Indian Journal of Fundamental and Applied Life Sciences ISSN: 2231– 6345 (Online)
An Open Access, Online International Journal Available at www.cibtech.org/sp.ed/jls/2015/01/jls.htm
2015 Vol.5 (S1), pp. 396-403/Yari et al.
Research Article

AN OPTIMAIZED ROUTING PROTOCOL FOR REDUCE CONTROL IN AD HOC NETWOEKS

***Meysam Yari[1], Amin Sargazi[2], Bahram Arbabi[3] and Naghme Egtedari[2]**
[1]*Department of Engineering, Germi Branch, Islamic Azad University, Germi, Iran*
[2]*Department of Engineering, Ferdowsi University, Mashhad, Iran*
[3]*Department of Engineering, Sistan Balouchestan, Branch, Islamic Azad University, Sistan Balouchestan, Iran*
**Author for Correspondence*

ABSTRACT
Mobile ad hoc networks consist of a number of mobile nodes connected by wireless links which are free to move. Routing in such networks is very complex due to the high mobility of nodes, the limited bandwidth and power supply limitation and the limited coverage radius of nodes. Many routing protocols have been proposed for ad hoc networks. Each of the protocols (which include a wide range of design techniques) has its own advantages and disadvantages, and can therefore be useful for specific environments with their own particular conditions. In this paper, among the protocols suitable for the conditions of high congestion and fast mobility of nodes, we choose the appropriate type (AODV) and then provide a method for route backup, to reduce the control overhead and improve the performance of the protocol. The simulation results show that the proposed method has high performance compared to similar approaches.

Keywords: *Control Overhead, Alternate Routes, Local Repair, Route Back up, Routing Protocol*

INTRODUCTION
Ad hoc networks (Lin *et al.*, 2003) are a new class of wireless networks, which have attracted much attention because of their unique characteristics. In these networks, there is no base, amplifier and fixed switching center, but rather the nodes themselves perform the operation to improve switching and routing data. Since continuous changes occur in the network topology because of the mobility of nodes, they must be supported by the routing protocols (Boukerche *et al.*, 2011) by adopting a consistent strategy so that the transmitted data can be safely delivered to the destination. In recent years, much work has been done on routing in mobile ad hoc networks, especially in the area of route backup, to reduce the control overhead. Figure 1 shows the AODV-BR technique (Lee and Gerla, 2000) (Lee *et al.*, 1999), which represents the route backup using alternate routes of neighbors to destination (ABR). In the method presented in AODV-ABL (Kuang *et al.*, 2007), since the algorithm is obtained by combining local repair (LR) protocols using alternate routes of neighbors to the destination (ABR), the two LR and ABR approaches were merged into this algorithm. In the proposed method, if the number of hops to the destination is less than the MAX-REPAIR-TTL, the ABL algorithm acts as LR, and otherwise it uses the handshake technique (ABR) for the route repair. The method presented in AR-AODV (Patil *et al.*, 2012) can improve the proposed protocol by reducing the overhead due to the route maintenance of AODV algorithm (Parkins *et al.*, 2003) (Sarikhanbigloo *et al.*, 2014). In the protocol, each node in the network maintains a single alternate route for transmitting and receiving data packets. If the route between two intermediate nodes is broken, the transmitting node can use the alternate route, instead of dropping the packet. Data packet carries information about the node whose route had the alternate route. The method presented in BFABL (Zhou and Li, 2012) has made two overall improvements for bidirectional application scenarios. First, it merges the two primary and secondary routing tables, *i.e.* a single entry for each destination is available in the table. When better alternate routes were heard with corresponding entries in the routing table, it can be replaced as soon as possible. Secondly, BFABL hears the data packet on both sides - from origin to destination and vice versa, to avoid the one-hop routes to the destination to be broken. In the next section, the method presented in AR-AODV is fully presented.

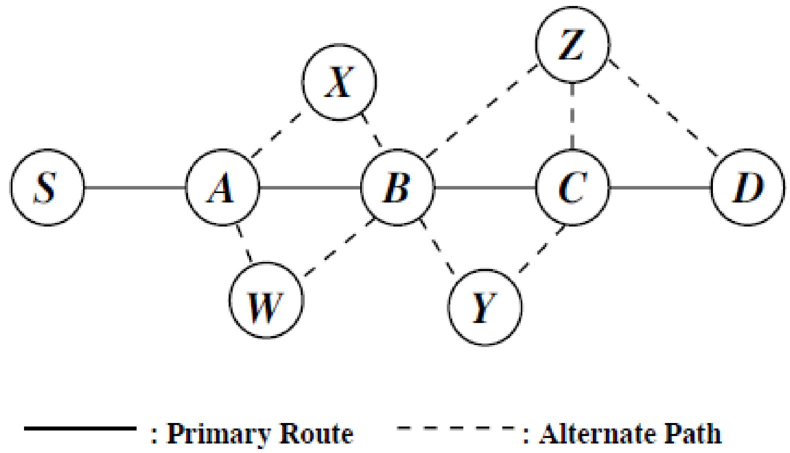

Figure 1: An alternate with the primary route (Lee and Gerla, 2000)

```
TapData ( Packet p){
   if(routing entry to D is not exist || routing entry to S is not exist){
      creates routing entry to D  or(and) S;
      update those newly created routing  entry;
      return;
   }
else{
      analyses packet head of p, look for
        (1)get hops count to D and S;
        (2)sequence number of  D and S;
       then calculates the new route to S and the new route to D;
       finds out corresponding routing entry to D and S;
       compares  the new route to the route in routing table.
       if (the sequence number in new route is less than that in routing
table entry || (the sequence number in new route is equal to that in routing
table entry && the hops is no more than that in table entry))
        {
          updates the corresponding routing entry with new route;
        }
     }
}
```

Figure 2: AODV-BFABL pseudo code of overhearing data packets

AR-AODV

The proposed scheme (Patil *et al.*, 2012) 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 blow.

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 The working of the proposed scheme can be understood by Figure 3.

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 1. But if node I do not have any alternate route then if it will look in to data packet as shown in Figure 4. 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.

Indian Journal of Fundamental and Applied Life Sciences ISSN: 2231– 6345 (Online)
An Open Access, Online International Journal Available at www.cibtech.org/sp.ed/jls/2015/01/jls.htm
2015 Vol.5 (S1), pp. 396-403/Yari et al.
Research Article

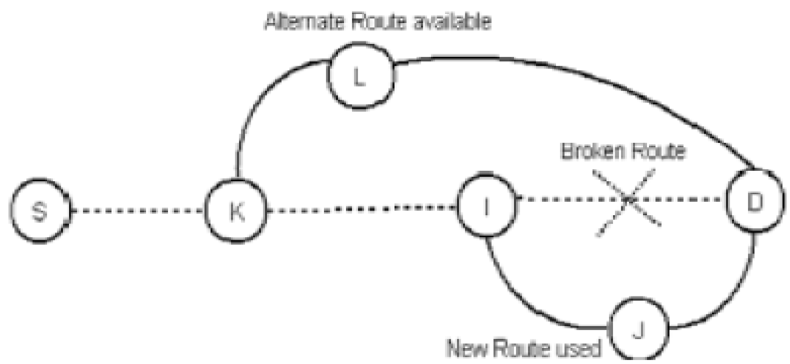

Figure 3: Alternete route available at upstream node of broken link (Patil *et al.*, 2012)

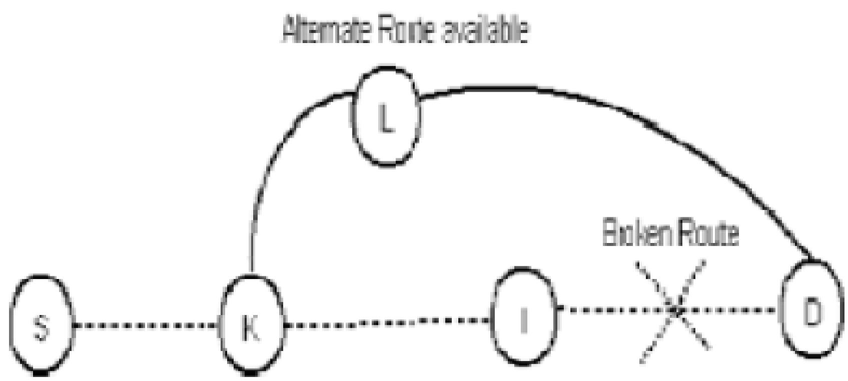

Figure 4: Packet is sent to BACKWARD_ID node (k) to forward data using its alternative route

The Proposed Method

To reduce the control overhead in mobile ad hoc networks, we can use the ideas used in AODV-ABL (Kuang *et al.*, 2007) and AR-AODV (Patil *et al.*, 2012) routing protocols, in this way that we separate the entire routing operations into two phases. First, we act as in the AR-AODV algorithm and try to use the alternate route that is available in the data packet or the node itself; then, if there is no route or the alternate route was expired, we move to the second phase, which is to use the ABL protocol, namely local repair (LR), or to use alternate routes of one-hop neighbors (ABR) (Lee and Gerla, 2000). In this case, we avoid from sending control messages as far as possible. In addition, to update tables in real time, we can use the idea of AODV-BFABL protocol (Zhou and Li, 2012), *i.e.*, we forced the nodes to hear the packets sent by their neighbors (which are located at its communication distance), with the difference that, unlike BFABL (which makes use of a routing table), two routing tables are used, as in the ABL protocol. When the data packet is heard, it convinces the node to focus the hop counts between the source and destination as well as the sequence number of the source and destination. The information is compared with corresponding information of the same origin and destination in the primary routing tables; and if new information are optimized, the routing entry will be transmitted from the primary table to the secondary table (the entry of previous alternate route is removed) and the newly discovered route will be immediately replaced by the corresponding entry in the primary table.

Simulation

In this section, three protocols mentioned in paper have been simulated using the network simulator NS-2 [6] for two different scenarios. Random way point is used as the model of node mobility, and the area of simulation environment is assumed to be a square with a length of 1000 meters. Simulation time is 1000 and 500 seconds, the packet size is 512 bytes, and the bandwidth is equal to 10 mbps. In addition, the pause time is considered to be zero. The pause time of nodes is considered to be between 0 to 400 seconds in the simulation of the first scenario, and the number of nodes in the simulation of the second scenario ranges from 60 to 100 (60, 70, 80, 90, 100).

Simulation Results

Simulation of the protocols mentioned above was performed in two different cases as follows, based on the pause time of nodes and the number of nodes.

• ***Simulation at the Pause Time of Nodes***

This scenario assumes the environment size of 1000 meters in 1000 meters, the pause time of 0s to 400s for the nodes (0, 50, 100, 200, 300, 400), and 50 nodes. In this case, the bandwidth is equal to 10 mbps and the simulation time is considered 500 seconds.

• ***Simulation with Different Numbers of Nodes***

The scenario considers 60 to 100 nodes (60, 70, 80, 90, and 100), the mobility speed of 10 m/s for the nodes and a simulation environment dimension of 1,000 meters by 1,000 meters. The pause time and simulation time are considered equal to 0 and 1000 seconds, respectively; and the bandwidth is 10 Mbps.

Simulation Metrics

To evaluate and compare the performance of the routing protocols mentioned in the paper, the three metrics of average delivery ratio (rate of data packets received), average end-to-end delay time, and control overhead were evaluated in two different scenarios of simulation, and then the simulation results and the corresponding charts were examined for each of the proposed metrics.

• ***The Simulation Results of the First Scenario***

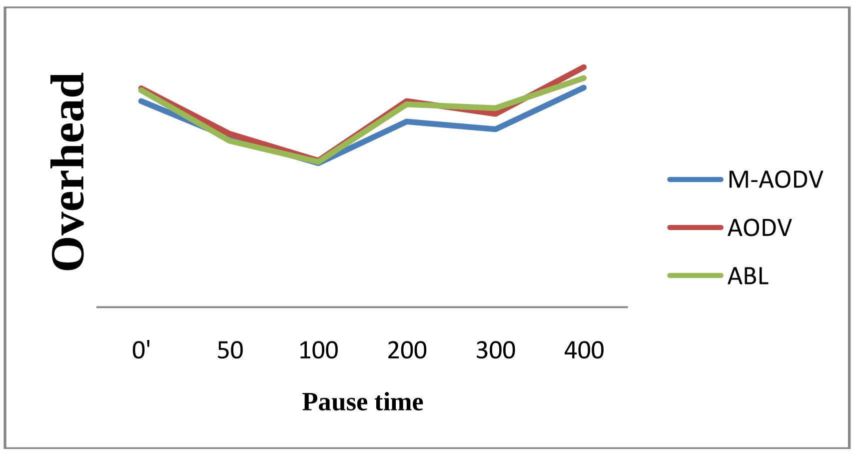

Figure 5: Control overhead

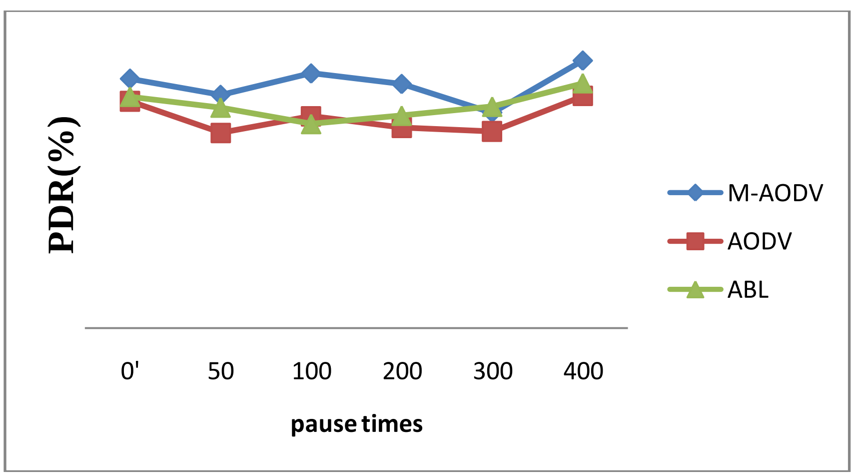

Figure 6: Packet delivery ratio

Research Article

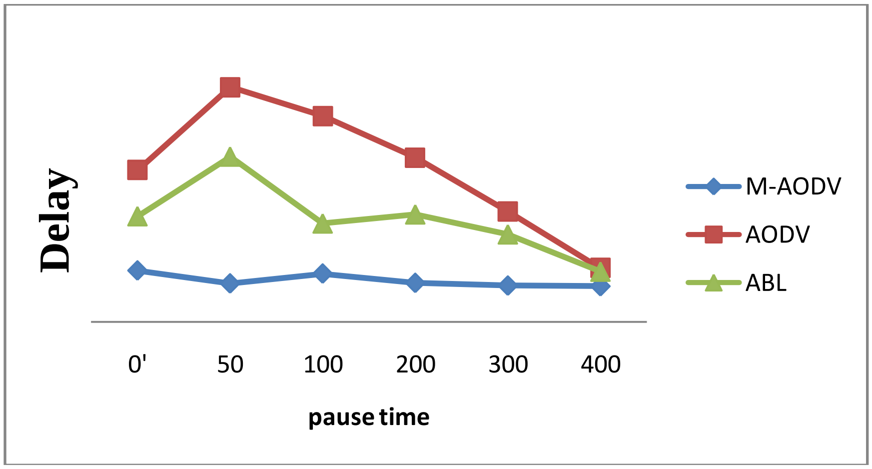

Figure 7: Average End to End delay

- ***The Simulation Results of the Second Scenario***

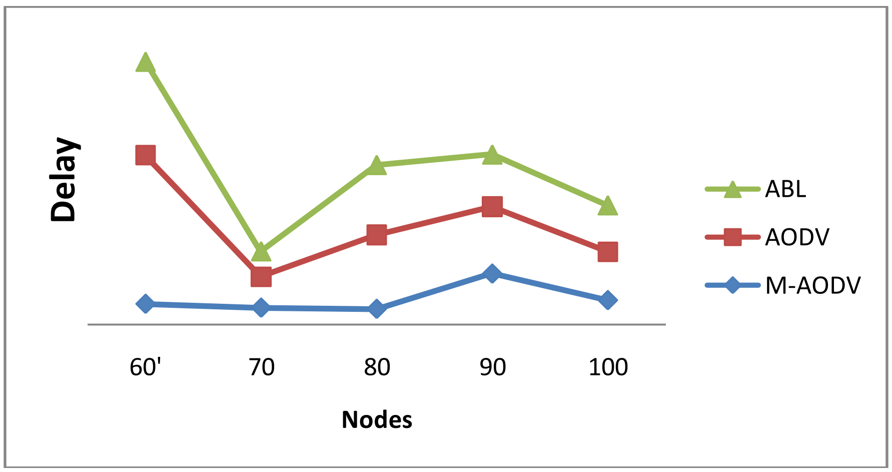

Figure 8: Average End to End delay

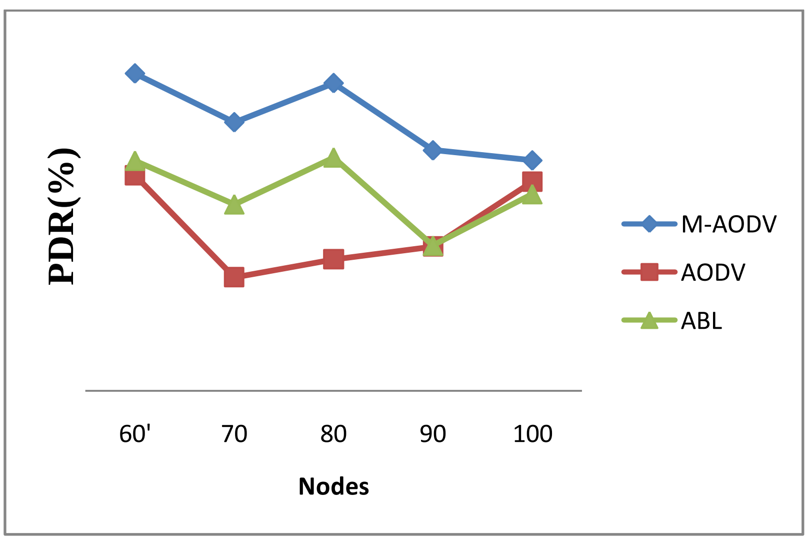

Figure 9: Packet delivery ratio

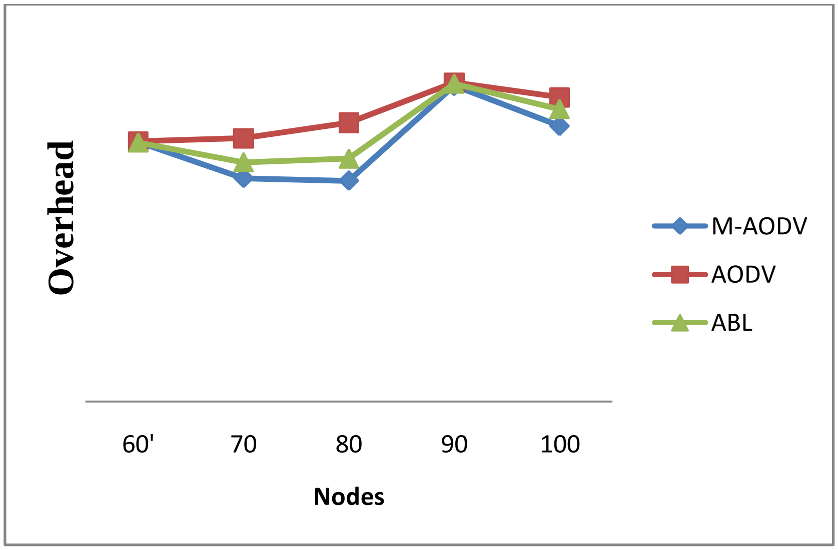

Figure 10: Control overhead

CONCLUSION

With the help of the network simulator NS-2, three AODV, ABL and M-AODV routing protocols were evaluated and compared in two quite different scenarios. According to the results obtained from the simulation and analysis of the three metrics listed above, it can be concluded that since the M-AODV routing protocol (the proposed approach) chooses the best and optimal route by using the alternate route available in the data packet or alternate route available in the node itself as well as with using local repair mechanism, or by using alternate routes of neighboring nodes with one-hop distance, it results in lower delay and control overhead and higher delivery ratio, compared to ABL and AODV, and has outstanding performance in the ad hoc network.

ACKNOWLEDGMENTS

We thank Prof. Yuh-chung-lin at the Pingtung University in Taiwan, who gave us valuable advice for improving this paper and shared the details of his own article.

REFERENCES

Boukerche A, Turgut B, Aydin N, Ahmad MZ and Blni L *et al.*, (2011). Routing Protocols in Ad Hoc Networks: A Survey. *Computer Networks* **55**(Sep) 3032-3080. doi:10.1016/j.comnet. 2011.05.010

Lee SJ, Gerla M and Chiang CC (1999). On-Demand Multicast Routing Protocol. *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC), New Orleans, LA*, September 1999 1298-1302.

Lee SJ, Royer E and Perkins CE (2003). Scalability study of the ad hoc on demand distance vector routing protocol. *International Journal of Network Management* **13**(2) 97–114.

Patil VP, Patil KT, Kharade AR and Gote DD (2012). Performance Enhancement of Reactive on Demand Routing Protocol in Wireless Ad Hoc Network. *International Journal of Smart Sensors and Ad Hoc (IJSSAN)* **1**(4).

Peng Zhou and Weihua Li (2010). A Bidirectional Backup Routing Protocol for Mobile Ad Hoc Networks 603–606. Available: http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=6380634 [Accessed on 12-14 October 2012].

Perkins CE, Royer EM and Das S (2003). Ad hoc on-demand distance vector (AODV) routing. IETE RFC 3561, 2003.

Sung-Ju Lee and Mario Gerla (2000). AODV-BR: Backup Routing in Ad hoc Networks. *Wireless Communications and Networking Confernce WCNC, IEEE* **3** 1311 – 1316.

Tao Lin, Scott F Midkiff and Jahng S Park (2003). A framework for Wireless Ad hoc Routing Protcols. *Wireless Communications and Networking, WCNC, IEEE* **2**.

Wei Kuang Lai A, Sheng-Yu Hsiao A and Yuh-Chung Lin (2007). Adaptive backup routing for ad-hoc networks. *Computer Communications* **30**(2) 453–464.

Yari Sarikhanbigloo, Meysam Mohsen Mohammadi and Zohre Asgharzadeh (2014). Comparison and Evaluation of AODV and DSR Routing Protocols in Ad Hoc Mobile Networks with high Bandwidth. *National Conference on Computer Engineering and Information Technology Management, Tehran, Tolou Farzin Science and Technology Company*. Available: http://www.civilica.com/Paper-CSITM01-CSITM01_148.html