

October 2012

Performance Enhancement of Reactive on Demand Routing Protocol in Wireless Ad Hoc Network

v. p. patil

Smt Indira Gandhi college of Engineering Navi Mumbai, INDIA, bkvpp@rediffmail.com

Follow this and additional works at: <https://www.interscience.in/ijssan>



Part of the [Digital Communications and Networking Commons](#), and the [Electrical and Computer Engineering Commons](#)

Recommended Citation

patil, v. p. (2012) "Performance Enhancement of Reactive on Demand Routing Protocol in Wireless Ad Hoc Network," *International Journal of Smart Sensor and Adhoc Network*. Vol. 2 : Iss. 1 , Article 14.

DOI: 10.47893/IJSSAN.2012.1140

Available at: <https://www.interscience.in/ijssan/vol2/iss1/14>

This Article is brought to you for free and open access by the Interscience Journals at Interscience Research Network. It has been accepted for inclusion in International Journal of Smart Sensor and Adhoc Network by an authorized editor of Interscience Research Network. For more information, please contact sritampatnaik@gmail.com.

Performance Enhancement of Reactive on Demand Routing Protocol in Wireless Ad Hoc Network

V. P. Patil, K.T.Patil, A. R. Kharade & D. D.Gote

Smt. Indira Gandhi College of Engineering, New Mumbai, India
E-mail : bkvpp@rediffmail.com,

Abstract - A Mobile Adhoc Network (MANET) is a class of wireless mobile nodes that dynamically organize themselves in arbitrary network topologies. Due to mobility of nodes, routing plays an important role in communication and routing protocols are divided into two basic classes: proactive and reactive. AODV is an on demand routing network protocol which is specially design for ad hoc networks. This protocol starts to, only when a certain node claims to send data and is an establish the route efficient adhoc network routing protocol for data transfer. AODV suffers from the limitation of low bandwidth. In the AODV routing protocol, wherever a link breaks an error message is sent to the source indicating the link failure and further communication is stopped temporarily. This paper proposes an improved AODV routing protocol by including route repair scheme to overcome the link failure problems of its predecessor. In the proposed scheme the data packet is sent through alternate routes instead of dropping it. It is shown through simulation results that the proposed scheme improves data delivery rate.

Keywords - MANET; Ad-hoc wireless networks; Routing Algorithm; AODV; pause time.

I. INTRODUCTION

A Mobile Adhoc Network (MANET)[1,2] is a class of wireless mobile nodes. Due to mobility of nodes, routing plays an important role in communication and routing protocols are divided into two basic classes: proactive and reactive. The proactive routing protocols are table-driven which usually use link state routing algorithms. The reactive protocols create and maintain routes only when needed. They usually use distance vector routing algorithm that keeps only information about next neighbors'. The Ad hoc On Demand Distance Vector (AODV) routing algorithm is a routing protocol designed for ad hoc mobile networks. AODV is capable of both unicast and multicast routing. It is an on demand algorithm, meaning that it builds routes between nodes only as desired by source nodes. It maintains these routes as long as they are needed by the sources. Additionally, AODV forms trees which connect multicast group members. The trees are composed of the group members and the nodes needed to connect the members. AODV uses sequence numbers to ensure the freshness of routes. It is loop-free, self-starting, and scales to large numbers of mobile nodes.

AODV builds routes using a route request / route reply query cycle. When a source node desires a route to

a destination for which it does not already have a route, it broadcasts a route request (RREQ) packet across the network. Nodes receiving this packet update their information for the source node and set up backwards pointers to the source node in the route tables. In addition to the source node's IP address, current sequence number, and broadcast ID, the RREQ also contains the most recent sequence number for the destination of which the source node is aware. A node receiving the RREQ may send a route reply (RREP) if it is either the destination or if it has a route to the destination with corresponding sequence number greater than or equal to that contained in the RREQ. If this is the case, it unicasts a RREP back to the source. Otherwise, it rebroadcasts the RREQ. Nodes keep track of the RREQ's source IP address and broadcast ID. If they receive a RREQ which they have already processed, they discard the RREQ and do not forward it.

As the RREP propagates back to the source, nodes set up forward pointers to the destination. Once the source node receives the RREP, it may begin to forward data packets to the destination. If the source later receives a RREP containing a greater sequence number or contains the same sequence number with a smaller

hop count, it may update its routing information for that destination and begin using the better route.

As long as the route remains active, it will continue to be maintained. A route is considered active as long as there are data packets periodically travelling from the source to the destination along that path. Once the source stops sending data packets, the links will time out and eventually be deleted from the intermediate node routing tables. If a link break occurs while the route is active, the node upstream of the break propagates a route error (RERR) message to the source node to inform it of the now unreachable destination(s). After receiving the RERR, if the source node still desires the route, it can reinitiate route discovery.

Multicast routes are set up in a similar manner. A node wishing to join a multicast group broadcasts a RREQ with the destination IP address set to that of the multicast group and with the 'J'(join) flag set to indicate that it would like to join the group. Any node receiving this RREQ that is a member of the multicast tree that has a fresh enough sequence number for the multicast group may send a RREP. As the RREPs propagate back to the source, the nodes forwarding the message set up pointers in their multicast route tables. As the source node receives the RREPs, it keeps track of the route with the freshest sequence number, and beyond that the smallest hop count to the next multicast group member. After the specified discovery period, the source nodes will unicast a Multicast Activation (MACT) message to its selected next hop. This message serves the purpose of activating the route. A node that does not receive this message that had set up a multicast route pointer will timeout and delete the pointer. If the node receiving the MACT was not already a part of the multicast tree, it will also have been keeping track of the best route from the RREPs it received. Hence it must also unicast a MACT to its next hop, and so on until a node that was previously a member of the multicast tree is reached.

AODV maintains routes for as long as the route is active. This includes maintaining a multicast tree for the life of the multicast group. Because the network nodes are mobile, it is likely that many link breakages along a route will occur during the lifetime of that route. The papers listed in [3, 4, 5] describe how link breakages are handled.

So, the fundamental requirements of on demand routing protocol [6, 7] is to discover routes to a node by flooding of request message. Once the route is established, it must be maintained as long as it is needed. When a link breaks between two nodes due to the node mobility, fading environment etc, the upstream node of the broken link will try to repair the route locally by local route repair process as described by Perkins et al. [6], if it is nearer to the destination than

source. Otherwise the node will send a Route Error (RERR) packet to the source to re initiate the route discovery process and drop data packets because of non availability of alternate path. Dropping of data packets at intermediate nodes can be costly as frequent route discovery will increase the contention and overheads.

The aim of this paper is to improve the route maintenance of AODV protocol by using an alternate route at each node, so that if link breaks between two nodes then upstream node can use the alternate path besides the dropping of data packets.

The remaining part of the paper is organized as follows: Section II discusses work related to AODV. The proposed scheme is discussed in Section III. Section IV presents the experimental results. Finally in conclusions are presented in Section V.

II. PREVIOUS RELATED WORK

Many schemes which are based on maintaining alternate route in AODV have been presented in literature[5,9,13,15]. Based on AODV routing protocol Sung-Ju Lee and Mario Gerla [7] have proposed a new scheme called AODV-BR (Backup Route) which improves the performance of the AODV routing protocols by constructing a mesh structure and providing multiple alternate routes. The scheme uses the flow of RREP packets during the route discovery phase to form the mesh like structure. Due to the wireless transmission of packets, each neighboring node overhears the RREP packets and records the source of one of RREP packets as the next hop to the destination into its alternate route table. Whenever a node overhears a RREP packet which is not transmitted to it, it will store the neighbor as next hop for the destination in its alternate table. In the situation of more than one overhears of one RREP the best route in its alternate table is stored.

In case of link failure, the upstream node will broadcast the data packet to its immediate neighbors by asking for alternate route. The neighbor who has an alternate route to corresponding destination will accept the packet and forwards it to its next hop node. When a node of the primary route receives the data packet from alternate routes, it operates normally and forwards the packet to its next hop when the packet is not a duplicate. The node that detected the link break also sends a Route Error (RERR) packet to the source to initiate a route rediscovery. In this way, alternate routes are created and used to improve the performance of AODV. One more modification was proposed in the form of AODV-ABR and AODV-ABL by Wei Kuang et al. [8]. This work was basically based on AODV-BR. In the proposed AODV-ABR (Adaptive Backup Routing), mesh structure and alternate routes are created by overhearing

control as well as data packets. By this approach, the adaptation of routing protocol to topology changes without transmitting many extra control messages is obtained. As in AODV protocol, the RREQ and RREP messages contain the information of hop count. In the proposed mechanism the information of hop count in the routing table and alternate route table is also kept and the hop count information is added into the header of data packet. When a node is looking for the routing table, the hop count information has to be updated before forwarding the data packet.

When a node detects a link break in AODV-ABR, it will perform a handshake process with its immediate neighbors to repair the broken route instead of a one-hop data broadcast to its immediate neighbors. The handshake process is accomplished by two one-hop control signals: BRRQ (Backup Route Request) and BRRP (Backup Route Reply). BRRQ is a broadcast message which contains the destination IP of the transmission path, the originator IP and an ID for this message. BRRP is a unicast message which contains the BRRQ's ID, the IP of the node which responds to the BRRQ, the IP of the node which initiates the broken-link-repair process and a hop count field that indicates the distance to the destination. Therefore, according to the hop count, the upstream node of the broken link can select a shorter alternate route. After selecting the alternate route, it will unicast data packet to the selected node which in turn send the data to its next hop. The second approach proposed is AODV-ABL which is a combination of AODV-ABR and local repair process [6] [10]. AODV-LR (local repair) will repair the link locally if the broken link is not far away from the destination, but AODV-ABR could repair the link anywhere along the primary route if alternate routes exist. However, the searching range of repair in AODV-LR is wider than AODV-ABR, which means AODV-LR has a higher probability to find an alternate route to the destination than AODV-ABR. So a combination of AODV-ABR with the Local repair algorithm was proposed named as AODV-ABL (ABR and Local repair). When the distance between the broken link and the destination is not farther than MAX_REPAIR_TTL hops, AODV-ABL would try to repair the link by broadcasting a RREQ control signal, just as AODV-LR, and if the broken link is far away from the destination (i.e., the distance is larger than MAX_REPAIR_TTL hops), AODV ABL will repair the link by a handshake process with immediate neighbors.

III. PROPOSED SCHEME

The proposed scheme improves the performance of original AODV protocol by reducing the overhead occurred during the route maintenance. In proposed scheme every node in the network maintains a single

alternate route for sending and receiving data packets. If the link breaks between two intermediate nodes then the upstream node can use the alternate route instead of dropping data packets. The data packet also carries the information about the node, which has an alternate route in its path. In case of link failure and absence of alternate path at that node the data packets are checked for alternate route. If the entry is found then upstream node will transfer the packet to that node which has the alternate route and will send the data packet to the destination. The two phases of proposed scheme for route maintenance in AODV protocol are described below.

A) Alternate Route entry at Node

In the original AODV protocol, every destination sends only one route reply corresponding to one route request and discards all further coming requests. But in our proposed approach, destination will send two route reply messages. Whenever a node listen both replies, it will keep better one in its primary route entry and the worse one in its secondary route entry.

Whenever a link breaks, the upstream node of the broken link will look in its secondary route entry and if it finds an alternate route corresponding to the destination, it will forward the data packet to the destination by using that alternate route. It will send Route Error (RERR) message to the source to re initiate the route discovery process as the path used is not optimal.

B) Alternate route information in Data Packets

If the link breaks and there is no alternate route available at the upstream node then the data packets may be dropped. For handling this type of situations, we maintain two extra fields in the Data packet itself. Backward_ID (BID): Address of the node which is the nearest from the upstream node of the broken link with alternate route to the destination. Backward_Hop_Count (BHC): It is the length of the alternate route to the destination from the node BID. Whenever the source node sends a data packet and it has an alternate route to the destination, it will keep its address in the BID field of the data packet with BHC as the length of the alternate path. But if the source doesn't has an alternate route, then BID will be the source node's address but the BHC field sets as zero which indicates that there is no existence of the alternate route.

Whenever the data packet reaches to an intermediate node, that node checks whether an alternate route exists in its routing table corresponding to the destination. If an alternate route found, then BID field's value is replaced by the address of that intermediate node with BHC as the length of alternate path added with path traversed. In this way, every data

packet is carrying the address of that node as well as hop count to the destination from that node which has an alternate route and which is nearer to the upstream node.

Whenever a link breaks between two nodes, the upstream node of the broken link checks whether it has an alternate route or not. If it finds the route, it will send the data packet by using its alternate route and sends a Route Error (RERR) packet to the source. But if the upstream node does not have an alternate route, then the node will look into the data packet for BID. If it finds an entry corresponding to BID, the node will send the data packet to that node which in turn on receiving the packet forwards it by using the alternate route.

If the upstream node finds BID as source node's address, then it will attempt to find a path by sending a request packet to its neighbors. If any neighbor has any route corresponding to that destination, it will send Route Reply (RREP) packet to the upstream node of the broken link. After finding the path, the upstream node will send data packet to the destination via new discovered path. But if the neighbors don't able to find any route then the upstream node will send an error packet to the source to re initiate the route discovery process.

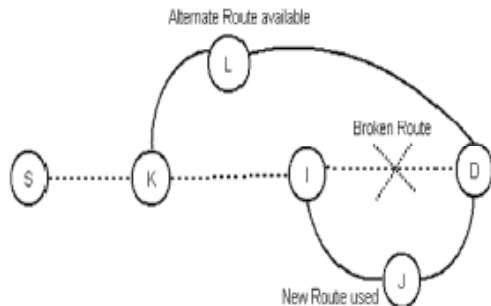


Fig. 1. Alternete route available at upstream node of broken link.

The working of the proposed scheme can be understood by Figure 1. Whenever node I find a link break between itself and node D, it will look in its secondary cache. If it finds an alternate route to D as node J as next hop, then it will send the data packet to the node D via node J. But if node I does not have any alternate route then it will look in to data packet as shown in Figure 2. As per the proposed scheme, the data packet carries the nearest node's address which has an alternate route corresponding to the destination.

The data packet will carry the address of node K as BID because the node K is nearest to upstream node with broken link and has an alternate path. Now when node I finds the link breakage, it will look to data packet

and finds BID as node K's address. So it will use reverse path and sends the data packet to node K which in turn sends data packet by using its alternate route (via node L) to the destination.

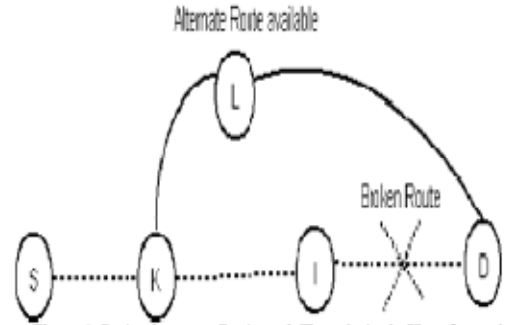


Fig 2. packet is sent to BACKWARD_ID node (k) to forward data using its alternative route.

IV. PERFORMANCE MEASUREMENT

The performance of the proposed scheme is compared with of with original AODV by simulating the results.

A) Simulation Experiment

The proposed scheme was implemented using Network Simulator (NS2). The NS-2 [11][12] [14] is a discrete event driven simulator used for implementation and simulations of various network protocols. In the proposed experiments the network was simulated for 50, 100, and 150 mobile hosts respectively which are placed randomly within a 500 x 500 m² area. Radio propagation range for each node was 250 meters and channel capacity was 2 Mb/s. Each run executed for 200 seconds of simulation time.

A traffic generator was developed to simulate constant bit rate sources. The sources and the destinations are randomly selected with uniform probabilities. There were ten data sessions, each with the traffic rate of four packets per second. The size of data payload was 512 bytes. The random waypoint mobility model was used. Each node randomly selects a position, and moves toward that location with a speed between the minimum and the maximum speed. Once it reaches that position, it becomes stationary for a predefined pause time. After that pause time, it selects another position and repeats the process. We varied the pause time to simulate different mobility degrees. Longer pause time implies less mobility. The speed of nodes movement was varied between zero and 10 mis, respectively.

B) Result and Analysis

The performance of proposed modified scheme was

compared with original AODV by considering three important parameters i.e. Delay, Load and Packet Delivery Ratio (PDR) on a network of 50 mobile hosts, 100 mobile hosts and 150 mobile hosts.

Experiment 1: Network with 50 Mobile hosts

Figure 3 plots the routing load versus pause time for both the schemes. It is evident from Figure 3 that the proposed scheme performs better than original AODV as less number of control packets flow in the modified approach. The comparison of End-to-End delay in original AODV and the modified approach is illustrated in Figure 4. From the figure, it is clear that delay increases in the modified approach. The reason of this increment is that data packet is forwarded back to the previous neighbor and then routed to destination through alternate when a link breaks and the upstream node does not have an alternate route. Similarly, Figure 5 shows the comparison on the basis of packet delivery ratio. From this graph, it is clear that packet delivery ratio decreases at higher mobility and increases at low mobility.

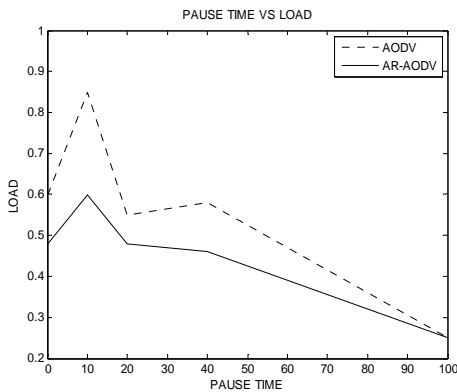


Fig.3: Normalized load (AODV and Enhanced AODV)

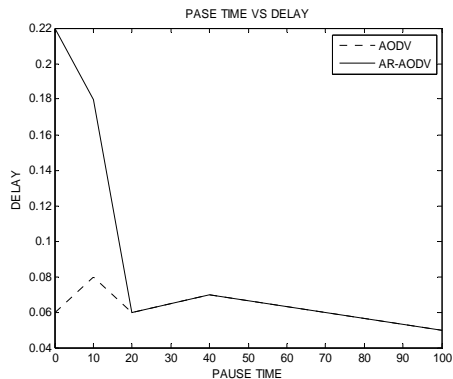


Fig. 4: Average end to end delay (AODV and Enhanced AODV)

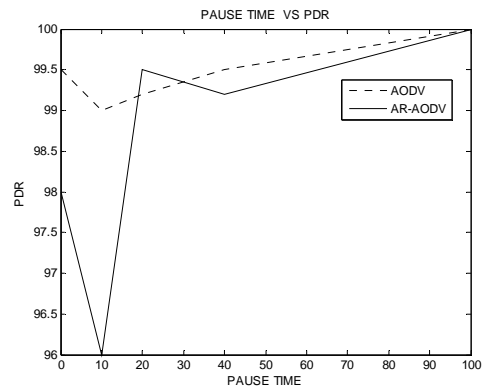


Fig. 5: Packet delivery ratio (AODV and Enhanced AODV)

Experiment 2: Network with 100 Mobile hosts

The performance comparison for the parameters End-to-End delay, routing load, and packet delivery ratio for 100 nodes scenario are respectively shown in Figure 6 through Figure 8.

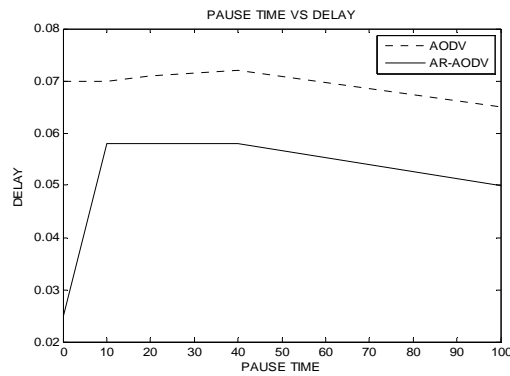


Fig. 6: Average time end to end delay (AODV and Enhanced AODV)

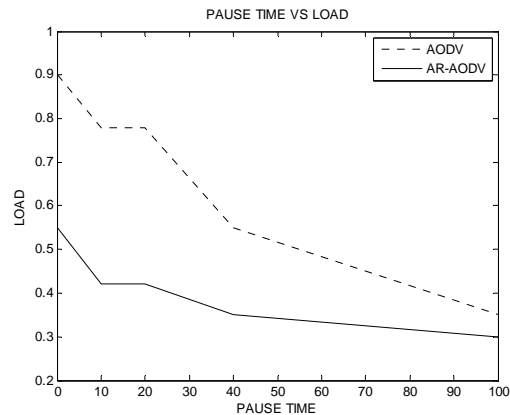


Fig. 7: Normalized load (AODV and Enhanced AODV)

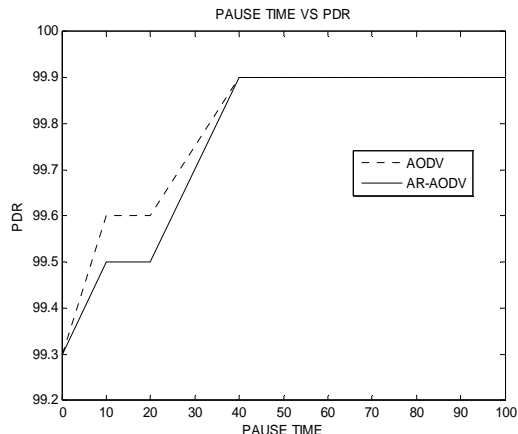


Fig.8: packet delivery ratio (AODV and Enhanced AODV)

It is clear from Figure 6 through 8 that the proposed approach shows the significant improvement in network load and packet delivery ratio. This improvement becomes even more prominent for larger size of networks. In this case we have studied for 50 and 100 nodes.

V. CONCLUSION

The proposed scheme uses alternate path scheme and can be incorporated into any ad hoc on-demand unicast routing protocol to improve reliable packet delivery and enhanced the route maintenance when route breaks due to node mobility. This scheme provides single alternate route which constructed without any extra overhead. Alternate route is utilized only when data packets cannot be delivered through the primary route. Simulation results indicated that proposed scheme provides robustness to mobility and enhances protocol performance. It enhances the packet delivery ratio and reduces the routing load and end to end delay on mobile network.

REFERENCES

- [1] Samir R. Das, Charles E. Perkins, Elizabeth M. Royer and Mahesh K. Marina. "Performance Comparison of Two On-demand Routing Protocols for Ad hoc Networks." IEEE Personal Communications Magazine special issue on Ad hoc Networking, February 2001, p. 16-28.
- [2] Elizabeth M. Royer and Charles E. Perkins. "An Implementation Study of the AODV Routing Protocol." Proceedings of the IEEE Wireless Communications and Networking Conference, Chicago, IL, September 2000.
- [3] Samir R. Das, Charles E. Perkins, and Elizabeth M. Royer. "Performance Comparison of Two On-demand Routing Protocols for Ad Hoc Networks." Proceedings of the IEEE Conference on Computer Communications (INFOCOM), Tel Aviv, Israel, March 2000, p. 3-12.
- [4] Elizabeth M. Royer and Charles E. Perkins. "Multicast Operation of the Ad hoc On-Demand Distance Vector Routing Protocol." Proceedings of MobiCom '99, Seattle, WA, August 1999, pp. 207-218.
- [5] Charles E. Perkins and Elizabeth M. Royer. "Ad hoc On-Demand Distance Vector Routing." Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications, New Orleans, LA, February 1999, pp. 90-100.
- [6] Charles E. Perkins, Elizabeth M. Belding-Royer, and Samir R. Das. Ad hoc on-demand distance vector (aodv) routing. RFC Experimental 3561, Internet Engineering Task Force, July 2003.
- [7] Sung-Ju Lee and Mario Gerla. AODV-BR: Backup routing in ad hoc networks. In Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC 2000), Chicago, IL, September 2000.
- [8] Wei Kuang Lai, Sheng-Yu Hsiao, and Yuh-Chung Lin. Adaptive backup routing for ad-hoc networks. *Comput. Commun.*, 30(2):453-464, 2007.
- [9] Michael Pan, Sheng-Yan Chuang, and Sheng-De Wang. Local repair mechanisms for on-demand routing in mobile ad hoc networks. In PRDC '05: Proceedings of the 11th Pacific Rim International Symposium on Dependable Computing, pages 317-324, Washington, DC, USA, 2005. IEEE Computer Society.
- [10] Editor Kevin Fall and Editor Kannan Varadhan. The network simulator ns-2: Documentation. <http://www.isi.edu/nsnam/ns-documentation.htm>.
- [11] The Network Simulator NS-2. <http://www.isi.edu/nsnam/>
- [12] C. E. Perkins, E. M. Royer, and S. R. Das, "Ad Hoc on Demand Distance Vector (AODY) Routing, <http://www.ietf.org/internet-drafts/draft-ietfmanet-aodv-06.txt>, IETF Internet Draft, July 2000, work in progress.
- [13] G.S. Tomar, Manish Dixit & Shekhar Verma, "AODV Routing Protocol With Selective Flooding", IEEE SOCPAR 2009, pp 682-686, 2009
- [14] Rendong Bai and Mukesh Singhal, "DOA: DSR over AODV Routing for Mobile Ad Hoc Networks" IEEE Trans on Mobile Computing, Vol.1.5, No.10, 2006.
- [15] S. Crisostomo, S. Sargento, P. Brandao and R. Prior, "Improving AODV with Preemptive Local Route Repair," Proc. of IEEE IWWAN, 2004.
- [16] D.B. Johnson and D.A. Maltz, "Dynamic Source Routing in Ad Hoc Wireless Networks," Mobile Computing, vol. 353. Kluwer Academic, 1996.