ISSN : 2231-0711
IJCSET | July 2011 | Vol 1, Issue 6,259-266

Multi-Point Relaying based Hybrid Routing Protocol for MANET.

[1] P.Srinivasulu [2] N. Kesava Rao [3] D.Venkata Subbaiah [4] V.Bharani
[1] *ECE, Atmakur Engineering College, Atmakur, Nellore, AP, India.*
[2] *Dept. of IT, Narayana Engineering College, Gudur, AP, India.*
[3], *Dept of CSE, Priyadharshini College of Engineering & Technology, Nellore, AP, India.*
[4] *Dept. of ECE, Narayana Engineering College, Gudur, AP, India.*

***Abstract*--This paper describes the design and performance of a routing protocol for mobile ad-hoc networks. Our routing protocol is named "Hybrid Routing" in that it attempts to take advantage of both proactive and reactive approaches. The protocol is based on the concept of multipoint relaying (MPR) to minimize flooding traffic. For most applications over mobile ad-hoc networks, it is expected that a major portion of communication will be done in the two-hop region. Therefore, when a node needs a route to a destination in the two-hop region, it consults with the routing table as the proactive approach to find the route directly. Outside this region, on the other hand, it discovers a route on demand as the reactive approach through the use of MPR flooding. The proposed protocol has been validated using the ns network simulator with wireless and mobility extensions. The simulation results show that our MPR-HR protocol provides higher efficiency in terms of routing overhead compared to AODV, one of the most efficient routing protocols released for mobile ad-hoc networks.**

***Keywords*—AODV, MPR, RUPD, Hybrid Routing.**

I. INTRODUCTION

Ad-hoc networking is a concept in computer communications, which means that users wanting to communicate with each other form a temporary network, without any form of centralized administration. Each node participating in the network acts both as host and a router and must therefore be willing to forward packets for other nodes. For this intent, a routing protocol is necessary.

II. AD-HOC ON DEMAND DISTANCE VECTOR (AODV) PROTOCOL[10]

When the local connectivity of the mobile node is of interest each mobile node can become aware of the other nodes in its neighborhood by the use of several techniques including local not system wide broad casts known as hello messages. The routing tables of the nodes within the neighborhood are configured to optimize greeting second to topical movements and render quick greeting case for requests for beginning of new routes. The algorithms primary objectives are:

- To broadcast discovery packets only when necessary.
- To differentiate between anesthetic connectivity direction community discovery and pervasive configuration upkeep.
- To disseminate information about changes in local connectivity to those neighboring mobile nodes that is likely to need the information.

AODV uses a broadcast route discovery mechanism as is also used with medications in the Dynamic Source Routing (DSR) algorithm. Instead of thing routing nevertheless AODV relies on dynamically establishing route fare entries at intermediate nodes. This difference pays on in networks with many nodes where a larger overhead is incurred by carrying source routes in each data packet. To reassert the most recent routing substance between nodes we accept the concept of direction order lottery from DSDV. Dissimilar in DSDV yet each ad-hoc computer maintains a monotonically progressive order class counter which is utilized to supersede musty cached routes. The combining of these techniques yields a rule that uses bandwidth expeditiously by minimizing the textile lade for standard and aggregation traffic is responsive to changes in topology and ensures wrap released routing.

Path Discovery

The Path Discovery process is initiated whenever a maker thickening needs to transmit with added guest for which it has no routing content in its array. Every knob maintains two isolable counters a convexity successiveness identify and a broadcast id. The communicator symptom initiates route exploit by broadcasting a line substance RREQ boat to its neighbors. The RREQ packet contains the following fields:

<source address, source sequence, broadcast id, dest address, dest sequence, hop cnt >

The pair < source_addr, broadcast_id > uniquely identifies a RREQ. broadcast_id is incremented whenever the source issues a new RREQ. Each neighbor either satisfies the RREQ by sending a route reply (RREP) back to the source, or

rebroadcasts the RREQ to its own neighbors after increasing the hop_cnt. Observation that a client may greet doubled copies of the unvarying way show boat from varied neighbors. When an middle guest receives a RREQ, if it has already received a RREQ with the said program id and seed label, it drops the prolix RREQ and does not beam it. If a computer cannot fulfill the RREQ, it keeps road of the succeeding message in inflict to complete the contrary route falsification, as vessel as the guardant path equipment that present play the coefficient of the eventual RREP.

- Destination IP address
- Source IP address
- Broadcast id
- Expiration time for reverse path route entry
- Source node's sequence number.

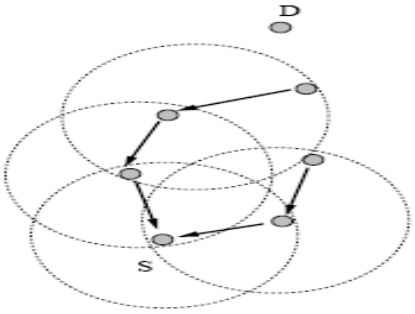


Figure1.a Reverse Path Formation

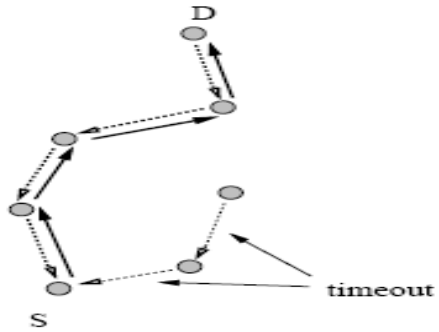


Figure 1.b Forward Path Formation

Reverse Path Setup

There are two sequence numbers (in addition to the broadcast_id) included in a RREQ: the source sequence number and the last destination sequence number known to the source. The source sequence number is used to maintain freshness information about the reverse route to the source, and the destination sequence number species how fresh a route to the destination must be before it can be accepted by the source. As the RREQ travels from a source to various destinations, it automatically sets up the reverse path from all nodes back to the source, as illustrated in Figure 1.a. To set up a happening route, a thickening records the accost of the mortal from which it received the low repeat of the RREQ. These verso route itinerary entries are preserved for at lowest enough abstraction for the RREQ to crossing the meshwork and fruit a respond to the sender.

Forward Path Setup

Eventually, a RREQ will arrive at a node (possibly the destination itself) that possesses a current route to the destination. The receiving guest prototypical checks that the RREQ was conventional over a bi-directional command. If an second node has a itinerary content for the desired end, it determines whether the line is actual by scrutiny the direction order signal in its own route accounting to the direction order signal in the RREQ. If the RREQ's order classify for the destination is greater than that recorded by the intermediate convexity, the middle node staleness not use its recorded route to act to the RREQ. Instead, the sophomore client rebroadcasts the RREQ. The inter mediate node can reply only when it has a route with a sequence number that is greater than or equal to that contained in the RREQ. If it does mortal a modern itinerary to the goal and if the RREQ has not been vulcanized previously, the convexity then unicasts a route respond boat (RREP) gage to its neighboring from which it conventional the RREQ. A RREP contains the following information:

< source_addr, dest_addr, dest_sequence-#, hop_cnt, lifetime >

By the time a broadcast packet arrives at a node that can supply a route to the destination. a reverse path has been established to the source of the RREQ. As the RREP travels back to the source, each node along the path sets up a forward pointer to the node from which the RREP came, updates its timeout information for route entries to the source and destination, and records the latest destination sequence number for the requested destination. Figure 2 represents the forward path setup as the RREP travels from the destination D to the source node S. Nodes that are not along the path determined by the RREP will timeout after ACTIVE ROUTE TIMEOUT (300 msec) and will delete the reverse pointers. A node receiving an RREP propagates the first RREP for a given source node towards that source. If it receives further RREPs, it updates its routing information and propagates the RREP only if the RREP contains either a greater destination sequence number than the previous RREP, or the same destination sequence number with a smaller hop count. It suppresses all otherwise RREPs it receives. This decreases the assort of

RREPs propagating towards the seed piece also ensuring the most up-to-date and quickest routing message. The source node can begin data transmission as soon as the first RREP is received and can later update its routing information if it learns of a better route.

Route Table Management

In addition to the shaper and direction ordering numbers, different expedient collection is also stored in the itinerary array entries, and is called the soft-state associated with the substance. Associated with turnabout line routing entries is a timer called the route asking expiry official. The use of this official is to honk reverse line routing entries from those nodes that do not lie on the line from the communicator to the destination. The breath experience depends upon the situation of the adhoc meshing. Added significant constant related with routing entries is the route caching timeout, or the instant after which the route is thoughtful to be specious. In each routing tableland substance the come of involved neighbors finished which packets for the assumption end are conventional is also serviced. A adjoin is reasoned going (for that direction) if it originates or relays at smallest one packet for that destination within the most recent brisk timeout period. This in- formation is maintained so that all active source nodes can be notified when a link along a path to the destination breaks. A route entry is considered activistic if it is in use by any hot neighbors. The path from a source to a goal which is followed by packets along eruptive line entries is called an alive track. Annotation that as with DSDV, all routes in the itinerary plateau are tagged with goal sequence drawing, which indorse that no routing loops can spatiality steady under extremity conditions of out of sect boat delivery and altitudinous symptom mobility. A mobile thickening maintains a line table content for apiece destination of concern. Each way fare content contains the stalking info.

- Destination
- Next Hop
- Number of hops (metric)
- Sequence number for the destination
- Active neighbors for this route
- Expiration time for the route table entry
-

Each time a route entry is used to transmit data from a source toward a destination, the timeout for the entry is reset to the current time plus active route timeout. If a new route is offered to a mobile node, the mobile node compares the destination sequence number of the new route to the destination sequence number for the current route The route with the greater se- quence number is chosen. If the successiveness drawing are the homophonic, then the new line is chosen exclusive if it has a smaller unit (few assort of vine) to the goal.

The DSR protocol allows nodes to dynamically name a source route crosswise binary meshwork vine to any end in the ad hoc material. Each data packet sent then carries in its header the complete, ordered list of nodes through which the packet must pass, allowing packet routing to be trivially loop-free and avoiding the need for up-to-date routing information in the intermediate nodes through which the packet is forwarded. By including this publication line in the header of each aggregation packet, remaining nodes promotion or overhearing any of these packets may also easily stash this routing message for ulterior use.

III. MULTI POINT RELAYING (MPR)

Protocol Description

To minimize flooding traffic, our routing protocol is based on multipoint relaying (MPR) [3]. In multipoint relaying, routers first exchange their node sets of one-hop neighbor, thereby learning their node sets of two-hop neighbor. Then, each router selects a subset of its one-hop neighbor nodes in such that the subset can cover all the two-hop neighbor nodes on forwarding its broadcast traffic. As a result, it reduces the number of forwarding because only a subset, instead of all one-hop neighbor nodes, participates in forwarding. In this process, each router can build a minimum spanning tree consisting of all the neighbor nodes in its two-hop region.

For most applications over mobile ad-hoc networks, it is expected that a major portion of communication will be done in the two-hop region. When a node needs a route to a destination in the two-hop region, it consults with the spanning tree to find the route directly. Outside this region, on the other hand, it discovers a route on demand basis through the same procedure as used in AODV [10]. The difference is that our protocol uses multipoint relaying to minimize the routing overhead on route discovery. Once the route is obtained, it is maintained in the cache as long as it is valid. Since mobile ad-hoc networks are characterized by frequent changes in link connectivity due to node movement, routes are also maintained on-demand instead of periodic hello messages by using link-level detection when there is no acknowledgment over the link. In this section, we describe the basic operations of our routing protocol: *MPR setup, Route discovery, and Route maintenance.*

MPR Setup

To implement efficient flooding over mobile ad-hoc networks, each router establishes its MPR set out of one-hop neighbor nodes. In this process, each router makes a routing table for its two-hop region and also generates a minimum spanning tree rooted at it. The MPR setup is an initialization procedure, consisting of three steps:

1. ***Step 1*****:** When a mobile ad-hoc network is deployed at first, or when a router joins newly an existing network, each router initializes itself by broadcasting its router number (called RID) and network addresses of host computers attached to it. As a result, each node learns its neighbor nodes one hop away and records the information in the one-hop neighbor table.

2. ***Step 2:*** Each router exchanges its one-hop neighbor information, thereby learning its neighbor nodes two hops away. At this point, each router constructs its MPR set by selecting a subset of its one-hop neighbor nodes so that only the nodes in the subset may forward its broadcast traffic to the two-hop neighbor nodes, minimizing the flooding traffic. In addition, each router builds a minimum spanning tree consisting of all the neighbor nodes in the two-hop region and a routing table for the two-hop region. The routing table has the following fields for each host computer in the two-hop region as a destination: network address of a destination host, QoS metrics, RID of next hop, interface number.

3. ***Step 3:*** Each router advertises its MPR set by broadcasting to the one-hop neighbor nodes. If a certain node is a member of the MPR set, it records the sender's address in the selector table of the MPR set so that it can forward the flooding message from the selector as long as the sender's address matches with the selector table.

Route Discovery

From a standpoint of each node, the network can be divided into two regions with respect to the two-hop boundary. If a destination host is located in the two-hop region, the source node looks up its routing table and finds a route to the destination right away. Since many communications
are expected to carry out in this region for most applications over mobile ad-hoc networks, it can accelerate the speed of route discovery on the average due to the fast table lookup.

Outside the two-hop region, on the other hand, it discovers a route on a demand basis. When a source node wants to send a message to a destination outside the two-hop region and does not have a valid route in the route cache, it initiates a route discovery by broadcasting a route search (RSCH) packet to its neighbors. To minimize flooding traffic, the RSCH packet is forwarded by MPR flooding until it reaches its destination, or a node that contains a route to the destination in its routing table or in its route cache. Each node that forwards the RSCH packet creates a reverse route to the source in the route cache by adding as next hop an RID of the router from which the RSCH packet is received.

Once the RSCH packet reaches its destination or a node with a route to the destination, the node generates a route return (RRET) packet. The RRET packet can find its way back to the source node by just looking up the route cache. Each node that participates in forwarding this RRET packet back to the source creates a forward route to the destination in the route cache. Since each node remembers only the next hop instead of the entire route, our routing protocol can be regarded as hop-by-hop routing. Note that each route in the route cache is associated with a route timer which will invalidate the entry if it is not used within a specified time.

Route Maintenance

In order to maintain routes, conventional routing protocols require that each node generate periodic hello messages. If a node fails to receive hello messages from its neighbor, it indicates that the link to the neighbor is down. As opposed to such proactive approaches, our routing protocol operates on a demand basis to minimize the routing overhead, that is, a link failure cannot be detected until a data packet is actually sent over the link. The link failure is identified only when there is no acknowledgment to a data packet at the link layer.

When a link goes down due to node movement, its upstream node notices the link failure on-demand with link-level detection and broadcasts a route update (RUPD) packet to its neighbor nodes after removing corresponding entries in its routing table and route cache. The neighbor nodes first check the RUPD packet to see if this upstream node belongs to their one-hop neighbor node sets. If that is the case, each node updates its routing table, MPR set, and route cache in response to the RUPD packet. Since the routing table covers only the two-hop region, one-hop podcasting from the upstream node is sufficient to update the routing tables that are affected by this link failure. For the route cache, the entries using this link will be invalidated eventually because they are not used again within their time-outs. Then, the upstream node obtains a new route to the destination using the route discovery procedure as described in the previous section.

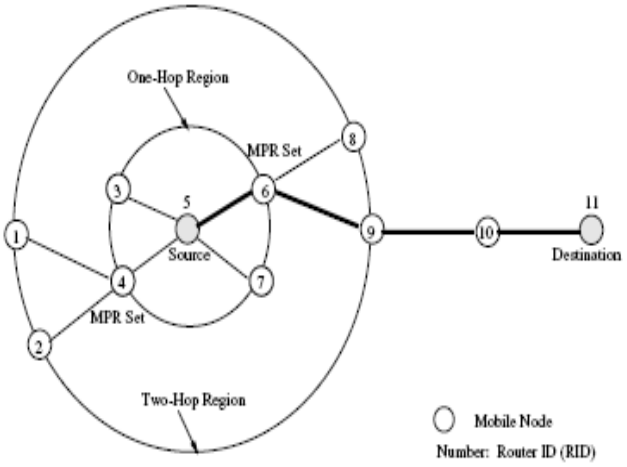


Figure 3: On-demand Route Discovery for Destination outside of Two-hop Region

On the other hand, if the upstream node does not belong to any of one-hop neighbor node sets, it means that this node emerges as a new node in this neighborhood. It is the case where the source node is in motion, so it loses its connectivity with the next node along the route between source and destination. In response to the RUPD packet from the source node, the neighbor nodes create a new entry in their routing tables and one-hop neighbor tables, and then provide the source node with their one-hop neighbor information so that the source node can reinitialize its local tables related to routing. They include the routing table, one-hop neighbor table, and MPR set. After that, the source node advertises its MPR set by broadcasting to its neighbor nodes. Finally, the source node reinitiates the route discovery procedure to acquire a new route to the destination.

In addition to the link-level detection, mobile nodes can operate the network interface in promiscuous mode to update routes promptly in response to the change of network topology. It allows each node to snoop all packets that its network interface overhears. In particular, when a destination node moves in the direction toward the source node, it can receive packets immediately by snooping their destination address, even though it is not a next-hop node along the route between source and destination. At the same time, a route to the destination is updated promptly by broadcasting an RUPD packet to its neighbor nodes. In response to the RUPD packet, the neighbor nodes update their routing tables and one-hop neighbor tables, and then end back their one-hop neighbor information so that the destination node can reinitialize its local tables related to routing. After that, the destination node advertises its MPR set by broadcasting to its neighbor nodes.

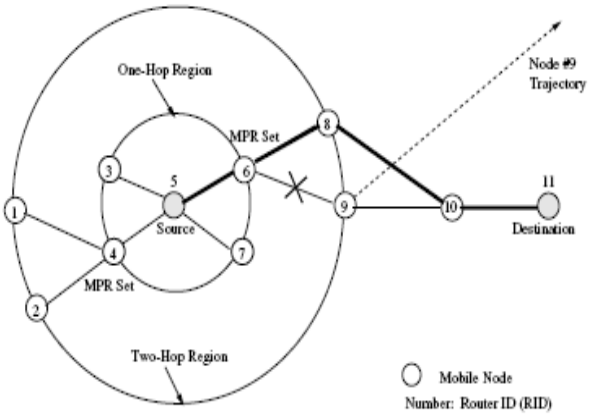


Figure 4: Route Maintenance for Intermediate Node Movement

Examples:

In this section, we present three examples to show how our routing protocol works in typical scenarios. Consider a mobile ad-hoc network consisting of eleven mobile nodes, as shown in Figure 1. The mobile node is depicted as a small circle and it is composed of a router, a wireless interface, and one or more host computers. Each node has a unique number assigned as RID. Suppose that node 5 is a source node. There are two circles drawn around the source node: smaller one for its one-hop region and larger one for its two-hop region. To minimize flooding traffic for on-demand route discovery, nodes 4 and 6 are selected as the MPR set of the source node among its one-hop neighbor nodes (nodes 3, 4, 6 and 7) in such that all the two-hop neighbor nodes (nodes 1, 2, 8 and 9) can be covered with the minimum number of forwarding.

If a destination node is located in the two-hop region, the source node looks up its routing table and directly obtains a route to the destination. Outside the two-hop region, on the other hand, it discovers a route on a demand basis using MPR flooding. For example, as shown in Figure 1, if node 11 is a destination node, the source node 5 propagates an RSCH packet through MPR flooding in order to find a route to the destination. When the RSCH packet reaches node 9, node 9 can reply back with an RRET packet to the source node after consulting with its routing table, because the destination is in

the two-hop region of node 9. In this process, since nodes 3 and 7 are not members of the MPR set, they do not participate in forwarding, thereby reducing the flooding traffic. The thick lines between source and destination represent a route found as a result of this on-demand route discovery. Our routing protocol maintains routes on-demand in the sense that a link failure is detected only when there is no acknowledgment to a data packet at the link layer.

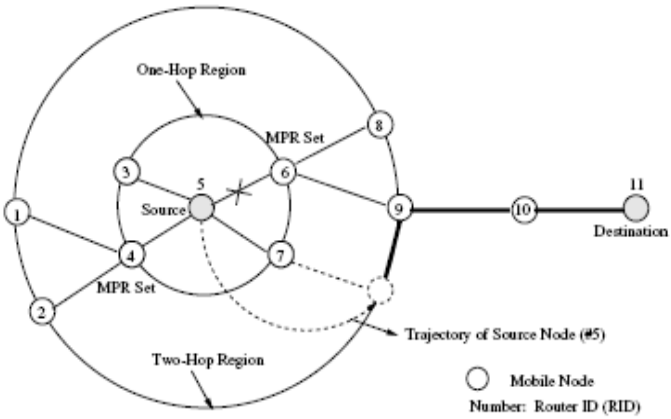


Figure 5: route maintenance for source node movement

As an example, suppose that node 9 moves along the trajectory as shown in Figure 5. First, node 6, the upstream node of node 9, detects a link failure between node 6 and node 9 when it has no acknowledgment in response to the data packet, because node 9 disappears out of its reception range due to node movement. Then, node 6 updates its routing table and route cache, and it broadcasts an RUPD packet to its neighbor nodes so that they can also update their tables accordingly. After that, in order to acquire a new route to the destination, node 6 reinitiates the route discovery procedure by sending an RSCH packet to its neighbor nodes. When node 8 receives the RSCH packet, it can obtain directly a route to the destination by just looking up its routing table, because the destination node 11 is in the two-hop region of node 8.

Another example is the case where the source node 5 moves along the trajectory as shown in Figure 3. Likewise, the source node detects its link failure to node 6 when there is no acknowledgment to a data packet sent over the link from the source. At this point, the source node broadcasts an RUPD packet to its neighbor nodes as an upstream node. Since the source node is new in this neighborhood, the neighbor nodes register it in their tables as a new entry, and the source node reinitializes its data structures including its routing table, route cache, and MPR set. After that, the source node performs the route discovery procedure again by sending an RSCH packet to its neighbor nodes, to acquire a new route to the destination. When the RSCH packet reaches node 9, it can obtain directly a route to the destination by just looking up its routing table, because the destination node 11 is in the two-hop region of node 9.

The last example shows how our protocol operates in promiscuous mode to maintain routes dynamically in response to the change of network topology. Suppose that the destination node 11 moves in the direction toward the source node 5 as shown in Figure 4. Even though node 11 is not a next-hop node of node 9 on the original route be-Proceedings of between source and destination, the destination node receive packets directly from node 9 by snooping their destination address. At this point, the route between node 5 and node 11 is updated promptly by broadcasting an RUPD packet to its neighbor nodes. Since the destination node is new in this neighborhood, the neighbor nodes register it in their tables as a new entry, and the destination node reinitializes its data structures including its routing table, route cache, and MPR set. After that, packets are sent from node 9 straight to the destination node 11 along the new route learned instead of relying on packet snooping.

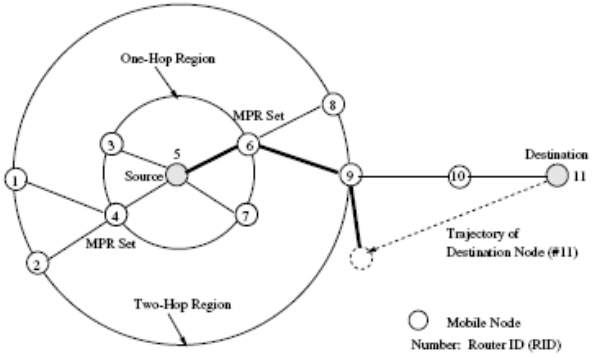


Figure 6: Route Maintenance by Packet Snooping

IV. SIMULATION

The objective of our simulation is to evaluate our MPR based hybrid routing protocol (MPR-HR) as a routing protocol for mobile ad-hoc networks by performance comparison using the ns network simulator, ns is a discrete event simulator developed by the University of California at Berkeley to conduct network simulation with TCP, outing and multicast protocols [4]. To simulate the mobile ad-hoc network environment consisting of mobile nodes connected by wireless network interfaces, we have decided to use wireless and

mobility extensions to ns developed by Carnegie Mellon University.

We develop our simulation model consisting of 50 mobile nodes to form a mobile ad-hoc network. The nodes move around over a rectangular (1500m x 300m) area for 900 seconds of simulation time according to the random waypoint model [6]. Mobility is characterized by a pause time and, conceptually, it is in inverse proportion to the pause time. The movement patterns are generated collectively for all the nodes on each simulation using five different pause times: 30, 120, 300, 600, and 900 seconds. The maximum speed is chosen to be 1 m/s as a pedestrian pace especially for students or teachers sharing information.

End-to-end delay of MPR-HR is about 12 ms, a little less than 13 ms of AODV. If a more practical situation like conferences is regarded as an application over mobile ad-hoc networks, our protocol is expected to have much less delay than this, because mobile nodes are more closely located to each other and as a result of that, most communications will be done using the routing table as a proactive approach.

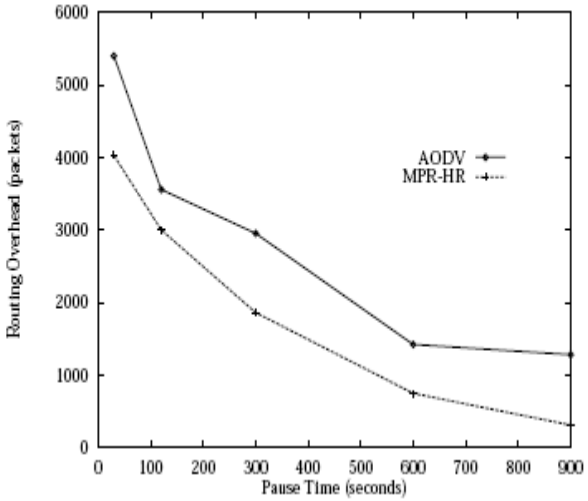
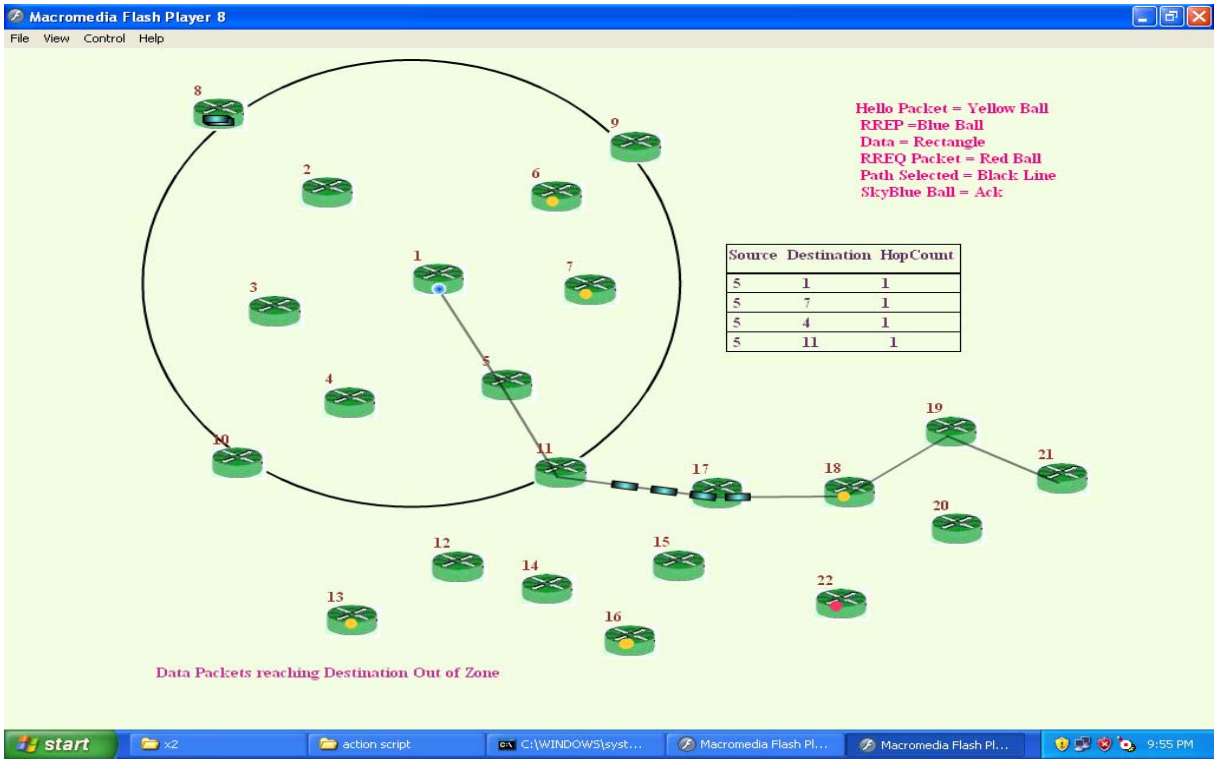
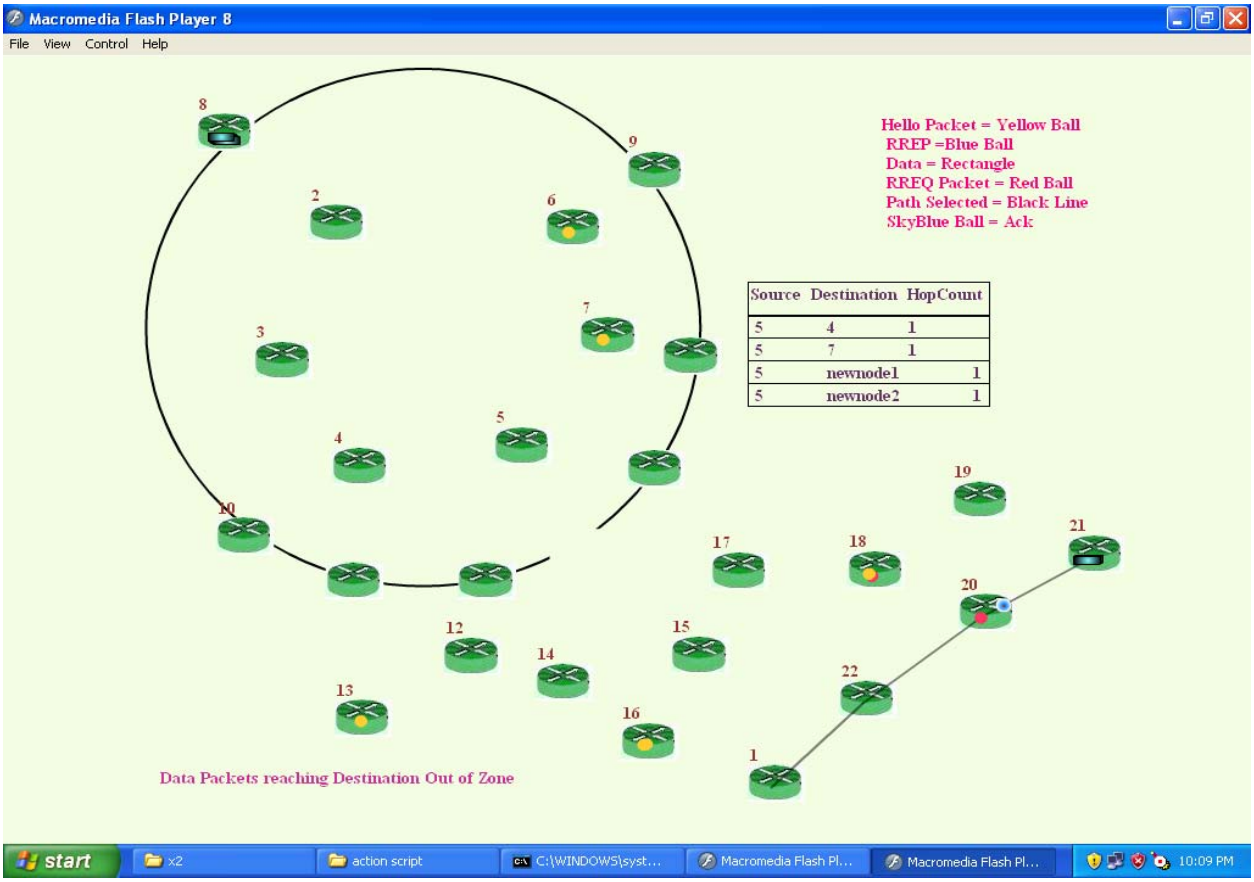


Figure 5.5: Comparison of Efficiency in terms of Routing Overhead





REFERENCES:

[1] J. Broch et al., "A Performance Comparison of Multihop Wireless Ad Hoc Network Routing Protocols," Proceedings of ACM/IEEE MobiCom'98, October 1998.

[2] CMUMonarch Project, "The CMUMonarch Project'sWireless and Mobility Extensions to ns," Carnegie MellonUniversity,August1999. http://www.monarch.cs.cmu.edu/.

[3] M.S. Corson et al., "An InternetMANET Encapsulation Protocol (IMEP) Specification," Internet Draft, draft-ietf-manetimep- spec-01.txt, August 1998.

[4] K. Fall and K. Varadhan, "ns Notes and Documentation," The VINT Project, UC Berkeley, May 1998. Work in progress. http://www-mash.cs.berkeley.edu/ns/.

[5] P. Jacquet, P. Muhlethaler and A. Qayyum, "Optimized Link State Routing Protocol," Internet Draft, draft-ietf-manetolsr-01.txt, November 1998.

[6] D.B. Johnson and D.A. Maltz, "Dynamic Source Routing in Ad Hoc Wireless Networks," Mobile Computing, Kluwer Academic Publishers, pp. 153-181, 1996. Proceedings of ACM/IEEE MobiCom'98, October 1998.

[7] S. Lee and C. Toh, "A Simulation Study of Table-Driven and On-Demand Routing Protocols for Mobile Ad Hoc Networks," IEEE Network, July/August 1999.

[8] V.D. Park and M.S. Corson, "A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks," Proceedings of IEEE INFOCOM'97, April 1997.

[9] C.E. Perkins and P. Bhagwat, "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers," Proceedings of ACM SIGCOMM'94, pp. 234- 244, August 1994.

[10] C.E. Perkins and E.M. Royer, "Ad Hoc On Demand Distance Vector (AODV) Routing," Internet Draft, draft-ietf-manetaodv- 02.txt, November 1998.