ISSN 2319-524X

Survey of On Demand Routing Protocols for Mobile Ad Hoc Network

B P Patil
Professor, E&TC Department
Army Institute of Technology
Pune, Maharashtra, India
E-mail: bp_patil@rediffmail.com

Rahul Desai
Asst Professor, IT Department
Army Institute of Technology
Pune, Maharashtra, India
E-mail: desaimrahul@yahoo.com

Abstract- **Ad hoc network are wireless network with no infrastructure support. Due to the limited transmission range of wireless network interfaces, multiple network hops may be needed for one node to exchange data with another across the network. In such a network, each mobile node operates not only as a host but also as a router, forwarding packets for other mobile nodes in the network that may not be within the direct reach. Routing protocols developed for wired networks such as the wired Internet are inadequate here as they not only assume mostly fixed topology but also have high overheads. This has lead to several routing algorithms/proposals specifically targeted for ad hoc networks. While some of these proposals are optimized variants of protocols originally designed for wired networks, the rest adopt new paradigms such as proactive or on demand routing. This paper concentrates on various on Demands (or also known as reactive) Routing protocols such as DSR and AODV, their optimization and their comparison with Proactive Routing Protocols.**

Keywords-Ad Hoc Networks,MANET, DSR, AODVetc.

I. INTRODUCTION OF AD HOC NETWORKS

An ad Hoc network is a collection of wireless mobile nodes dynamically forming a temporary network without the use of existing network infrastructure or centralized administration. Due to the limited transmission range of wireless network interfaces, multiple network hops may be needed for one node to exchange data with another across the network. In such a network, each mobile node operates not only as a host but also as a router, forwarding packets for other mobile nodes in the network that may not be within the direct reach. Each node participates in an Ad Hoc routing protocol that allows it to discover multi hop paths through the network to any other node. The idea of an Ad Hoc network is sometimes also called as an Infrastructure less network, since the mobile hosts in the network dynamically establish routing among themselves to form their own network on the fly. Ad Hoc networks are typically set up for a limited period of time. The protocols are tuned to the particular applications (send a video stream across the battlefield; find out of a fire has started in the forest; establish a video conference among three teams engaged in a rescue effort). The application may be mobile and the environment may change dynamically.

Thus "mobile ad hoc network" (MANET) is an autonomous system of mobile routers (and associated hosts) connected by wireless links -- the union of which form an arbitrary graph. The routers are free to move randomly and organize themselves arbitrarily; thus, the network's wireless topology may change rapidly and unpredictably. Such a network may operate in a standalone fashion, or may be connected to the larger Internet. Due to a lack of infrastructure support, each node acts as a router, forwarding data packets for other nodes. Because of its mobile, non-infrastructure nature, the ad hoc network poses new design requirements; the first is self-configuration (of addresses and routing) in the face of mobility. At the application level, ad hoc network users typically communicate and collaborate as teams (for example, police, firefighters, medical personnel's teams in a search and rescue mission), These applications thus require efficient group communications (multicasting) for both data and real time traffic. Moreover, mobility stimulates a host of location based services nonexistent in the wired Internet.

This work is ordered as follows. We described the need and specialty of Ad Hoc Routing in section 2 and basic protocols used in wired Internet in section 3. Section 4 deals with on demand routing and section 5 with on-demand routing protocols that are used in ad hoc networks. Finally section 6 compares proactive protocols with on demand reactive protocols and also the various on-demand routing protocols and their performance.

II. SPECIALITY OF AD HOC NETWORK ROUTING

Routing consists of two fundamental steps; Forwarding packets to the next hop (from an input interface to an output interface in a traditional wired network) and Determining how to forward packets (building a routing table or specifying a route) [1]. Forwarding packets is easy, but knowing where to forward packets (efficiently) is hard as the packets should reach to the destination with minimum number of hops (path length), delay, and with almost zero packet loss and minimum cost. Thus routing is the mechanism used in communications to find a path between two entities. As an OSI layer, this mechanism receives data ready to send from the upper layer, then calculates the best path for the destination and forwards it to layer 2.

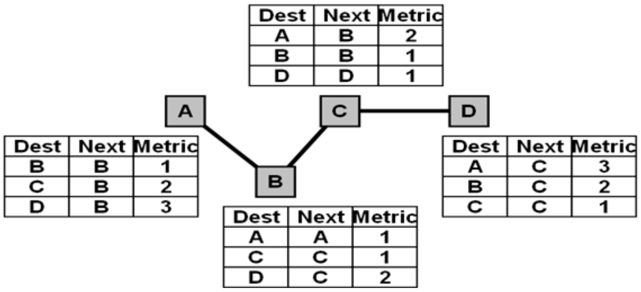
To judge the merit of a routing protocol, one needs metrics both qualitative and quantitative with which to measure its suitability and performance. These metrics should be independent of any given routing protocol. Qualitative properties include distributed operation, Loop freedom techniques, demand based or proactive operation, sleep period operation and Unidirectional link support. In order to perform quantitative analysis the various metrics such as Routing overhead, end to end delay, delay jitter, round trip time, number of data packets dropped, throughput, efficiency and most important path optimality are used.

Developing support for routing is one of the most significant challenges in ad hoc networks and is critical for the basic network operations. Certain unique combinations of characteristics make routing in ad hoc networks interesting. First, nodes in an ad hoc network are allowed to move in an uncontrolled manner. Such node mobility results in a highly dynamic network with rapid topological changes causing frequent route failures. A good routing protocol for the network environment has to dynamically adapt to the changing network topology. Second, the underlying wireless channel provides must lower and more variable bandwidth than wired network. The wireless channels working as a shared medium makes available bandwidth per node even lower. So routing protocols should be bandwidth efficient by expending a minimal overhead for comparing routes so that much of the reaming bandwidth is available for the actual data communication. Third, nodes run on batteries which have limited energy supply. In order for nodes to stay and communicate for longer periods, it is desirable that a routing protocol be energy efficient as well. Thus, routing protocols meet the conflicting goals of dynamic adaptation and low overhead to deliver good overall performance.

Thus, every node is potentially a router in a MANET, while most nodes in traditional wired networks do not route packets. Nodes transmit and receive their own packets and, also, forward packets for other nodes. Topologies are dynamic in MANETs due to mobile nodes, but are relatively static in traditional networks. Routing in MANETs must consider both Layer 3 and Layer 2 information, while traditional protocols rely on Layer 3 information only. Link layer information can indicate connectivity and interference. MANET topologies tend to have many more redundant links than traditional networks. A MANET router typically has a single interface, while a traditional router has an interface for each network to which it connects. Routed packet sent forward when transmitted, but also sent to previous transmitter. Channel properties, including capacity and error rates, are relatively static in traditional networks, but may vary in MANETs. Interference is an issue in MANETs, but not in traditional networks. Channels can be asymmetric with some Layer 2 technologies (as IEEE 802.11 MAC assumes symmetric channels). Power efficiency is an issue in MANETs, while it is normally not an issue in traditional networks. MANETs may have gateways to fixed network, but are typically "stub networks," while traditional networks can be stub networks or transit networks. There is limited physical security in a MANET compared to a traditional network. There are increased possibilities of eavesdropping, spoofing, and denial-of-security attacks in ad hoc networks. Traditional routing protocols for wired networks do not work well in most MANETs. MANETs are too dynamic. Wireless links present problems of interference, limited capacity, etc.

III. DISTANCE VECTOR VS. LINK STATE PROTOCOL

In Distance vector (DV) algorithms, "Distance" of each link in the network is a metric that is to be minimized. Each link may have "distance" 1 to minimize hop count. Algorithm attempts to minimize distance. The routing table at each node specifies the next hop for each destination and specifies the distance to that destination. Neighbors can exchange routing table information to find a route (or a better route) to a destination.



Dest	Next	Metric
A	B	2
B	B	1
D	D	1

Dest	Next	Metric
B	B	1
C	B	2
D	B	3

Dest	Next	Metric
A	C	3
B	C	2
C	C	1

Dest	Next	Metric
A	A	1
C	C	1
D	C	2

Figure 1. Example of Distance Vector Algorithm

In Link state algorithms, each node shares its link information so that all nodes can build a map of the full network topology. Link information is updated when a link changes state (goes up or down). Link state can be determined by sending small "hello" packets to neighbors. Given full topology information, a node can determine the next best hop or a route from the source.

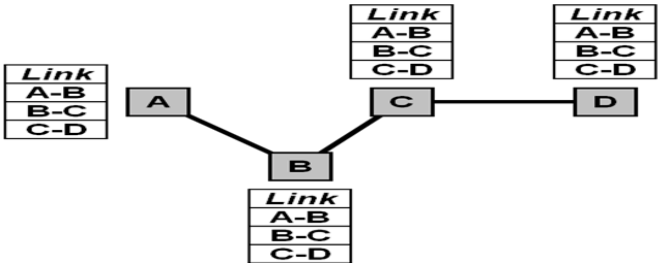


Figure 2. Example of Link State Algorithm

Routing protocols developed for wired networks such as the wired Internet are inadequate here as they not only assume mostly fixed topology but also have high overheads. This has lead to several routing proposals specifically targeted for ad hoc networks. While some of these proposals are optimized variants of protocols originally designed for wired networks, the rest adopt new paradigms such as on demand routing, where routes are maintained reactively only when needed. This is in contrast with the traditional, proactive Internet-based protocols.

Proactive protocols maintain unicast routes between all pairs of nodes regardless of whether all routes are actually used. Therefore, when the need arises the traffic source has a route readily available and does not have to incur any delay for route discovery. These protocols also can also find optimal routes (shortest path) given a model of link costs. Routing protocols on the Internet (distance vector based RIP and link state based OSPF) fall under this category. However, these protocols are not directly suitable for resource poor and mobile ad hoc networks because of their high overheads and/or somewhat poor convergence behavior. Therefore, several optimized variations of these protocols have been proposed for use in ad hoc networks. These protocols are broadly classified into the two traditional categories: distance vector and link state.

A different approach from table driven routing (proactive approach) is source initiated on demand routing. Main idea in on demand routing is to find and maintain only needed routes. Proactive routing protocols maintain all routes without regard to their ultimate use. The obvious advantage with discovering routes on demand is to avoid incurring the cost of maintaining routes that are not used. On demand or reactive Routing creates routes when desired by the source node. When a node requires a route to a destination, it initiates a route discovery process within the network. This process is completed once a route is found or all possible route permutations have been examined. Once a route has been established, it is maintained by a route maintenance procedure until either the destination becomes inaccessible along every path from the source or until the route is no longer designed. Reactive means discover route only when you need it. This saves energy and bandwidth during inactivity but congestion occurs during high activity. Significant delay might occur as a result of route discovery. It is good for light loads but collapse in large loads.

IV. ON DEMAND DSR AND AODV PROOTOCOLS

The Dynamic Source Routing Protocol for Multihop Wireless Ad Hoc Networks (DSR) [2, 3] is characterized by the use of source routing. That is, the sender knows the complete hop-by-hop route to the destination. These routes are stored in a route cache. The data packets carry the source route in the packet header. When a node in the ad hoc network attempts to send a data packet to a destination for which it does not already know the route, it uses route discovery process to dynamically determine such a route. Route discovery works by flooding the network with route request packets. Each node receiving a request rebroadcasts it, unless it is the destination or it has a route to the destination in its route cache. Such a node replies to the request with a route reply packet that is routed back to the original source. Route request-reply packets are also source routed. The request builds up the path traversed so far. The reply routes itself back; to the source by traversing this path forward. The route carried back by the reply packet is cached at the source for future use. Entries in route caches updated as nodes learn new routes. Packet carries complete ordered list of nodes, through which packet will pass. Sender checks its route cache, if route exists; sender constructs a source route in the packet's header. If route expires or does not exist, sender initiates the Route Discovery Mechanism.

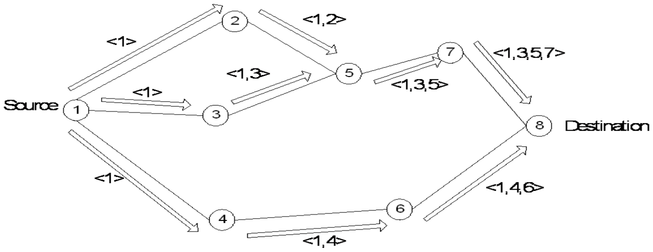


Figure 3. RREQ Request from Source to destination

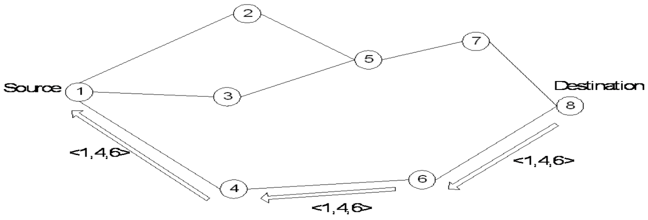


Figure 4. RREP Reply from destination to source

Each RREQ contains the source and destination addresses, a unique RREQ sequence ID and a list of all the nodes it traverses on its way to the destination. Nodes forward RREQs after appending their names. Destination node receives RREQ and unicast a RREP back to sender node. Each neighboring node checks if it has received the RREQ before, and discards it if it has. If it hasn't received it before it adds its address to the path recorded in the RREQ and forwards it to its neighbors. The source may also use the time to live (TTL) field in the IP header to limit the number of hops it is allowed to travel so that the RREQ will not be flooded uncontrollably through the network. When link changes, (a link is considered broken if the sending node does not receive an acknowledgement after sending a packet a certain number of times) existing source routes no longer works. Routing nodes respond to source routes with a Route Error (RERR), triggering a new route discovery. Routing nodes may attempt to change source route and re-forward on a cached route. For route Maintenance, Two types of packets used: Route Error Packet and Acknowledgement. If transmission error is detected at data link layer, Route Error Packet is generated and sends to the original sender of the packet. ACKs are used to verify the correction of the route links.

RFC 4728 suggests two methods of organizing the routing cache; a path cache organization or a link cache organization. In the former, routing information is listed by destination address, along with the corresponding path or paths to the destination. In link cache, the node breaks up any paths it knows of into links, and uses these links to establish a graph which reflects the topology of the network as seen by the node. To determine paths from the links stored in the routing cache, an algorithm such as Djikstra's algorithm has to be used to determine an optimal path to the destination. Clearly, the path cache approach is simpler to implement and use, while the link cache approach is more complex, and requires more processing and resources. However, the link cache approach may be more efficient in the sense that it allows the selection of the 'best' paths through the network [2]. DSR does not require cache entries to expire, and so they may remain in the cache for a long time. Requiring cache entries to expire prevents the use of stale routes, and reduces caching capacity required. Since nodes store routing information in routing caches rather than routing tables, it is possible to store more than one route per source and destination, i.e. DSR supports multipath operation, in which case any method or metric can be used to choose from amongst different routes available to a destination, for instance least number of hops [2].

DSR makes aggressive use of source routing and route caching. With source routing, complete path information is available and routing loops can be easily detected and eliminated without requiring any special mechanism. Because route requests and replies are source routed, the source and destination, in addition to learning routes to each other, can also learn and cache routes to all intermediate nodes. Also, any forwarding node caches any source route in a packet it forwards for possible future use. DSR employs several optimizations including promiscuous listening which allows nodes that are not participating in forwarding to overhear on-going data transmissions nearby to learn different routes free of cost. To take full advantage of route caching, DSR replies to all requests reaching a destination from a single request cycle. Thus the source learns many alternate routes to the destination, which will be useful in the case that the primary or shortest route fails. Having access to many alternate routes saves route discovery floods, which is often a performance bottleneck. This may however, result in route reply flood unless care is taken. However, aggressive use of route caching comes with a penalty. Basic DSR protocol lacks effective mechanisms to purge stale routes. Use of stale routes not only wastes precious network bandwidth for packets that are eventually dropped, but also causes cache pollution at other nodes when they forward/overhear stale routes. Several performance studies [4] have shown that stale caches can significantly hurt performance especially at high mobility and/or high loads. These results have motivated subsequent work on improved caching strategies for DSR [5, 6]. Besides stale cache problems, the use of source routes in data packets increases the byte overhead of DSR. This limitation was addresses in a later work by the DSR designers [6]. Ad Hoc on Demand Distance Vector Routing (AODV) [7] is pure on-demand routing protocol. AODV uses traditional routing tables, one entry per destination. This is in contrast to DSR, which can maintain multiple route cache entries for each destination. Without source routing, AODV relies on routing table entries to propagate a RREP back to the source and, subsequently, to route data packets to the destinations. AODV uses destination sequence numbers as in DSDV [8] to prevent routing loops and to determine freshness of routing information. These sequence numbers are carried by all routing packets. The absence of source routing and promiscuous listening allows AODV to gather only a very limited amount of routing information with each route discovery. Besides, AODV is conservative in dealing with stale routes. It uses the sequence numbers to infer the freshness of routing information and nodes maintain only the route information for a destination corresponding to the latest known sequence number; routes with older sequence numbers are discarded even though they may still be valid. AODV also uses a time based route expiry mechanism to promptly purge stales routes. Again if a low value is chosen for the timeout, valid routes may be needlessly discarded.

In AODV, each node maintains at most one route per destination and as a result, the destination replies only once to the first arriving request during a route discovery. Being a single path protocol, it had to invoke a new route discovery whenever the only path from the source to the destination fails. When topology changes frequently, route discovery needs to be initiated often which can be very inefficient since route discovery flood is associated with significant latency and overhead. To overcome this limitation, another Multipath extension to AODV called Ad Hoc On-Demand Multipath Distance Vector (AOMDV) [9, 10] is used. AOMDV discovers multiple paths between source and destination in a single route discovery. As a result, a new route discovery is necessary only when each of the multiple paths fail. Local HELLO messages are used to determine local connectivity. It can reduce response time to routing requests and can trigger updates when necessary. Sequence numbers are assigned to routes and routing table entries. They are used to supersede stale cached routing entries. Every node maintains two counters - Node sequence number and Broadcast ID. AODV Route Request (RREQ) initiated when a node wants to communicate with another node, but does not have a route to that node. Source node broadcasts a route request (RREQ) packet to its neighbors. Sequence numbers are used in headers where Source sequence indicates "freshness" of reverse route to the source while destination sequence number indicates freshness of route to the destination. Every neighbor receives the RREQ and either returns a route reply (RREP) packet, or forwards the RREQ to its neighbors. (source_addr, broadcast_id) uniquely identifies the RREQ. Broadcast_id is incremented for every RREQ packet sent. Receivers can identify and discard duplicate RREQ packets. If a node cannot respond to the RREQ; the node increments the hop counts. The node saves information such as Neighbor that sent this RREQ packet, destination IP address, source IP address, broadcast ID, source node's sequence number and expiration time for reverse path entry (to enable garbage collection) etc to implement a reverse path set up (AODV assumes symmetrical links). If a node receives an RREQ packet and it has a current route to the target destination, then it unicast a route reply packet (RREP) to the neighbor that sent the RREQ packet.In AODV, if any link along the established route breaks the upstream node has to send a RERR to the source node. The RERR message contains the IP address of the link on the other side of the broken link. An advantage of AODV is that the upstream node (the node that failed to send data over a link towards the destination) also forwards the RERR to any other nodes it thinks are using the broken link (i.e. the link's precursors). These nodes in turn update their routing tables, setting the hop count to the destination to infinity and forward the RERR to any other nodes using the broken link, if there are any. This way, concerned nodes know very quickly when a link breaks. However, an entry for a broken link is not immediately removed from routing tables, as it often contains useful information. After receiving the RERR, the source can send a new RREQ to find a new route if it still has data to send [11]. The other situation where a node generates a RERR is when it receives a data packet destined for a node it does not have a routing table entry for. In this case, the RERR contains the IP address of the destination, and it is sent to the previous hop, i.e. the node that the data packet was received from. AODV nodes send periodic "Hello" messages to their neighbors (TTL field is set to 1). These are used to confirm that neighbors a node is aware of are still within range, and to know if any new nodes have moved to the vicinity recently. Not all nearby nodes have to send "Hello" messages; these messages are not required if the node has sent any data packets within the past "Hello Interval", which is by default 1 second. A Hello message is a special RREP message, unprompted by an RREQ that contains the sending node's IP address and sequence number [11].

COMPARISON

Aggregate throughout and end-to-end delays are key measures of interest when assessing protocol performance. Throughout is directly related to the packet drops. Packet drops typically happens because of network congestion or for lack of route. Since most dynamic protocols (proactive or reactive) try to keep the latter type (no route) of drops low by being responsive to topology changes, network congestion drops become the dominant factor when judging relative throughput performance. For the same data traffic load, routing protocol efficiency (in terms of control overhead in bytes or packets) determines the relative level of network congestion because both routing control packets and data packets share the same bandwidth and buffers. End-to-end delay of a packet depends on route discovery latency, additional delays at each hop (comprising of queuing, channel access and transmission delays), and the number of hops. At low or moderate loads, queuing and channel access delays do not contribute much to the overall delay. In this regime, proactive protocols are likely to have better delay performance. However, at high loads, queuing and channel access delays become significant enough to exceed route discovery latency. So like in the case of throughput, routing protocol overhead again becomes key factor in determining relative delay performance.

On-demand routing opposed to proactive routing is naturally adaptive to traffic diversity and therefore its overhead proportionately increases with increase in traffic diversity. On the other hand, for proactive routing overhead is independent of the traffic diversity. So when the traffic diversity is low, on demand routing is relatively very efficient in terms of the control overhead regardless of relative node mobility. When the majority of traffic is destined to only few nodes, a proactive protocol maintaining routes to every possible destination incurs a lot of unnecessary overhead. Mobility does not alter this advantage of on demand routing. This is because an on demand protocol reacts only to link failures that break a currently used path, whereas proactive protocol reacts to every link failure without regard to whether the link is on a used path. On demand routing can also significantly benefit by caching multiple paths when node mobility is low.

With high traffic diversity, the routing overhead for on demand routing could approach that of proactive routing. The overhead alone is not the whole picture. Path optimality also plays a role in determining the overall overhead – using a suboptimal path results in excess transmissions which contribute to overhead. Using suboptimal routes also increases the end-to-end delay. Pure proactive protocols aim to always provide shortest paths whereas in pure on demand protocols, a path is used until it becomes invalid even through the path may become suboptimal due to node mobility. The issue of path sub-optimality becomes more significant at low node mobility because each path is usable for a longer period. Thus, accounting suboptimal path overhead increases the total overhead with on demand approach. According to various studies, it has found that DSR with the help of caching is more effective at low mobility and low loads. AODV performs well in more stressful scenarios of high mobility and high loads. These relative performance differentials are attributed to DSR's lack of effective mechanisms to purge stale routes and AODV's need for resorting to route discovery often because of its single path nature. However, DSR with improved caching strategies, and AODV with the ability to maintain multiple paths are expected to have similar performance.

In DSR, aggressive caching and multipath support enables nodes to know more alternative routes which speeds up recovery from link breaks and reduces chances of having to drop packets when buffers are full. As DSR lacks a clear policy of expiring outdated cache entries, cached data may be based on stale routes, leading to transmission errors, unsuccessful retransmission attempts followed by new route requests. Such delays increase the probability of packets having to be dropped due to limited buffer capacities. In DSR, aggressive caching makes it more likely for source node to find a route in its cache to the destination without initiating an RREQ. In AODV, control packets are lower. In DSR, Aggressive caching approach helps reduce the number of route requests required which speeds up route set up. However, as DSR lacks a clear policy of expiring outdated cache entries, cached data may be based on stale routes, leading to transmission errors, unsuccessful retransmission attempts followed by new route requests. In AODV, Sequence numbers, explicit routing table entry timeouts prevent use of stale routing data and the associated delays. Hello messages keep routing tables up to date.

CONCLUSION

Various simulation results performed on the analysis of various proactive and on-demand routing protocols shows, for low to moderate loads, proactive protocols works well as compared with high loads. End-to-end delay is minimum in proactive protocols as compared with on demand routing policy. On demand Routing protocols are more effective in high traffic diversity as well as high mobility. Average end to end delay, the performance of DSR and AODV are almost uniform. In terms of Packet Delivery Traction (PDF), DSR performs well when the number of nodes is less as the nodes increase performance declines. The performance of AODV is consistently uniform. PDF changes rapidly when number of nodes increases. In terms of throughput, DSR remains consistent. AODV toggle with respect to increase in number of nodes. In terms of Normalized Routing Load, AODV performs well even the nodes are increased in comparison with DSDV and DSR.

REFERENCES

[1] K. Sivakumar, "Mobile Ad-Hoc Network Routing Principle and Protocol Evaluation Metrics" ISSN: 0976-8491(Online) | ISSN: 2229-4333(Print) IJCST Vol. 2, Issue 4, Oct . - Dec. **2011**

[2] Haseeb Zafar, Nancy Alhamahmy, David Harle and Ivan Andonovic "Survey of Reactive and Hybrid Routing Protocols for Mobile Ad Hoc Networks" International Journal of Communication Networks and Information Security (IJCNIS) Vol. 3, No. 3, December 2011

[3] D B Johnson, D A Maltz, Y. Hu and J G Jetcheva. The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR) http://www.ietf.org/internet-drafts/draft-ietf-manet-dsr-07.txt , Feb 2002, IETF Internet Draft.

[4] C E Perkins, E M Royer, S R Das and M K Marina. Performance Comparison of two On Demand Routing Protocols for ad hoc networks. IEEE Personal Communications, 8(1):16-28, 2001

[5] Y-C Hu and D. Johnson. Caching strategies in On Demand Routing Protocols for wireless Ad Hoc Networks. In Proceedings of IEEEE/ACM MobiCom, pages 231-242, 2000

[6] Y-C Hu and D. Johnson. Implicit Source Routes for on Demand Ad Hoc Network Routing. In Proceedings of ACM MOBIHOC, pages 1-12, 2001

[7] Z. Haas, J.Halpern, and L. Li. Gosssip-based Ad Hoc Routing. In Proceedings of IEEE infoCom, pages 1707-1716, 2002

[8] E. Kulla, M. Hiyama, M. Ikeda, L. Barolli, V. Kolici, R. Miho, "MANET performance for source and destination moving scenarios considering OLSR and AODV protocols", *Mobile Information Systems*, vol. 6, no. 4, **2010**, pp. 325–339.

[9] C. Wu, F.Zhang and H.Yang, "A Novel QoS Multipath Path Routing in MANET," JDCTA: International Journal of Digital Content Technology and its Applications, Vol.4, No. 3, pp. 132 - 136, **2010**.

[10] Mr. Rajneesh Gujral, Dr. Anil Kapil "Secure QoS Enabled On-Demand Link-State Multipath Routing in MANETS" Proceeding of BAIP 2010, pp. 250-257 SPRINGER LNCSCCIS, Trivandrum, Kerala, India, March 26-27, 2010

[11] A. Quayyum, L Viewnnot and A. Laouiti, Multipoint Relaying for flooding broadcast Messages in Mobile Wireless Networks. In proceedings of the 35th Annual Hawaii International Conference on System sciences, pages 3898-3907, 2002.