SIMULATION AND PERFORMANCE ANALYSIS OF AD HOC ON-DEMAND MULTIPATH DISTANCE VECTOR ROUTING PROTOCOL (AOMDV) IN NS-2

Dr. Manpreet Singh *

Rajneesh Gujral **

Alpna Bansal ***

ABSTRACT

An ad hoc network is a collection of wireless mobile hosts forming a temporary network without the aid of any established infrastructure or centralized administration. In such an environment, it may be necessary for one mobile host to transfer large amount of data through other hosts to its destination but problem arises due to the limited bandwidth, congestion and also occur excessive delay. In this paper, we present a multipath protocol for routing in ad hoc networks, that uses on-demand distance vector routing, called Ad hoc On-demand Multipath Distance Vector (AOMDV) Routing Protocol. It eliminates the need for further routing when there is a broken link in the path. Hence reduces delay and provide required end-to-end bandwidth. We also analyze its performance using NS-2. Results shows that AOMDV has better efficiency in case of throughput of number of packets received or number of packets send. We performed several experiments in order to study the performance of AOMDV.

* Professor, Department of Computer Science and Engineering, Maharishi Markandeshwar Engineering College, Mullana.

** Associate Professor, Department of Computer Science and Engineering, Maharishi Markandeshwar Engineering College, Mullana.

*** Department of Computer Science and Engineering, Maharishi Markandeshwar Engineering College, Mullana.

1. INTRODUCTION

Wireless networking is a technology that enables two or more computers to communicate using standard network protocols, but without network cabling [1]. Now, there exist a large number of network protocols that are developed just for the purpose of Wireless networks. Wireless communication technology have been developed with two primary models one is fix infrastructure based model in which much of the nodes are mobile and connected through fixed backbone nodes using wireless medium. Another model is Mobile Ad-hoc network. Mobile Ad-Hoc Networks (MANETs) are comprised of mobile nodes (MNs) that are self-organizing and cooperative to ensure efficient and accurate packet routing between nodes (and, potentially, base stations). Design issue for developing a routing protocol for wireless environment with mobility is very different and more complex than those for wired network with static nodes. Main problem in mobile ad hoc network are Limited bandwidth and frequently change in the topology. Although there are lots of routing protocols that can be used for the unicast and multicast communication within the Mobile Ad hoc networks, it observes that any one protocol cannot fit in all the different scenarios, different topologies and traffic patterns of Mobile Ad-Hoc Networks applications. For instance, proactive routing protocols are very useful for a small- scale MANETs with high mobility, while reactive routing protocols are very useful for a large-scale, MANETs with moderate or less topology changes. Hybrid routing protocol attempts to strike balance between the two such as proactive for neighborhood, reactive for far away.

1.1 Need of multipath routing

- **Balancing Energy Consumption:** To balance the energy consumption, we make the source node function as the packet balancing agent which collects energy information of paths and decides sending packets via the path with the best energy condition, namely, the path with the maximal energy metric.
- **Reduce Congestion:** Multipath protocols may cause fewer interruptions to the application data traffic when routes fail. They also have the potential to lower the routing overhead.
- **Reduce End-to-End Delay:** When transfer data over all discovered paths concurrently, it reduces end to end delay and increases end to end bandwidth.
- **Increases Data Packet Throughput:** Multipath routing can increase end-to-end throughput and provide load balancing in wired networks.
- **Reduce Route discovery Frequency:** Multi-path on-demand routing algorithms

discover several paths instead of one, once the routing is performed [6]. This eliminates the need for further routing when there is a broken link in the path, reducing the average number of Route Discovery for each node and achieving higher fault tolerance for the Mobile Ad-hoc networks.

2. LITERATURE SURVEY

Alvin Valera et al [8] introduced data packet caching in the context of mobile ad hoc networks. They proposed a new routing protocol called Caching and Multipath (CHAMP) Routing protocol. CHAMP uses cooperative packet caching and shortest multipath routing to reduce packet loss due to frequent route breakages. Performance improvements in terms of higher packet delivery, lower delays and reduced routing overhead are obtained due to the temporal locality of dropped packets.

Asis Nasipuri et al [10] extend DSR to compute multiple link disjoint paths for overhead reduction in mobile networks. Besides, they also use analytical modeling to study the effect of number of multiple paths and path lengths on on-demand routing performance. Their modeling effort shows that while multipath routing is significantly better than single path routing, the performance advantage is small beyond a few paths and for long path lengths.

Tom Goff et al [17] extended DSR and AODV for preemptive maintenance. Modified DSR and AODV algorithms are called Preemptive Dynamic Source Routing (PDSR) and Preemptive Ad hoc On-demand Distance Vector (PAODV) respectively. Preemptive routing is another mechanism which proactively repairs routes by monitoring the likelihood of a path break and informing the source which will initiate an early route discovery.

3. AD HOC ON-DEMAND MULTIPATH DISTANCE VECTOR (AOMDV) ROUTING

AOMDV routing protocol is the extension of AODV protocol to compute multiple disjoint loop-free paths in a route discovery. It has several characteristics. It is based on the distance vector concept and uses hop-by-hop routing approach. It also finds routes on demand using route discovery procedure. RREQ propagation from the source towards the destination establishes multiple reverse paths both at intermediate nodes as well as the destination. Multiple RREPs traverse these reverse paths back to form multiple forward paths to the destination at the source and destination nodes. The core of the AOMDV protocol lies in ensuring that multiple paths discovered are loop-free and disjoint, and in efficiently finding such paths using a flood-based route discovery. AOMDV route update rules, applied locally at each node, play a key role in maintaining loop-freedom and disjointness properties.

3.1. Loop freedom

Two issues arise when computing multiple loop-free paths at a node for a destination. First, which one of the multiple paths should a node offer or advertise to others? Since each of these paths may have different hop counts, an arbitrary choice can result in loops. Second, which of the advertised paths should a node accept? Again, accepting all paths naively may cause loops.

Based on the above, we formulate below a set of sufficient conditions for loop-freedom.

- *Sequence number rule*: Maintain routes only for the highest known destination sequence number. For each destination, we restrict that multiple paths maintained by a node have the same destination sequence number. Once a route advertisement containing a higher destination sequence number is received, all routes corresponding to the older sequence number are discarded.
- For the same destination sequence number,

  (a) *Route advertisement rule*: Never advertise a route shorter than one already advertised.

  (b) *Route acceptance rule*: Never accept a route longer than one already advertised.

To maintain multiple paths for the same sequence number, AOMDV uses the notion of an '*advertised hop count*.' Every node maintains a variable called advertised hop count for each destination. This variable is set to the length of the longest available path for the destination at the time of first advertisement for a particular destination sequence number. The advertised hop count remains unchanged until the sequence number changes. Advertising the longest path length permits more number of alternate paths to be maintained.

3.2. Disjoint paths

Besides maintaining multiple loop-free paths, AOMDV seeks to find disjoint alternate paths. For our purpose of improving fault tolerance using multiple paths, disjoint paths are a natural choice. We consider two types of disjoint paths: link disjoint and node disjoint. Link disjoint set of paths between a pair of nodes have no common links, whereas node-disjoint means no common intermediate nodes. In distributed routing algorithms of the distance vector type, a node forms paths to a destination incrementally based on paths obtained from downstream neighbors towards the destination. So finding a set of link disjoint paths at a node can be seen as a two step process: First, identifying a set of downstream neighbors having mutually link disjoint paths to the destination; Second, forms exactly one path via each of those downstream neighbors. Note that the second step is trivial –the node simply needs to ensure that every path has a unique next hop, which is purely local operation. However, performing the first step requires knowledge of some or all downstream nodes on each path.In a typical distance vector protocol, a node only keeps track of the next hop and distance via the next hop for each path. This limited one hop information is insufficient for a node to ascertain whether two paths obtained from two distinct neighbors are indeed link disjoint.

Two observations for link disjoint paths are:

If two paths from a node P to a destination D are link disjoint, then they must have unique next hops as well as unique last hops. Note that the converse of this observation is not necessarily true. However, the converse also holds true in general with an additional restriction:

If every node on a path ensures that all paths to the destination from that node differ in their next and last hops. This implication provides us with a tool to determine whether two paths via two unique downstream neighbors are link disjoint. They simply need to have unique last hops

3.3 Detailed Protocol Description

We describe the protocol in four components: routing table structure, route discovery, route maintenance, and data packet forwarding. Here we describe the link disjoint version of the protocol in detail, so all references to disjointness actually imply link disjointness. A straightforward modification to this protocol yields node disjoint paths instead.

3.3.1. Routing table

Figure 1 shows the difference in the routing table entry structure between AODV and AOMDV. AOMDV route table entry has a new field for the advertised hop count. Besides a route list is used in AOMDV to store additional information for each alternate path including: next hop, last hop, hop count, and expiration timeout. As already discussed, last hop information is useful in checking the disjointness of alternate paths.

destination	Sequence number	Advertised hop count	Route list			
			$next\_hop_1$ $next\_hop_2$ . .	$last\_hop_1$ $last\_hop_2$ . .	hop_count_1 hop_count_2 . .	$timeout_1$ $timeout_2$ . .

Figure 1: Routing table entry structure in AOMDV

Consider a destination *d* and a node *i*. Whenever the destination sequence number for *d* at *i* is updated, the corresponding advertised hop count is initialized. For a given destination sequence number, let $hop\_count_{ik}^{d}$ denote the hop count of *k*th path (for some *k*) in the routing table entry for *d* at *i*, that is ($next\_hop^{d}_{ik}$, $last\_hop^{d}_{ik}$, $hop\_count^{d}_{ik}$) € $route\_list^{d}_{i}$. When i is about to sends its first route advertisement for d, it updates the advertised hop count as follows:

$$advertised\_hop\_count^{d}_{i} := \max_{k} \{hop\_count^{d}_{ik}\},\ i \neq d$$

$$:= 0, \text{ otherwise}$$

Whenever a node receives a route advertisement, it invokes the AOMDV route update rules listed in Figure 2. Note that lines (1) and (10) in Figure 2 ensure loop freedom, whereas lines (12) and (15) check for link disjointness.

1: **if** ($seq\_num^{d}_{i} < seq\_num^{d}_{j}$) **then** {/* enforces the sequence number rule*/}

2: $seq\_num^{d}_{i} := seq\_num^{d}_{j}$;

3: $advertised\_hop\_count^{d}_{i} := \infty$;

4: $route\_list^{d}_{i} := NULL$;

5: **if** ($j=d$) **then** {/* neighbor is the destination*/}

6: insert($j,i,1$) into $route\_list^{d}_{i}$;

7: **else**

8: insert($j, last\_hop^{d}_{jk}, advertised\_hop\_count^{d}_{j}+1$) into $route\_list^{d}_{i}$;

9: **end if**

10: **else if** (($seq\_num^{d}_{i} = seq\_num^{d}_{j}$) and ($advertised\_hop\_count^{d}_{i} > advertised\_hop\_count^{d}_{j}$))

then {/* enforces the route acceptance rule*/}

11: **if** ($j=d$) **then** {/* neighbor is the destination*/}

12: **if** (($\exists\, k_1 : (next\_hop^{d}_{ik_1} = j)$) **and** ($\exists\, k_2 : (last\_hop^{d}_{ik_2} = i)$))

then {/* establishes uniqueness of next and last hops */}

13: insert($j,i,1$) into $route\_list^{d}_{i}$;

14: **end if**

15: **else if** (($\exists\, k_3 : (next\_hop^{d}_{ik_3} = j)$) **and** ($\exists\, k_4 : (last\_hop^{d}_{ik_4} = last\_hop^{d}_{jk}$))

then {/* establishes uniqueness of next and last hops */}

16: insert($j, last\_hop^{d}_{jk}, advertised\_hop\_count^{d}_{j}+1$) into $route\_list^{d}_{i}$;

17: **end if**

18: **end if**

Figure 2: AOMDV route update rules.

3.3.2. Route Discovery

When a traffic source needs a route to a destination, the source initiates a route discovery process by generating a RREQ as shown in figure 3(a). Since the RREQ is flooded network-wide, a node may receive several copies of the same RREQ. In AODV[2]only the first copy of the RREQ is used to form reverse paths; the duplicate copies that arrive later are simply discarded as shown in figure 3(b). Note that some of these duplicate copies can be gainfully used to form alternate reverse paths. Thus, all duplicate copies are examined in AOMDV for potential alternate reverse paths, but reverse paths are formed only using those copies that preserve loop-freedom and disjointness among the resulting set of paths to the source as shown in figure 3(d).

When an intermediate node obtains a reverse path via RREQ copy, it checks whether there are one or more valid forward paths to the destination. If so, the node generates a RREP and sends it back to the source along the reverse path; the RREP includes a forward path that was not used in any previous RREPs for this route discovery. In this case, the intermediate node does not propagate the RREQ further. Otherwise, the node re-broadcasts the RREQ copy if it has not previously forwarded any other copy of this RREQ and this copy resulted in the formation or updation of a reverse path.

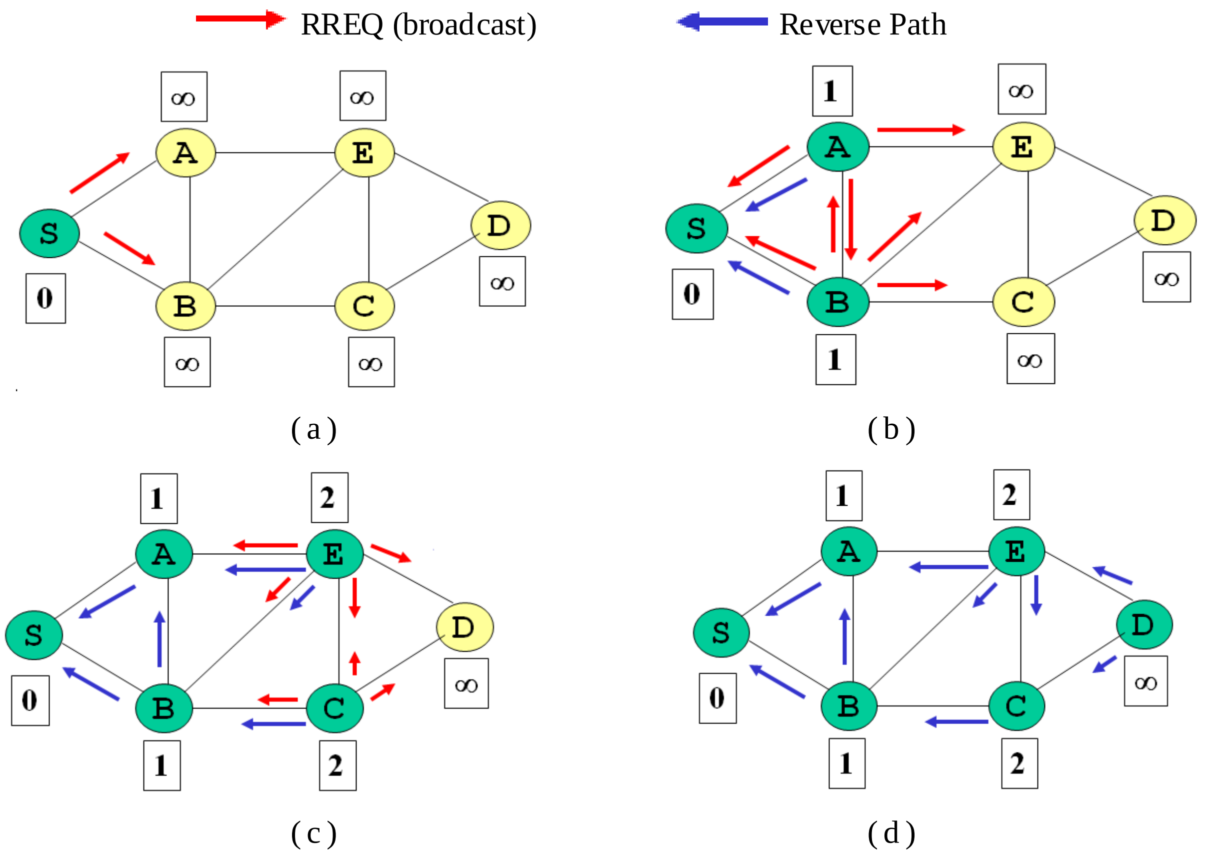

Figure 3: AOMDV Route Discovery

3.3.3. Route maintenance

A node generates or forwards a RERR for a destination when the last path to the destination breaks. AOMDV also includes an optimization to salvage packets forwarded over failed links by re-forwarding them over alternate paths. This is similar to the packet salvaging mechanism in DSR [3]. The timeout mechanism similarly extends from a single path to multiple paths (Figure 1) although the problem of setting proper timeout values is more difficult for AOMDV compared to AODV[2]. With multiple paths, the possibility of paths becoming stale is more likely. But using very small timeout values to avoid stale paths can limit the benefit of using multiple paths. In our experiments, we use a moderate setting of timeout values and additionally use HELLO messages to proactively remove stale routes. Thus, the timeouts in the current version of AOMDV primarily serve as a soft-state mechanism to deal with unforeseen events such as routing table corruption and to a lesser extent for promptly purging stale routes.

3.3.4. Data packet forwarding

For data packet forwarding at a node having multiple paths to a destination, we adopt a simple approach of using a path until it fails and then switch to an alternate path; we use paths in the order of their creation. There are other alternatives for data packet forwarding which concurrently use all paths. With 'diversity coding' [19], an overhead is added to each data packet (coding) and the resulting coded packet is split into smaller blocks each of which is transmitted along a different path. With adequate redundancy, this scheme can improve the packet delivery probability in highly dynamic mobile networks. This scheme can also be employed in a selective way to ensure delivery of important packets.

4. SIMULATION BASED ANALYSIS OF AOMDV ROUTING PROTOCOL

We implement AOMDV routing protocol using network simulator ns-2.34. After executing .tcl files .tr (trace files) are generated. To filter the data from trace files, we have used simple grep command and awk filter both. **grep** is a command to search a particular pattern from a file or a string. grep is search **G**lobally for lines matching **R**egular **E**xpression and **P**rint them. **awk** is a good filter and a report writer. It is a pattern searching and processing language. The name was composed from the initial letters of three original authors Alfred V. Aho, Brian W. Kernighan, and Peter J. Weinberger.

Simulation environment of network simulator ns-2.34 is described below by table1.

Table 1: Simulation Parameters

Parameter	Value
Simulation Time	900 seconds
Network Size	1000*1000(m^2)
Number of Nodes	16
Transmission Range	250 m
Traffic type	Constant Bit Rate
Traffic rate	5.4 Mb/s

Parameters Measured:

Throughput is a measure of the amount of data that can be sent over a link in a given amount of time as shown in figure 4. The throughput is determined by the formula:

$$\text{Throughput} = \frac{\text{Data Transferred}}{\text{Simulation Time}}$$

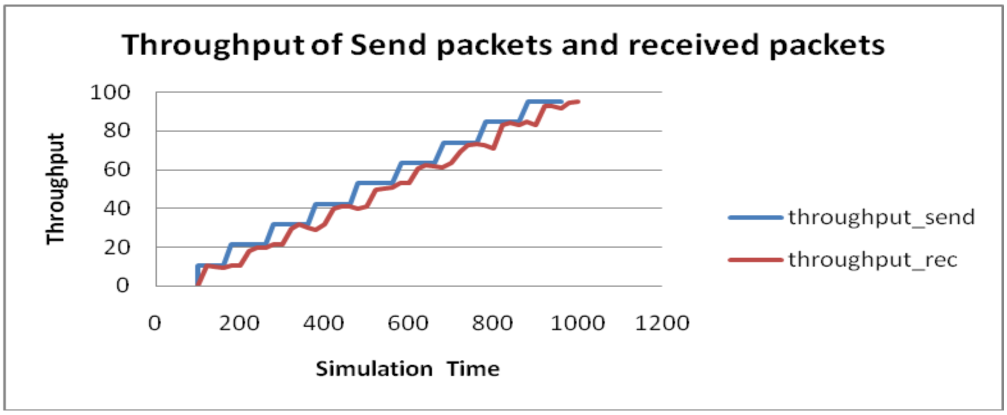

Figure 4: Throughput of send and receive packets

Packet delivery ratio is the ratio of the number of data packets actually delivered to the number of data packets are supposed to be received. We calculate the packet delivery ratio for AOMDV with the scenario of different number of nodes. Graph is shown in figure 5.

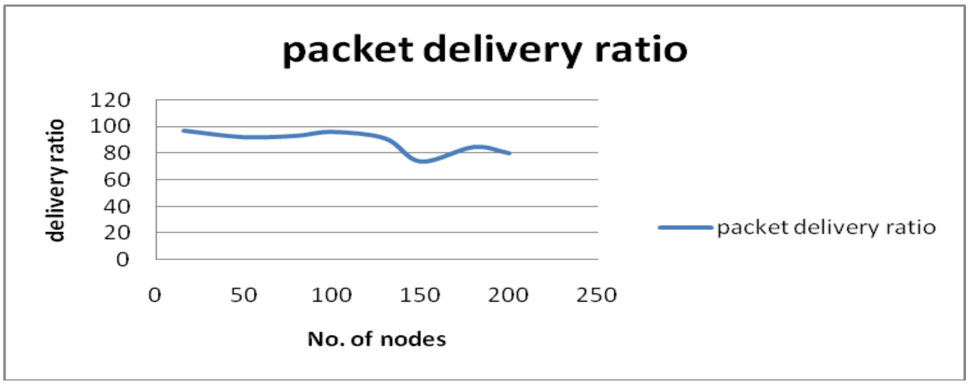

Figure 5: Packet delivery ratio (no. of nodes variabilty)

After Simulation, some of the parameters calculated through awk filter are shown in table 2.

Table 2: Simulation Result

Parameter	Value
Total number of packets sent	47467
Total number of packets received	46180
Average throughput of number of sent packets	49.444
Average throughput of number of received packets	48.104

5. CONCLUSION AND FUTURE SCOPE

In this paper, we have simulated an on-demand multipath protocol called AOMDV. It ensures loop-free and disjoint multiple paths. We have studied the performance of AOMDV using ns-2 simulations under number of nodes variability. Several additional issues related to the design and evaluation of AOMDV protocol may require further investigation. Protocol can be improved to effectively deal with the route cutoff problem and compute more disjoint paths when source-destination pairs are far apart.

6. REFERENCES

1. Vicomsoft, "Knowledge share whitepapers wireless networking Q&A", Vicomsoft connect and protect, Jan, 2003
2. Charles E. Perkins and Elizabeth M. Royer, "Ad hoc on demand distance vector (AODV) routing (Internet-Draft)", Aug,1998
3. Johnson DB, Maltz DA, Hu Y. "The dynamic source routing protocol for mobile ad hoc networks (DSR)", http://www.ietf.org/internet-drafts/draft-ietf-manet-dsr-10.txt, July 2004
4. Marina MK, Das SR. "Routing performance in the presence of unidirectional links in multihop wireless networks", in Proceedings of ACM MobiHoc, 2002.
5. The VINT Project, "The ns Manual (formely ns notes and documentation)", http://www.isi.edu/nsnam/ns/ns-documentation.html, Nov, 2011
6. S. Mueller, R. Tsang, D. Ghosal, "Multipath Routing in Mobile Ad Hoc Networks: Issues and Challenges", Lecture Notes in Computer Science (LNCS 2965), pp. 209-234, 2004.
7. Lee SJ, Gerla M. "AODV-BR: backup routing in ad hoc networks", in Proceedings of IEEE Wireless Communications and Networking Conference (WCNC), 2000
8. Valera A, Seah WKG, Rao SV. "Cooperative packet caching and shortest multipath routing in mobile ad hoc networks", in Proceedings of IEEE Infocom, 2003

9. Garcia-Luna-Aceves JJ, Mosko M, Perkins CE. “A new approach to on-demand loop-free routing in ad hoc networks”, in Proceedings of ACM PODC, 2003
10. Nasipuri A, Castaneda R, Das SR. “Performance of multipath routing for on-demand protocols in mobile ad hoc networks”, ACM/Kluwer Mobile Networks and Applications (MONET), vol. 6(4),pp. 339–349; 2001.
11. Ahmed S, Alam M.S, “ Performance evaluation of important ad-hoc network protocols”, Proceedings of EURASIP Journal on Wireless Communication and Networking vol. 6(2), pp. 42, Apr, 2006
12. Klaus Nieminen, “Introduction to Ad-hoc Networking”, http://www.citeseerx.edu/.
13. Humaira Ehsan and Zartash Afzal Uzmi, “Performance Comparison of ad hoc wireless network routing protocol”,Proceedings of INMIC 8th International, pp. 457-465, 24-26 Dec,2004
14. C. Siva Ram Murthy and B.S. Manoj, “Ad hoc Wireless Networks Architecture and protocols”, Prentice Hall, 2004
15. Francisco J. Ros, Pedro M. Ruiz, “Implementing a new MANET unicast routing protocol in NS2”, Dec 2004.
16. Kapang Lego, Pranav K.Singh, Dipankar Sutradhar, “Comparative Study of Ad hoc routing protocols AODV, DSR, DSDV in mobile ad hoc network”, International Journal of CSE vol. 1(4), pp.364-371.
17. Goff T, Abu-Ghazaleh N, Phatak D, Kahvecioglu R. “Preemptive routing in ad hoc networks”, in Proceedings of IEEE/ACM MobiCom, 2001.
18. Marina M.K, Das S.R, “Ad hoc on-demand multipath distance vector routing”, in Proceeding IEEE Wireless Communication and mobile computing, vol. 6, pp. 969-988,2006.
19. Tsirigos A, Haas Z. J, “Multipath routing in the presence of frequent topological changes”, IEEE Communications Magazine, vol. 39(11), pp. 132-138, Nov,2001.