AUSTRALIAN JOURNAL OF BASIC AND APPLIED SCIENCES

ISSN:1991-8178 EISSN: 2309-8414
Journal home page: www.ajbasweb.com

Improved Local Route Repair And Congestion Control In Self Organizing Networks

[1]V. Vinoth Kumar and [2]Dr.S. Ramamoorthy

[1]*Department of Information Technology, Research Scholar, Dr.M.G.R.Educational and Research Institute University, Chennai, India.*
[2]*Department of CSE & IT, Professor, Dr.M.G.R.Educational and Research Institute University, Chennai, India.*

Address For Correspondence:
V. Vinoth Kumar, Department of Information Technology, Research Scholar, Dr.M.G.R.Educational and Research Institute University, Chennai, India.
E-mail: vinoagra@gmail.com

ARTICLE INFO

Article history:
Received 3 April 2016
Accepted 21 May 2016
Published 2 June 2016

Keywords:
Ad-hoc networks, Congestion, Link breakage, Route repair, Queue Length, IOAS-AODV, and NS-2.

ABSTRACT

In the future, Mobile Ad hoc Network (MANET) is predicted to play more important roles for device communication with self organized infrastructure. To deal with dynamic changes in network conditions, mobile node randomly moves within the coverage area and resulting in topology change, leads to frequent link breakage. To repair the broken link AODV and DSR uses re-initialization of route discovery through broadcasting mechanism. Broadcasting floods control packets like RREQ and RREP Flooding of control packets results in congestion owing to limited buffer space, bandwidth and battery power of mobile nodes. Due to Congestion more number of packets is dropped or lost and resulting in decreasing the performance of the network. Congestion can be controlled if the numbers of control packets are reduced. In order to improve the performance a new routing algorithm is proposed which control the congestion and repair the link break by choosing a set of nodes for alternate route based on the parameters such as quadrant position, battery status, queue length, and forwarding region. The proposed method system modifies the existing AODV algorithm by using congestion control phenomena. In this paper discuss congestion control by using Improved Optimum Angle Selection AODV (IOAS-AODV) routing algorithm. The Experimental result showed that the most suitable routing protocol with congestion control can improve network performance and it is implemented, tested and the performance is analyzed by using NS-2 simulator. Based on the results it is observed that the proposed method improves the routing performance such as end to end delay, throughput, routing overhead, and packet delivery ratio.

INTRODUCTION

In MANET mobile nodes communicate over wireless channels without any base station. In this network, due to mobility, nodes arbitrarily leave or join the network. Due to dynamic changing topology of MANET, maintaining the current route for improving network performance is difficult. Currently, MANET uses many routing algorithms. All these algorithms are classified into two broad categories: proactive and reactive routing algorithms. Proactive routing algorithms also called as Table-driven algorithms (e.g. – DSDV (Senthil Kumaran, T., V. Sankaranarayanan, 2011), OLSR (Eiman Alotaibi, Biswanath Mukherjee, 2012). These algorithms update and maintain the latest information of routes in the form of tables. This table (routing table) is updated based on the periodic exchange of HELLO packets among the nodes in the network. These control packets will causes increased network overhead, network congestion, battery power consumption, and lowering data throughput. Reactive routing algorithms (e.g., AODV (Pardeep Kumar Mittal and Meenakshi Devi; Perkins, C.E., *et al*.,

Open Access Journal
Published BY AENSI Publication
© 2016 AENSI Publisher All rights reserved
This work is licensed under the Creative Commons Attribution International License (CC BY).
http://creativecommons.org/licenses/by/4.0/

Open Access

To Cite This Article: V. Vinoth Kumar and Dr.S. Ramamoorthy., Improved Local Route Repair And Congestion Control In Self Organizing Networks. ***Aust. J. Basic & Appl. Sci.,*** *10(10): 170-177, 2016*

2013; Grundy, A., M. Radenkovic, 2010), DSR (Pardeep Kumar Mittal and Meenakshi Devi; Perkins, C.E., *et al*., 2013; Gafur, M.A., *et al*., 2012) are also known as On-demand algorithms. They only find routes when communication is needed, rather than keeping consistent view of routing table, as done in proactive algorithms.

The routing processes of reactive algorithms generally have two phases such as route discovery and route maintenance. The first phase such as route discovery is initiated by a node when it wishes to send the data to a destination, but there is no route found in its routing table for the destination. In this case, it broadcast a control packet i.e. Route Request (RREQ) packet to all its neighbors, which in turn rebroadcast to each intermediate node until it reaches either an intermediate node that has a fresh route to the destination or reaches the destination node itself (Liu, C., J. Kaiser, 2005). The intermediate or destination node respond to RREQ by sending a Route Reply (RREP) packet that contains a path to the destination using unicast mechanism to the source node. The route maintenance phase is initiated by a node when it feels that an active route which was transporting data packet is either broken or congested. A node declares that the active route is broken or congested, when its periodic HELLO timer expires and there is no HELLO packet received.

In this case, the intermediate node intimates this news to source node by sending the Route Error (RERR) packet or perform local route repair strategy. The AODV and other reactive routing algorithm repair the route by broadcasting the RREQ packet to find an alternate route to destination as shown in Fig. 1. In this figure the link is broken between node 2 and 3 and then node 2 start flooding mechanism to establish the new route. This broadcasting leads to two problems: (i) Broadcast storm problem (BSP) due to large number of control packets (e.g. RREQ, RREP) present in network and (ii) The set of nodes may get selected during route discoveries which have less resources such as battery and buffer to store the packets, which leads to further breakage of link. The above two problems causes network congestion and degradation in performance.

The paper is organized as follows: Section II introduces the existing approaches to reduce the overhead and delay during route repair and congestion. The proposed IOAS-AODV approach to resolve the above mentioned problems is explained in section III, section IV discuss the simulation of the proposed IOAS-AODV algorithm and compares the result obtained with AODV and OAS-AODV (Pravin Ranjan and R. Leela Velusamy, 2014) routing algorithms. Finally, Section V concludes the paper concepts.

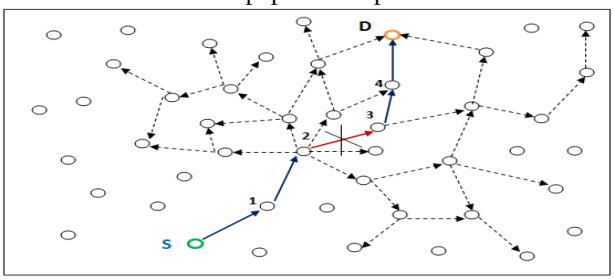


Fig. 1: Route discovery by flooding after link breakage between node 2-3.

2. Proposed Methodology:

A. Background:

In MANETs every nodes act as a router and share the information to its neighbor nodes that are within their transmission range. The information is stored based on the route maintenance message such as HELLO packet periodically sent by the intermediate nodes. The HELLO packets contain the information such as location, battery status and queue length etc. The format of frame used for HELLO packets is shown in Fig. 2(a). In the proposed algorithm, when congestion occurs the congested node in the path sends a C-HELLO packet as shown in Fig. 2(b) to its predecessor node and it contains the location of a non-congested node.

Node Id	X-coordinate	Y-coordinate	Battery_status	Queue length

Fig. 2(a): HELLO packet frame format.

Node Id	Nid-X coordinate	Nid-Y coordinate

Fig. 2(b): C-HELLO packet frame format.

Where, Nid-X coordinate is X coordinate of non congested node. Nid-Y coordinate is Y coordinate of non-congested node.

The following assumptions are made to implement the IOAS- AODV algorithm

i. All nodes are of similar type with same transmission range (T), battery power and buffer capacity.

ii. The mobility of nodes is random with various speeds.

iii. The transmission range of a node is divided into four quadrants Q1, Q2, Q3 and Q4 as shown in Fig 3(a). The ranges of these quadrants are shown below.

Q1= [Tx, Ty]; Q2= [-Tx, Ty]; Q3= [-Tx, -Ty]; Q4= [Tx,-Ty]

B. IOAS-AODV Algorithm:

A node along the current route declares that there is link breakage or congestion when its HELLO timer expires before reaching destination node or there is no entry found in its routing table. In this situation the IOAS-AODV algorithm is used to select the next suitable node. For selecting the next node, this algorithm selects a set of nodes based on four components: Battery_status, Queue_length, Quadrant position and forwarding range. The First component, Battery_status represents the available battery power of monitored node.

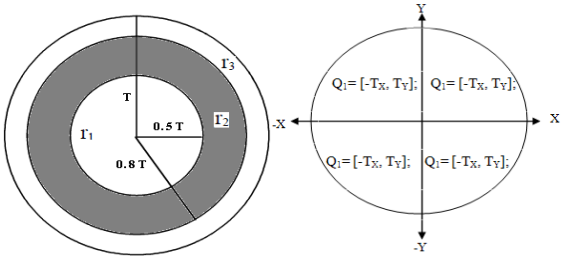


Fig. 3(a): Quadrants region **Fig. 3(b):** Forwarding region

There are three levels of battery status: Low, Normal and High. Low indicates that the node has a capacity less than 25% of total battery power and hence will not participate in new route discoveries. Normal indicates that the node has sufficient battery with a capacity greater than 40%. High indicates that node has battery power more than 60%. The second component, queue length is defined as the buffer capacity of a node. In this algorithm, two values are taken thershold1 (i.e 80% of total queue length) and thershold2 (40% of total queue length) in order to find, if a node is congested or not. If the queue length of a node is greater than thershold1 then is consider as congested and it will not participate in new route discovery process. The third component quadrant position, describes the position of next node. The IOAS algorithm selects a set of nodes from the same quadrant where next node (unreachable node /failed) belongs. In case, alternate node not found in the same quadrant, it searches nodes in other quadrants position. The fourth component is forwarding region which is best region of transmission range from where the chances of nodes to move out in near future are low. So, in this algorithm transmission region (T) is divided into three sub regions r1, r2 (also called as forwarding region) and r3 as shown in Fig. 3(b). These sub regions r1, r2 and r3 are defined as the area between 0 to 0.5 of T, 0.5 to 0.8 of T, and 0.8 to T respectively. The sub region r2 is searched first for selecting the forwarding nodes because the chances of nodes to move out from this region in near future are low.

A new route can be found by an algorithm OAS-AODV (Pravin Ranjan and R. Leela Velusamy, 2014). The purpose of OAS-AODV was to find a route with minimum number of broadcasting control packets. In this paper, an improved version of OAS-AODV is proposed in which route repair and congestion control is also taken under consideration.

Algorithm 1: The IOAS algorithm

---

1. ***if***(next_node_Battery_status < *Low* OR next_node_unreach
begin
2. Call function *Find_Alternate_Nodes();*
return Forwarding nodes
end
3. ***if*** (queue_length > *Thersold1*)
begin
4. send C-HELLO packet to predecessor ;
5. call function *Find_Aletrante_Nodes();*
End

Algorithm2: The function *Find_Alternate_Nodes ( )*

```
Find_Alternate_Nodes ( )
{
Initialize:
1.  Q1= [TX,TY];
2.  Q2= [-TX, TY];
3.  Q3= [-TX, -TY];
4.  Q4= [TX, -TY];
5.  nd_x      next_to_next_node_loc_x;
6.  nd_y      next_to_next_node_loc_y;

 //Find the quadrant from which next node belongs
8.  if (nd_x > 0)
{
9.  if (nd_y > 0)
10.   return Q1 ;
11. else return Q3 ;
}
12 else {
13   if (y>0)
14    return Q2 ;
15    else return Q4  ;
}

16. Select a random node from forwarding region and store in  array [].
17. Initialize: Node_id[2], k=1;
18. for i =1 to n do
19. if (array[i].battery_status > Recharged && array[i].queue_length < Thersold2) {
 20. if (k<=2) {
21.  Node_id[k++] = array[i]; }

22.  else  return Node_id [ ];

23. if (i to n)
   24.          repeat 16 to 25 for region r1 and r3 respectively;
 }
```

In IOAS-AODV, once a node found that link to next hop is broken, it tries to find out the reason as shown in Algorithm1. First it checks the last battery_status of that node. If battery_status is less than Low value then it waits Δt time for recharging. After Δt time if that node is still not found in its transmission range then that node is stated as unreachable and call Find_Alternate_Nodes( ) function in order to find new route as shown in Algorithm2. The flow diagram of IOAS- AODV algorithm is shown in Fig. 4. This algorithm first, finds the quadrant position of next to next node (NN_Node). Once the quadrant position is known, it selects a set of nodes from forwarding region (r2) of that quadrant. If nodes are not present in forwarding region then it searches the nodes in other regions (r1 and r3 respectively). If there is no node found in selected quadrant then it search in other quadrant and select nodes. The selected nodes are again checked with respect to battery status and queue length. If a node is found that has queue length smaller than Thershold2 or battery power greater than Recharged, it is stored in Node_id [] array. If two nodes are selected then stop the process and declare these nodes as forwarding nodes and now it's their responsibility to select further forwarding nodes, this process is continued till NN_Node or destination node is found.

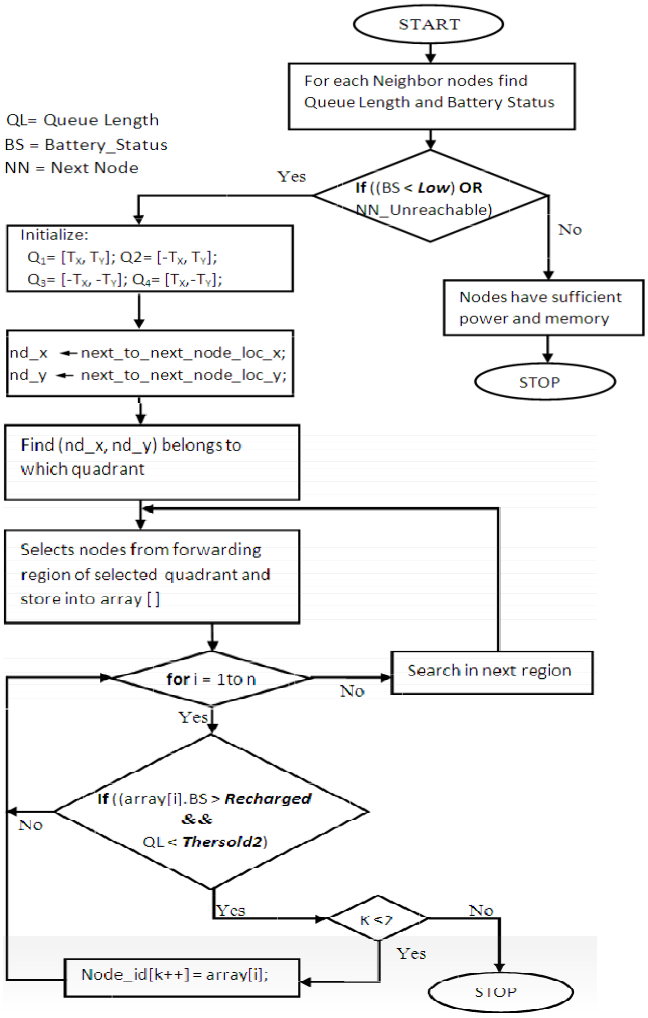


Fig. 4: Flow diagram of IOAS Algorithm ( )

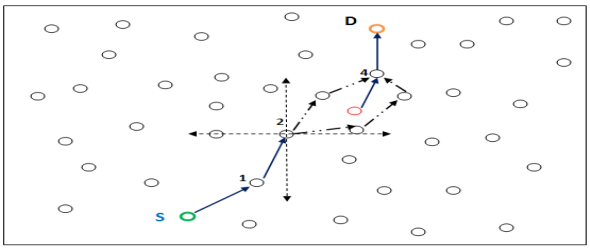


Fig. 5: Route Repair by IOAS Algorithm.

The destination node replies back to intermediate/source node. Later that node selects the path with less number of hops as shown in Fig.5. In Fig. 5 node 2 start route repair mechanism by selecting two nodes from that quadrant from which node 4 belongs and send the RREQ message to only these two nodes to find the new route.

Congestion in a network signifies that a node at any interval became congested and started to lose packets. This happens generally because of the load on the node become greater than node. To handle this situation, a new congestion prevention the capacity of mechanism is proposed. In this algorithm when a node finds that its queue length is greater than *Thershold1*, it intimates this information by sending C-HELLO packets to its predecessor with non-congested successor node location. When a node gets this type of information from its successor it call function *Find_Alternate_Nodes ( )* in order to find new sub path up to its non-congested successor node. In network there may be chance that two or more node continuously get congested. In this case also IOAS algorithm finds sub path up to first non-congested node in the upstream of primary path. Let us consider the example shown in Fig. 6(a), where node 2, 3 and 4 are congested. In this case IOAS algorithm find sub path 1- 6- 7- 8-5, as shown in Fig. 6(b). The reason is, when node 4 is

congested, it intimate this news to its predecessor node 3 with its successor node (5) location. If node 3 also gets congested then it forwards this information to node 2 with node location of 5 only, because node 3 knows that node 4 is also congested. This process continues till non-congested predecessor found.

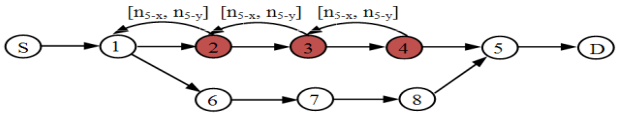


Fig. 6(a): congested nodes sending C-HELLO packets to its predecessor.

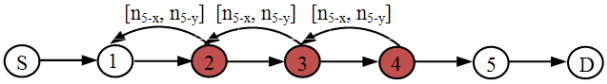
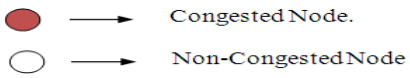


Fig. 6(b): Finding sub path.

Where:



3. Simulation And Result Analysis:

A. Simulation Setup:

In this section the proposed IOAS-AODV is implemented and compared with traditional AODV and modified version of OAS-AODV (Pravin Ranjan and R. Leela Velusamy, 2014) routing algorithm. NS-2 simulator (version 2.35) is used to evaluate the performance of these routing algorithms in different conditions. At the start of the simulation, each node stays at its initial position for pause time seconds, and starts moving with a randomly chosen speed between 0 and 10 meters after choosing the destination randomly. Once the node reaches its destination, it stays there for pause time seconds and then chooses a new destination and new speed and move again. A node repeats this process until simulation terminates. The other simulation parameters and values are shown in Table 1.

Table 1: Simulation Parameters

Parameters	Values
Simulator	NS 2.35
Routing Algorithm	AODV
Simulation Area	800 x 700
Number of Nodes	25,50…..200
Simulation Duration	150 Seconds
Antenna Type	Omni Directional Antenna
Mac Layer	IEEE 802.11 b
Packet Size	512 bytes
Packet sending Rate	5,10, 15, 20 packets/Sec
Bandwidth	2Mbps
Traffic Type	CBR
Speed	0-20 m/s
Paused Time	2 Sec.
Height of Antenna	1.5 m

B. Performance And Result Analysis:

In this section performance and efficiency of IOAS-AODV is evaluated and compared with AODV and OAS-AODV. The proposed routing algorithm is analyzed using the following performance parameters:

1. *Packet delivery ratio:* The ratio between total numbers of packets received at destination to the total number of data packets sent from the source.
2. *Average end-to-end delay:* the average time taken by data packets to reach destination which includes buffer delay during a route discovery, queuing delay at the interface, retransmission delay at the MAC layer, and propagation delay.
3. *Routing overhead:* The routing overhead is considered as the total number of control packets (RREQ/RREP/RERR) generated during Hello mechanism.
4. *Route breakages:* The route breakages are the average route breakages that a flow experiences during the simulation time.
5. *Throughput:* It is defined as the successful message delivery at destination node.

Fig. 7(a) represents the graph plotted for packet delivery ratio as a function of node mobility speed. The

graph shows that packet delivery ratio of OAS-AODV and AODV declines remarkably. This is because, the AODV and OAS-AODV select random nodes during route discoveries and route maintenance which leads to unsuitable selection of nodes.

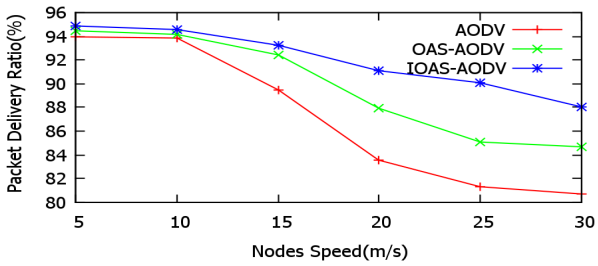


Fig. 7(a): Packet Delivery Ratio

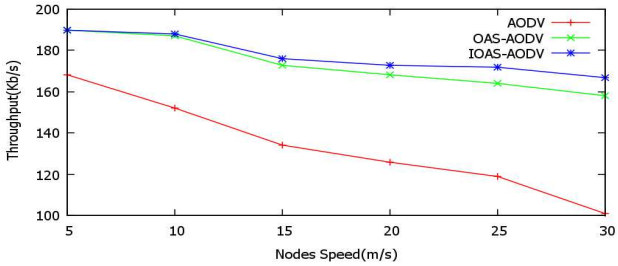


Fig. 7(b): End to End Delay

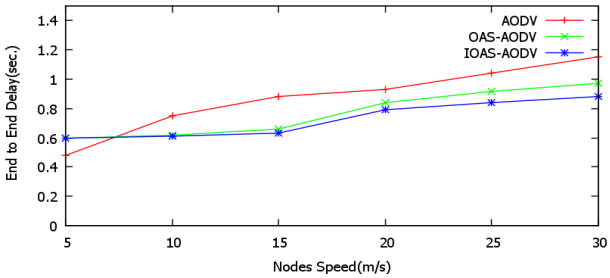


Fig. 7(c): Throughput

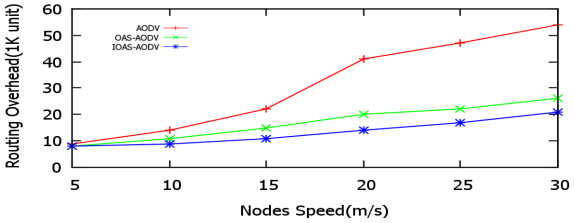


Fig. 7(d): Routing Overhead

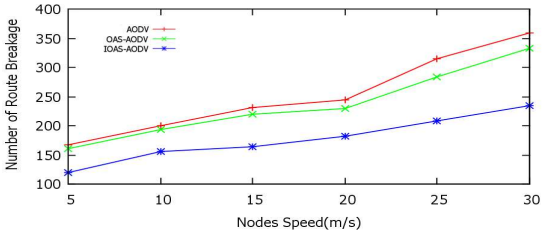


Fig. 7(e): Route Breakage

Further, some of these nodes fail to forward the packets due to insufficient battery power and storage capacity. The IOAS- AODV achieves highest packet delivery ratio in almost all the simulation conditions tested because it select more suitable node to stable routes that are not affected by mobility of nodes and buffer constraints. Therefore most data packets are delivered to receiver without being dropped which can enhance the probability of successful delivery.

Fig. 7 (b) shows End to End delay for these routing algorithms. From the plotted graph, it can be seen

that the end to end latency in these routing algorithms rise with increase of maximum speed. The AODV routing algorithm has higher end to end latency than remaining two. The reason is that, average end-to-end delay is caused by queuing delays and retransmission delays. In reactive routing algorithm, with the increase in speed, the routing route breaks

Fig.7(c) shows the achieved throughput in dense network scenario by changing the speeds of nodes. After the simulation, it is concluded that the throughput of the IOAS - AODV algorithm is better than the OAS-AODV and AODV algorithms due to significant reduction in control packet overheads. The packet overhead is reduced during the route repair by selecting the limited set of nodes rather than broadcasting the control packets to all neighbor nodes. In this paper congestion is also avoided by choosing alternate path of non-congested nodes.

Fig.7 (d) illustrates that network overhead in these routing algorithms rise with increased mobility speeds of nodes. The routing overheads in IOAS-ADOV and OAS-AODV ascend slowly, whereas it ascends sharply in AODV. The reason is that: along with the increase of speed, routing route break down easily, and it needs to be repaired. To repair, algorithms need to send more number of route request and route maintenance packets. AODV uses flooding during route repair which generates more control packets than the two algorithms.

In the Fig. 7(e) the number of breakage of route is presented with respect to speed of nodes. From plotted graph, it can be seen that the number of breakages of route in AODV and OAS-AODV are more than IOAS-AODV when maximum speed increases. The reason is that: when a route breaks IOAS-AODV finds proper forwarding nodes with enough resources like battery and queue length and also based on forwarding region. If these resources are not considered then there may be a chance that the new selected forwarding nodes don't have enough resources to transmit data. This random selection of nodes leads to further breakage of link. This causes degradation in performance.

Conclusion:

Many routing protocols proposed for self organizing networks use the route until any link breakage occurs. A frequent disconnection can cause some packet drops and transmission delays. In these networks all nodes have random mobility behavior which leads to changes in topology and frequent breakage of link. To repair the link, the algorithms such as AODV, DSR etc. This flooding leads to broadcast storm problem and congestion in the network. To avoid above mentioned problems, a new route repair and congestion avoidance mechanism is proposed, termed as IOAS-AODV. This algorithm selects a limited set of nodes in order to find a new alternate route based on battery status, queue length, quadrant position, and forwarding region. The IOAS-AODV algorithm is tested using NS-2 simulator. The simulation result reveal that the performance of IOAS-AODV is better than the OAS- AODV and AODV in terms of packet delivery ratio, end to end delay, throughput, overhead and number of route breakage.So,in future other metrics such as energy, behavior of nodes will be considered for better performance.

REFERENCES

Eiman Alotaibi, Biswanath Mukherjee, 2012 *"A survey on routing algorithms for wireless Ad-Hoc and mesh networks"*, Computer Networks, 56: 940–965.

Gafur, M.A., N. Upadhayaya, S.A. Sattar, 2012. *" An Efficient Approach For Local Repairing In Mobile Ad hoc Network*", Canadian Journal on Network and Information Security, 3: 1.

Grundy, A., M. Radenkovic, 2010. *Promoting congestion control in opportunistic networks*, in: IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob'2010), Niagara Falls, Canada.

Liu, C., J. Kaiser, 2005. *"A Survey of Mobile Ad-Hoc network Protocols"*, Tech.Report, University of Ulm, no. 2003-08.

Milena Radenkovic, Andrew Grundy, 2012.*" Efficient and adaptivecongestion control for heterogeneous delay-tolerant networks"*,ELSEVIER, journal on Ad hoc network.

Nair, G. and N.J.R. Muniraj, 2012. "*Prediction based Link Stability Scheme for Mobile Ad Hoc Networks*", IJCSI International Journal of Computer Science Issues, 9(6): 3.

Ns-2, "The Network Simulator version 2", http://www.isi.edu/nsnam/ns

Pardeep Kumar Mittal and Meenakshi Devi, *"Local Route Repair in MANET for on Demand Routing Protocol: A Review*, IJarcsse, 4: 5.

Perkins, C.E., E.M. Royer and S. Das, 2013 "Adhoc On- Demand Distance vector Routing (AODV)", IETF RFC 3561

Pravin Ranjan and R. Leela Velusamy, 2014 *"Route Discovery in Mobile Ad Hoc Network using an Optimum Angle Selection based approach"*,International Conference on Computer and Communication Technology,IEEE 978-1-4799-6757.

Senthil Kumaran, T., V. Sankaranarayanan, 2011 *"Early congestion detection and adaptive routing in MANET,"* Elsevier, Cairo University, Egyptian Informatics Journal.