasa Rao," Energy Aware Clustered Based Multipath Routing in Mobile Ad Hoc Networks", *International Journal of Communications, Network and System Sciences, pp:91-169, Vol 2, 2009.*

[3] Jane Y. Yu and Peter H. J. Chong, “A survey of clustering schemes for mobile ad hoc networks”, *in Proceedings of IEEE Communications Surveys,* Volume 7, 2005 ,Page(s): 32 – 48.

[4] M. Anupama and Bachala Sathyanarayana “Survey of Cluster Based Routing Protocols in Mobile Ad hoc Networks”, *International Journal of Computer Theory and Engineering, Vol. 3, No. 6, December, 2011.*

[5] J.-H. Ryu, S. Song, and D.-H. Cho, “New Clustering Schemes for Energy Conservation in Two-Tiered Mobile Ad-Hoc Networks,” in *Proc. IEEE ICC’01*, vo1. 3, June 2001, pp. 862–66.

[6] Prerna Malhotra, Ajay Dureja , “A Survey of Weight-Based Clustering Algorithms in MANET”, *IOSR Journal of Computer Engineering (IOSR-JCE), ISSN: 2278-8727Volume 9, PP 34-40, Issue 6 , 2013.*

[7] Prerna Bharti and Gaurav Pandey,” A Survey of Weight Based Clustering lgorithm”, International Journal of Advanced Research , April 2017.

[8] S.K Dhurandher, GV Singh, “ Weight based adaptive clustering in wireless ad hoc networks”, IEEE International Conference on Personal Wireless Communications, PP:95-100 2005.

[9] M. Chatterjee, S. K. Das, and D. Turgut, “An On-Demand Weighted Clustering Algorithm (WCA) for Ad hoc Networks,” in *Proc. IEEE Globecom’00*, 2000, pp. 1697–701.

[10] Ratish Agarwal, Roopam Gupta, and Mahesh Motwani “Energy Aware Load Balancing Clustering in Mobile Ad Hoc Networks”, International Journal of Computer Science and Electronics Engineering (IJCSEE) Volume 2, Issue 1 (2014) ISSN 2320-401X; EISSN 2320-4028.

[11] A. D. Amis and R. Prakash, “Load-Balancing Clusters in Wireless Ad Hoc Networks,” in *Proc. 3rd IEEE ASSET’00*, Mar. 2000, pp. 25–32.

[12] Hamid Ali, Waseem Shahzad, Farrukh Aslam Khan, “Energy-efficient clustering in mobile ad-hoc networks using multi-objective particle swarm optimization”, Elsevier Journal of Applied Soft Computing , may 2011.

[13] J. Broch, D. Johnson, and D. Maltz, “The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks”, *IETF* Internet draft, 2004.

[14] Charles E. P, E. M. Belding-Royer and I. Chakeres, “Ad hoc On-Demand Distance Vector Routing”, IETF Internet draft, 2003.

[15] Mahesh K. M and Samir R. D, “On-demand Multipath Distance Vector Routing in Ad hoc Networks”, *In Proceedings of IEEE International Conference on Network Protocols*, pp.14-23, 2001.

[16] S. J. Lee and M. Gerla, “Split Multipath Routing with Maximally Disjoint Paths in Ad hoc Networks”, *In Proceedings of IEEE ICC*, pp. 3201-3205, 2001.

[17] L. Wang, Y. Shu, Z. Zhao, L. Zhang and O. Yang, “Load Balancing of Multipath Source Routing in Ad hoc Networks”, *In Proceedings of IEEE ICCC,* 2002, Vol. 5, pp. 3197 - 3201.

[18] Zhenqiang Y, Krishnamurthy S. V and Tripathi S. K. “A Framework for Reliable Routing in Mobile Ad hoc Networks”, *in Proceedings of IEEE INFOCOM,* 2003, Vol. 1, pp. 270 -280.

[19] M.Bheemalingaia,M.M.Naidu, D.Sreenivasa Rao and P.Vishvapathi,” Energy Aware On-demand Routing Protocol in Mobile Ad Hoc Networks”, IRACST-International Journal of Computer Networks and Wireless Communications (IJCNWC), ISSN: 2250-3501 Vol .6, No 5, Sep-Oct 2016.

[20] Yuan P, Bai Y and Wang H, “A Multipath Energy-Efficient Routing Protocol for Adhoc Networks”, *In Proceedings of International Conference on Communications, Circuits and Systems*, Vol. 3, 2006, pp. 1462 -1466.

[21] Liansheng T, Ling X, King T. K, M. Lei and Zukerman, “LAMOR: Lifetime-Aware Multipath Optimized Routing Algorithm for Video Transmission over Ad hoc Networks”, *in Proceedings of IEEE Vehicular Technology Conference*, 2006, Vol. 2, pp. 623-627.

[22] Jin S. Y, K. Kang, Y. J. Cho and S. Y. Chae, “Power-Aware Multi-Path Routing Protocol for Wireless Ad hoc Network”, *in Proceedings of IEEE Wireless Communications and Networking Conference*, 2008, pp. 2247-2252.

[23] Anand P. S, Anto A. J, Janani V and Narayanasamy P, “Multipath Power Sensitive Routing Protocol for Mobile Ad hoc Networks”, *in Proceedings of Wireless on Demand Network Systems, Springer*, Vol. 2928, 2004, pp. 84-89.

[24] Do-Youn H, Eui-Hyeok K, and Jae-Sung L, “An Energy Aware Source Routing with Disjoint Multipath Selection for Energy-Efficient Multihop Wireless Ad hoc Networks”, *in Proceedings of International Federation for Information Processing*, 2006, pp. 41-50.

[25] Hui-Yao An, Ling Zhong, Xi-Cheng Lu, and Wei Peng, “A Cluster-Based Multipath Dynamic Source Routing in MANET”, *in Proceedings of IEEE International Conference on Wireless and Mobile Computing, Networking and Communications*, Aug. 2005, Vol. 3, page(s): 369- 376.

[26] Zhengyu W, Xiangjun D and Lin C, “A Grid-Based Energy Aware Node-Disjoint Multipath Routing Algorithm for MANETs”, *in Proceedings of International Conference on Natural Computation*, 2007, Vol. 5, pp. 244-248.

[27] Network Simulation 2 official website https://ns2tutor.weebly.com/

[28] M Bheemalingaiah, MM Naidu, D Sreenivasa Rao, P Vishvapathi “Performance analysis of power-aware node-disjoint multipath source routing in mobile ad hoc networks” . IEEE 7th International Advance Computing Conference (IACC), pp: 361-371, 2017.