An Enhanced Reliable Ad Hoc Multicasting Protocol

A. Ibrahim **M. Hashem** **A. Fahmy** **F. Amer**

Abstract - ***A Mobile Ad Hoc Networks (MANET) is characterized by a lack of infrastructure, and by a random and quickly changing network topology; thus the need for a robust dynamic routing protocol that can accommodate such an environments is very important. Consequently, many multicast routing algorithms have been deployed to achieve a high data delivery ratio with very limited control overhead. This paper presents an enhanced reliable Ad Hoc multicasting protocol, E-PUMA, of PUMA (protocol for unified multicasting through announcements) that satisfy efficiency and robustness and is effective in relative applications; after words a comparative study has been made between the proposed protocol and the conventional protocol.***

Keywords: Ad Hoc Networks, Multicasting.

1. Introduction

Traditional network routing techniques fall short when asked to provide mobile hosts with a reliable connection in a wireless environment. Wireless links allow for a high degree of mobility, but have two obstacles; first, they support low data rates second, they have a limited range that can lead to frequent link failures. These two obstacles necessitate a new approach to routing protocols. An emerging class of networks, known as Mobile Ad Hoc Networks [1], promises to provide connectivity among hosts in a highly volatile environment, while minimizing routing overhead.

Wireless networks do not share the robust and high-speed links enjoyed by their wired counterparts. Wireless connections have a small data carrying capacity, a relatively high error rate, and are unreliable when compared to traditional wired connections. MANET may be an adequate solution to the wireless networking problem. MANETs operate independently of a fixed backbone network, conserve bandwidth, and react quickly to changes in network topology.

Multicasting is the transmission of datagrams to a group of hosts identified by a single destination address and hence is intended for group-oriented computing. In MANETs, multicasting can efficiently support a variety of applications that are characterized by close collaborative efforts. It has a self-organizing capability and can be effectively used where other technologies either fail or cannot be deployed effectively. Advanced features of wireless mobile systems, including data rates compatible with multimedia applications, global roaming capability, and coordination with other network structures, are enabling new applications. Therefore, if we can efficiently combine the features of a MANET with the usefulness of multicasting, it will be possible to realize a number of envisioned group-oriented applications [2].

Ad hoc wireless networks are self-organizing, dynamic topology networks formed by a collection of mobile nodes through radio links. Minimal configuration, absence of infrastructure, and quick deployment, make them convenient for emergency situations other than military applications. Multicasting plays a very crucial role in the application of Ad hoc networks. As the number of participants increases, scalability of the multicast protocol becomes an important issue.

The majority of applications are in areas where rapid deployment and dynamic reconfiguration are necessary and a wire line network is not available. These include military battlefields, emergency search and rescue sites, classrooms, and conventions where participants share information dynamically using their mobile devices. These applications lend themselves well to multicast operation where multicasting can improve the efficiency of the wireless links, when sending multiple copies of messages.

The rest of the paper is organized as follows: Section 2 describes briefly the related work of Ad Hoc multicasting protocols, section 3 present overview of the PUMA protocol, section 4 present our proposed protocol, the simulation results and performance evaluation are presented in section 5, finally the conclusion is presented in section 6.

2. Multicast Routing Protocols

One straightforward way to provide multicast in a MANET is through flooding. With this approach, data packets are sent throughout the MANET, and every node that receives this packet broadcasts it to all its immediate neighbor nodes exactly once. It is suggested that in a highly mobile ad hoc network, flooding of the whole network may be a viable alternative for reliable multicast. However, this approach has considerable overhead since a number of duplicated packets are sent and packet collisions do occur in a multiple-access-based MANET. We can classify these protocols into two categories based on how routes are created to the members of the group: Tree-based and Mesh-based

2.1 Tree-based Multicast Routing Protocols

In tree-based multicast protocols, there is only one path between a source-receiver pair. The main drawback of these protocols is that they are not robust enough to operate in highly mobile environments.

Bandwidth Efficient Multicast Routing Protocol (BEMRP): BEMRP[5] tries to find the nearest forwarding node, rather than the shortest path between source and receiver. The multicast tree construction is initiated by the receivers. When a receiver wants to join the group, it initiates flooding of Join control packets. The existing members of the multicast tree, on receiving these packets, respond with Reply packets. When many such Reply packets reach the requesting node, it chooses one of them which has the lowest hop count and sends a Reserve packet on the path taken by the chosen Reply packet.

Multicast Routing Protocol Based on Zone Routing (MZRP): In zone routing[6], each node is associated with a routing zone. For routing, a pro-active approach is used inside the zone (the node maintains the topology inside the zone, using a table-driven routing protocol), whereas a reactive approach is used across zones. To create a multicast delivery tree over the network, the source initiates a two-stage process. In the first stage, the source tries to form the tree inside the zone by sending a TREE-CREATE control packet to nodes within its zone through unicast routing as it is aware of the topology within its zone, then receivers which are interested in joining the group, replies with a TREE-CREATE-ACK packet and forms the route, and then in the second stage to extend the tree outside the zone the source sends a TREE- PROPAGATE packet to all the border nodes of the zone.

Multicast Core-Extraction Distributed Ad Hoc Routing (MCEDAR): A source-tree over an underlying mesh infrastructure called *mgraph* is used for forwarding data packets. The MCEDAR [3] architecture is used by this protocol for the mesh construction. In this architecture, a minimum dominating set (MDS) [7], which consists of certain nodes (called core nodes) in the network, is formed using a core computation algorithm. After joining the MDS, each core node issues a piggy-backed broadcast through its beacon packet to inform its presence up to the next three hops. This process helps each core node to identify its nearby core nodes and to build virtual links. When a new receiver wants to join the multicast group, it requests its dominator to transmit a *JoinReq* packet. The *JoinReq* packet consists of an option called *JoinID*, which is used to prevent any loop formation in the *mgraph.* Initially, the value of *JoinID* is set to infinity. When a tree node of the multicast group receives this *JoinReq* packet, it replies with a *JoinAck* packet if its *JoinID* is less than the *JoinID* of the requesting node. Before sending the *JoinAck*, the node sets its own *JoinID* in the *JoinAck* packet.

Associativity Based Ad Hoc Multicast Routing (ABAM): ABAM[8] is an on-demand source-tree-based multicast protocol in which a path (from source to receiver) is constructed based on link stability rather than hop distance (the number of beacons continuously received from neighboring nodes which reflects the stability of the link). Hence, this multicast protocol is adaptive to the network mobility. The source node initiates the multicast tree construction phase. Joining a group is a three-step process: flooding by the source, replies along the stable path by the receivers, and the source sends Setup packets to all receivers in order to establish the multicast tree.

Differential Destination Multicast Routing Protocol (DDM)[9,10]: It is particularly applicable where the group size is small. To join a particular multicast session, each interested destination node unicasts a Join control packet to the source. When the source receives a Join packet from a destination, it sends an ACK control packet to the destination after storing the destination address in its member list (ML). Each destination periodically sends Join control packets to the source. These Join control packets refresh the ML table at the source. The source removes stale member information if it does not receive any Join message from that particular destination for a certain time period.

Weight-Based Multicast Protocol (WBM)[11]: The main aim here is to find the best point of entry for a new node joining the multicast group. A receiver-initiated approach is adopted here. When a new receiver intends to join the group, it broadcasts a JoinReq packet with a certain time-to-live (TTL) value set. These JoinReq packets are forwarded until they are received by a tree node. Upon receiving a JoinReq packet, a tree node sends a JoinReply (Reply) packet. There can be several such replier nodes which send Reply packets. The Reply packet initially contains the distance of the node from the source (hop's count).

Preferred Link-Based Multicast Protocol (PLBM): The main concepts involved in PLBM[12] are the selection of a set of links to neighbor nodes, called preferred links, and the use of only those links for forwarding of JoinQuery packets. PLBM is a tree-based receiver-initiated protocol. Each member node is responsible for getting connected to the multicast source. Each node maintains its two-hop local network topology information and multicast tree information in two tables: neighbors neighbors table (NNT) and connect table (CT), respectively. Every node in the network periodically transmits small control packets called beacons. On receiving each beacon, a node updates the corresponding entry in its NNT.

Multicast Ad Hoc On-Demand Distance Vector Routing Protocol (MAODV): MAODV [13, 14] maintains a shared tree for each multicast group, consisting of only receivers and relays. Sources wishing to send to the group acquire routes to the group on demand in a way similar to the ad hoc on demand distance vector AODV[15] protocol. Each multicast tree has a group leader, which is the first node to join the group in the connected component. The group leader in each connected component periodically transmits a group hello packet to become aware of reconnections. Receivers join the shared tree with a special route request. The route replies coming from different multicast tree members specify the number of hops to the nearest tree member. The node wishing to join the tree joins through the node reporting the freshest route with the minimum hop count to the tree. Although the performance of MAODV is very good for small groups, low mobility, and light traffic loads, its performance degrades sharply once a given value of group size, mobility, or traffic load is reached, which is due to a sharp increase in the MAODV control packets transmitted to maintain the multicast tree of a group.

Ad Hoc Multicast Routing Protocol Utilizing Increasing ID-Numbers (AMRIS): AMRIS [16] is an on-demand, source-initiated, shared-tree-based multicast protocol. In this protocol, each node in a multicast session generates session-specific *multicast session member id* (msm-id), after receiving the *NEW-SESSION* message from its parent node. The NEW-SESSION message transmission is initiated by a special node called *Sid*, at which the shared tree is rooted. The msm-id increases from the root towards leaf nodes radically, which indicates the flow of multicast data. The protocol uses periodic, short broadcast beacon packets to determine whether a link has been broken. Upon link break, it executes a *branch reconstruction* process to maintain the multicast tree.

Ad Hoc Multicast Routing Protocol (AMRoute): AMRoute [4] assumes the existence of a unicast routing protocol in the network environment but it is independent of a specific unicast routing protocol. This protocol has two main phases - mesh creation and virtual user-multicast tree creation. After formation of mesh by the logical core, it periodically creates a virtual multicast tree over the mesh. This multicast tree uses unicast tunnels to connect group members. Due to the underlying mesh, there is no need for frequent tree readjustments, thus providing robustness in a high mobility environment.

Adaptive Shared-Tree Multicast Routing Protocol [17]: A shared-tree Multicast Routing Protocol is rooted at the rendezvous point (RP) and is shared by multiple sources. Shared-tree-based protocols are scalable as the number of sources in the multicast group increases. In multicast shared mode, sources send unencapsulated data packets to RP so that intermediate node can forward the data packets to receiver directly, after receiving them from the source. In multicast sender mode, intermediate node does not forward data packets (sent by the source) when it receives them from RP because it has already seen those unencapsulated data packets.

Multicast Optimized Link State Routing (MOLSR): It operates as a table driven and proactive protocol, thus regularly exchanges topology information with other nodes of the network. The nodes which are selected as multipoint relays (MPR) by some neighbor nodes periodically announce this information in their control messages. Thereby, a node announces to the network, that it has reachability to the nodes which have selected it as MPR.

All nodes which have multicast capabilities (Multicast Routers), periodically declare themselves in entire network via disseminating MC_CLAIM message. Any source ready to send multicast data to a specific multicast group floods a SOURCE_CLAIM message within the ad hoc network (MPR based flooding), to enable nodes which are members of this group. By this it shows its presence and all the member nodes can attach themselves to the particular multicast group tree. Trees are periodically refreshed, by flooding SOURCE_CLAIM and CONFIRM_PARENT messages. Tree updates are triggered by the detection of topology changes.

Adaptive Demand Driven Multicast Routing (ADMR): ADMR[23] maintains source-based trees, i.e., a multicast tree for each source of a multicast group. A new receiver performs a network-wide flood of a multicast solicitation packet when it needs to join a multicast tree. Each group source replies to the solicitation, and the receiver sends a receiver join packet to each source answering its solicitation.
An individual source-based tree is maintained by periodic keep-alive packets from the source, which allow routers to detect link breaks in the tree by the absence of data or keep-alive packets. A new source of a multicast group also sends a network-wide flood to allow existing group receivers to send receiver joins to the source.

2.2 Mesh-based multicast routing protocols

Multicast routing protocols which provide multiple paths between a source-receiver pair are classified as mesh-based multicast routing protocols. The presence of multiple paths adds to the robustness of the mesh-based protocols at the cost of multicast efficiency.

On-Demand Multicast Routing Protocol (ODMRP): ODMRP is an ad hoc multicast routing protocol based on a multicast mesh [18, 27]. In ODMRP, if a source node has data to send, it periodically broadcasts “Join Request” to find and maintain multicast routes. All the other nodes re-broadcast the packet when they receive non-duplicate one. When a multicast group member receives “Join Request”, the node replies with “Join Table” and subsequent replies by the nodes along a reverse path establish a route. ODMRP uses soft states, so leaving a group is automatically handled by timeout. As shown, ODMRP relies on frequent network-wide flooding, which may lead to a scalability problem when the number of source nodes is large. The control packet overhead becomes more prominent when the multicast group is small in comparison with the entire network.

Dynamic Core-Based Multicast Routing Protocol (DCMP): DCMP [19] is an extension to ODMRP that designates certain sources as cores and reduces the number of sources performing flooding. In DCMP, there are three kinds of sources: passive sources, active sources, and core active sources. Each passive source is associated with a core active source, which plays the role of a proxy for the passive source. The mesh establishment protocol is similar to that in ODMRP. Data packets of the active sources and core active sources are sent over the mesh created by themselves, while a passive source forwards the packet to its proxy core active node, which in turn sends it over its mesh. The control overhead is reduced, as compared to ODMRP, because there are a fewer number of sources which flood their *JoinReq* packets, and thus the number of forwarding nodes is also fewer.

Forwarding Group Multicast Protocol(FGMP): FGMP [20] is a receiver-initiated multicast routing protocol which is based on the forwarding group concept. When receivers want to join the multicast group, they send *JoinReq* control packets. After receiving the JoinReq control packets, sources update their member tables. After refreshing the member tables, each source creates its forwarding table and broadcasts it. The forwarding table contains the next-hop information, which is obtained by using the underlying unicast routing protocol. After receiving the forwarding table, neighbor nodes, whose node ID matches with the entry in the forwarding table, set their forwarding flags and become the forwarding nodes for that particular multicast group. Now nodes 16 and 19 build their own forwarding tables and forward them again. In this way, the forwarding tables reach receivers along the reverse shortest path and the route is established

Neighbor Supporting Ad Hoc Multicast Routing Protocol (NSMP): NSMP [21] is a mesh-based multicast protocol which does selective and localized forwarding of control packets. To initialize the mesh, the source floods the control message throughout the network. But for maintenance of the mesh, local route discovery is used, that is, only mesh nodes and multicast neighbor nodes forward the control message to refresh the routes. Multicast neighbor nodes are those nodes which are directly connected to at least one mesh node.

In order to form the multicast mesh, initially the multicast source floods the *FLOOD-REQ* packets which are forwarded by all the nodes in the network. When receiver receives this *FLOOD-REQ* packet, it sends a reply packet (REP) along the reverse path to the source, establishing the route. Each source node periodically transmits a *LOCAL-REQ* packet, which is relayed by all mesh nodes and multicast neighbor nodes. If new receiver wants to join the multicast group, it waits for a specified time for a *LOCAL-REQ* packet from any mesh node or multicast neighbor node. When this new receiver receives a *LOCAL-REQ* packet, from the multicast neighbor node, it joins the group by sending a REP control packet.

Core-Assisted Mesh Protocol (CAMP): CAMP [22] uses core nodes in the mesh. This protocol expands the idea of *core based tree*, to form the mesh. But unlike the core based tree protocol, it contains more than one core. When any node wants to join the multicast group, it sends a *join Request* to a core node if none of its neighbor nodes are present in that particular multicast group. If all core nodes are unreachable, it uses an *expanded ring search* method to reach any group member.

3. The PUMA protocol

PUMA[29] (protocol for unified multicasting through announcements) supports the IP multicast service model of allowing any source to send multicast packets addressed to a given multicast group, without having to know the constituency of the group. Furthermore, sources need not join a multicast group in order to send data packets to the group.

Like CAMP and MAODV, PUMA uses a receiver initiated approach in which receivers join a multicast group using the address of a special node (core in CAMP or group leader in MAODV), without the need for network-wide flooding of control or data packets from all the sources of a group. Like MAODV, PUMA eliminates the need for a unicast routing protocol and the pre-assignment of cores to multicast groups.

PUMA implements a distributed algorithm to elect one of the receivers of a group as the core of the group, and to inform each router in the network of at least one next-hop to the elected core of each group. The election algorithm used in PUMA is essentially the same as the spanning tree algorithm introduced by Perlman for internetworks of transparent bridges [28]. Within a finite time proportional to the time needed to reach the router farthest away from the eventual core of a group, each router has one or multiple paths to the elected core.

Every receiver connects to the elected core along all shortest paths between the receiver and the core. All nodes on shortest paths between any receiver and the core collectively form the mesh. A sender sends a data packet to the group along any of the shortest paths between the sender and the core. When the data packet reaches a mesh member, it is flooded within the mesh, and nodes maintain a packet ID cache to drop duplicate data packets.

PUMA uses a single control message for all its functions, the *multicast announcement*. Each multicast announcement specifies a sequence number, the address of the group (group ID), the address of the core (core ID), the distance to the core, a mesh member flag that is set when the sending node belongs to the mesh, and a parent that states the preferred neighbor to reach the core. Successive multicast announcements have a higher sequence number than previous multicast announcements sent by the same core.

With the information contained in such announcements, nodes elect cores, determine the routes for sources outside a multicast group to unicast multicast data packets towards the group, notify others about joining or leaving the mesh of a group, and maintain the mesh of the group.

For the same core ID, only multicast announcements with the highest sequence number are considered valid.

For the same core ID and sequence number, multicast announcements with smaller distances to the core are considered better. When all those fields are the same, the multicast announcement that arrived earlier is considered better. After selecting the best multicast announcement, the node generates the fields of its own multicast announcement in the following way:

- **Core ID:** The core ID in the best multicast announcement
- **Group ID:** The group ID in the best multicast announcement
- **Sequence number**: The sequence number in the best multicast announcement
- **Distance to core**: One plus the distance to core in the best multicast announcement
- **Parent**: The neighbor from which it received the best multicast announcement
- **Mesh member**: Receivers consider themselves mesh-members and set the mesh member flag to TRUE.

The connectivity list stores information about one or more routes that exist to the core.

4. The proposed (E-PUMA) protocol

It should be clear that the conservation of bandwidth is imperative to the success of any wireless network. While previous MANET multicast protocols focused only on the reductions of control overhead, the multicast protocol investigated in this study attempts to reduce the amount of bandwidth used by the network both in terms of control overhead and data rebroadcasts. It can usually be assumed that data transmission consumes more bandwidth than control overhead. Even a small decrease in data retransmissions should substantially improve network performance. Unlike previously proposed MANET multicast algorithms, this new protocol will focuses on:

- **Sequence number:** Generated by the core.
- **Route length:** Number of hops to the core.
- **Route load:** Detects the route which has the minimum load.
- **Route *stability/Quality:*** The route that will remain connected for the longest duration of time

In what follows the basic modules of the proposed protocol will be discussed.

4.1 Mesh Establishment Phase

In mesh establishment phase we use a receiver initiated approach in which receivers join a multicast group using the address of a special node (core ID), without the need of network-wide flooding of control or data packets from all the senders of a group.

We implement the same algorithm of PUMA to elect one receiver of a group to be the core of this group, which is essentially the same as the spanning tree algorithm introduced by Perlman for internetworks of transparent bridges [28].

Every receiver connects to the elected core along all shortest paths between the receiver and the core. All nodes on shortest paths between any receiver and the core collectively form the mesh. A sender sends a data packet to the group along any of the shortest paths between the sender and the core. When the data packet reaches a mesh member, it is flooded within the mesh, and nodes maintain a packet ID cache to drop duplicate data packets.

We use a single control message for all its functions, the *multicast announcement*. Each multicast announcement consists of:

- **Sequence Number**: Successive multicast announcements have a higher sequence number than previous multicast announcements sent by the same core.
- **Group address** (group ID).
- **Core address** (core ID)
- Distance to the core.
- **Mesh member**: flag that is set when the sending node belongs to the mesh.
- **Parent:** that states the preferred neighbor to reach the core.

We added these fields to the multicast announcements

- X-Y Coordinates of the node.
- Node speed.
- Node load.

During the scenario each node measures its traffic load in the last period (Here, the load period is 3 seconds). Each node gets the control packet calculates the available time between this node and the neighbors. With the information contained in such announcements, nodes elect cores, determine the routes for senders to unicast multicast data packets towards the group and maintain the mesh of the group.

4.2 Data Transfer Phase

A node that believes itself to be the core of a group transmits multicast announcements periodically for that group. As the multicast announcement travels through the network, it establishes a *connectivity list* at every node in the network. Using connectivity lists, nodes are able to establish a mesh, and route data packets from senders to receivers.

A node stores the data from all the multicast announcements it receives from its neighbors in the connectivity list. Fresher multicast announcements from a neighbor (i.e., one with a higher sequence number) overwrite entries with lower sequence numbers for the same group.

For the same core ID and sequence number, multicast announcements with smaller distances to the core are considered better. When all those fields are the same, the multicast announcement that the neighbors has minimum load is considered better. The last check is to detect the route that will remain connected for the longest duration of time. After selecting the best multicast announcement, the node generates the fields of its own connectivity list in the following way:

- **Core_ID:** The core ID in the best multicast announcement.
- **Group_ID:** The group ID in the best multicast announcement.
- **Next_Hop:** The neighbor node.
- **Parent**: The neighbor from which it received the best multicast announcement.
- **Distance_to_Core:** One plus the distance to core in the best multicast announcement.
- **Sequence number:** The sequence number in the best multicast announcement.
- **Time_Received:** The time of the multicast announcement received.
- **Mesh_Member:** Receivers consider themselves mesh-members and set the mesh member flag to TRUE.
- **The Load:** The traffic load of the neighbor node.
- **Route Stability:** Using [(X, Y), Speed] of the current node and the neighbor the duration that the link between the two nodes stays connected is calculated, using the worst case (the nodes are moving in opposite sides).

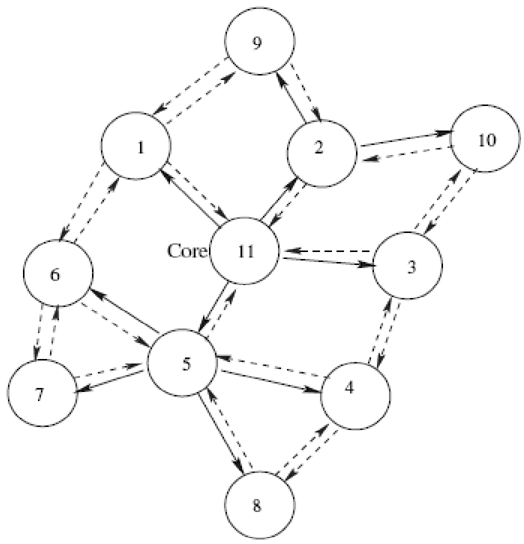


Fig. 1: Mesh Creation

Neighbor	Multicast Announcement		Time (ms)
	Distance To Core	Parent	
5	1	11	12152
1	1	11	12180
7	2	5	12260

Table 2: Connectivity List of PUMA at node 6

Neighbor	Multicast Announcement				Time (ms)
	Distance To Core	Parent	Load %	Stability (sec)	
1	1	11	30	40	12180
5	1	11	40	40	12152
7	2	5	60	30	12260

Table 2: Connectivity List of E-PUMA at node 6

Within a finite time the forwarding mesh is constructed and every node in the network will have the routing information of the new multicast session in the Connectivity List. The sender can receive multiple Receiver Control packets from multiple nodes in the forwarding group. The sender chooses one of the routes, as an active route, according to the path quality and sends the data packets through it.

As shown in Fig. 1, Table 1 and Table 2 show the two connectivity lists of the two protocols, PUMA and E-PUMA respectively, at certain group/core/sequence number for node 6, there is a difference between the two protocols because the two parameters “Node Load” and “Node Stability” are added in the enhanced protocol. The next section will show the effect of this enhancement.

5. Performance Evaluation

In this section a case study, as shown in Table 3 which consists of 50 simulated wireless mobile nodes roaming in a 1500 meters x 300 meters flat space for 900 seconds of simulated time. The radio transmission range is 250 meters. A free space propagation channel is assumed. Group scenario files determine which nodes are receivers or senders and when they join or leave a group. It is assumed that a multicast member node joins the multicast group at the beginning of the simulation (first 30 seconds) and remains as a member throughout the whole simulation.

Hence, the simulation experiments do not account for the overhead produced when a multicast member leaves a group, multicast senders start and stop sending packets in the same fashion (number of packets per second, each packet has a constant size of 256 bytes).

Each mobile node moves randomly at a preset average speed according to a “random waypoint model”. Here, each node starts its journey from a random location to a random destination with a randomly chosen speed. Once the destination is reached, another random destination is targeted after a pause. By varying the pause time, the relative speeds of the mobiles are affected.

Here the pause time is always set to zero to create a harsher mobility environment. The speeds are used chosen between 0m/s to 20m/s.

The metric used for our evaluation is packet delivery ratio which is defined as the data packets delivered divided by the data packets expected to be delivered. The data packets expected to be delivered is the data packets sent times number of receivers. This metric represents the multicast routing efficiency.

Simulator	NS-2.32
Total Nodes	50
Simulation Time	900 sec
Simulator Area	1500 x 300
Node Placement	Random
Pause Time	0
Mobility Modem	Random Waypoint
Radio Range	250 meter
Data Packet Size	256 bytes

Table 3: Simulation Environment.

To compare the proposed protocol with the conventional protocol, four experiments are performed to explore the performance with respect to some parameters such as: Traffic load, number of senders, number of receivers, and node mobility. The details of each experiment performed are as follows:

- **Experiment 1:** Traffic Load varied across 1, 2, 5, 10, 20, 25, 100 pkts/sec. Mobility = 0, (Senders , Receivers) = (1,1), (2,2), (3,3), (4,4), (1,10), (2,10), (5,10), (10,30). Multicast groups = 1.
- **Experiment 2:** Senders varied across 1, 2, …., 10 Mobility = 5 m/s, Members = 10, Traffic Load = 10 pkts/sec. Multicast groups = 1.
- **Experiment** 3: Receivers varied across 1,2, ….30 Mobility = 5 m/s, Senders = 5, Traffic Load = 10 pkts/sec, Multicast groups = 1.
- **Experiment 4:** Mobility varied across 0, 1, 2, 4, 6, 10, 20 m/s. Senders = 5, Receivers = 5, Traffic Load = 10 pkts/sec, Multicast groups = 1.

A. The impact of traffic load

In traffic load experiment, node mobility speed is moderate with maximum speed 5 m/s, because we want to focus on packet drops caused by congestion. Both the senders and receivers were chosen randomly from among the 50 nodes. Traffic load was equally distributed among all senders.

The packet delivery ratio (PDR) as a function of the traffic load which changes from 1 pkt/sec to 100 pkt/sec is presented in different cases as shown in table 4.

Case	No. of Senders	No. of Receivers
1	1	1
2	2	2
3	3	3
4	4	4
5	1	10
6	2	10
7	5	10
8	10	30

Table 4: Experiment’s cases

Fig. 2.1: Packet Delivery Ratio Vs Traffic load
Case 1: Senders=1 & receivers=1

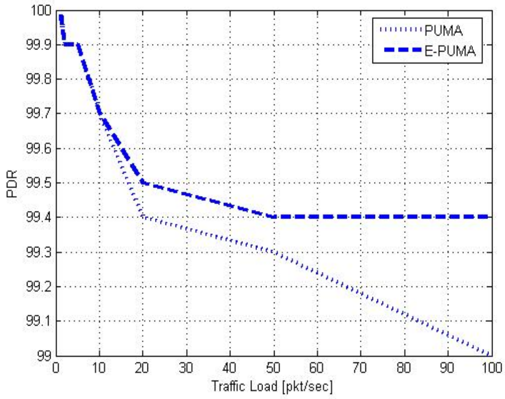


Fig. 2.2: Packet Delivery Ratio Vs Traffic load
Case 2: Senders=2 & Receivers=2

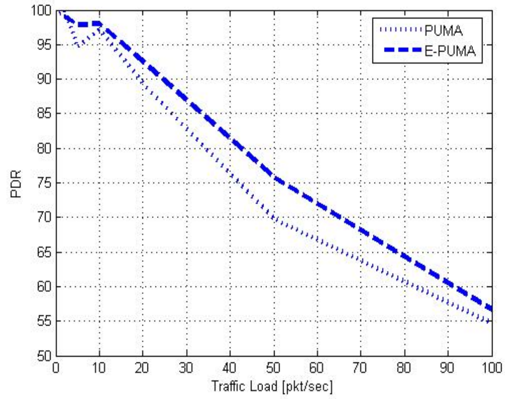


Fig. 2.3: Packet Delivery Ratio Vs Traffic load
Case 3: Senders=3 & Receivers=3

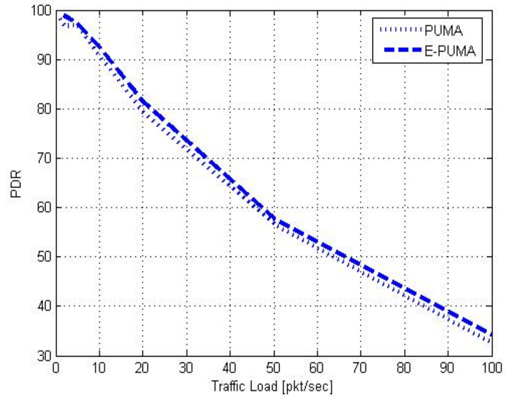


Fig. 2.4: Packet Delivery Ratio Vs Traffic load
Case 4: Senders=4 & Receivers=4

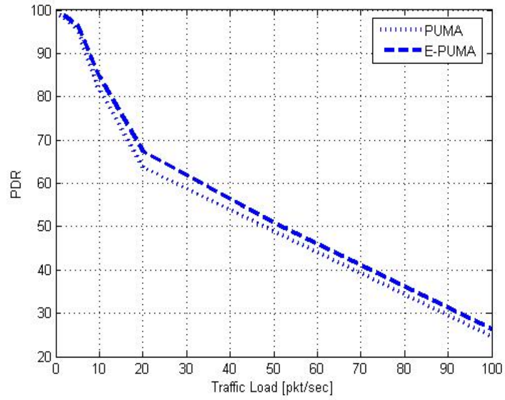


Fig. 2.5: Packet Delivery Ratio Vs Traffic load
Case 5: Senders=1 & Receivers=10

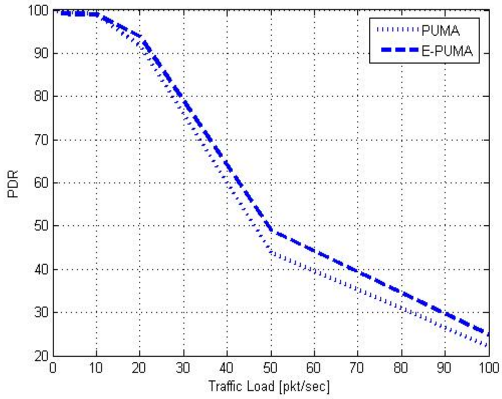


Fig. 2.6: Packet Delivery Ratio Vs Traffic load
Case 6: Senders=2 & Receivers=10

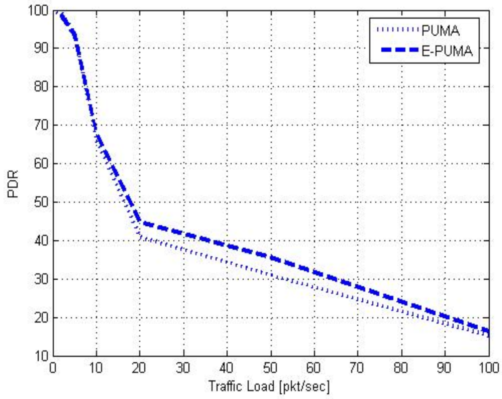
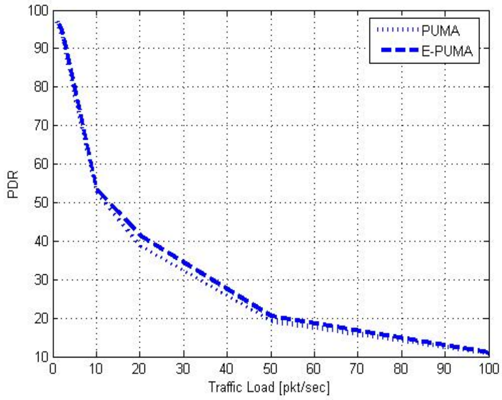


Fig. 2.7: Packet Delivery Ratio Vs Traffic load
Case 7: Senders=5 & Receivers=10

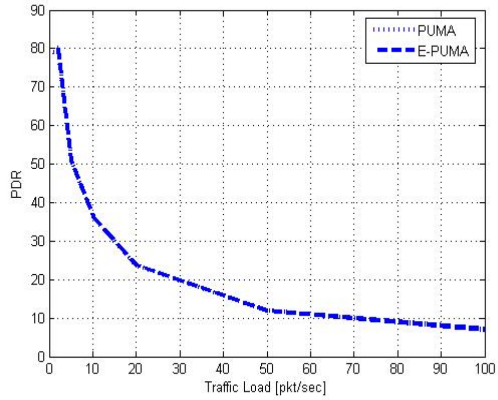


As shown in figures 2.1 to 2.8, E-PUMA maintains packet delivery higher than PUMA.

However, the performance of the two protocols exponentially degrades. And it becomes serious when network load increases and the number of senders and receivers greater than 10, but still E-PUMA is better than PUMA.

The original senders add considerable extra traffic to the network, which raises collision risk and introduces congestion. That's why the packet delivery ratio decreases after 5 senders and decreases more quickly in the case of more than 10 senders.

Fig. 2.8: Packet Delivery Ratio Vs Traffic load
Case 8: Senders=10 & Receivers=30



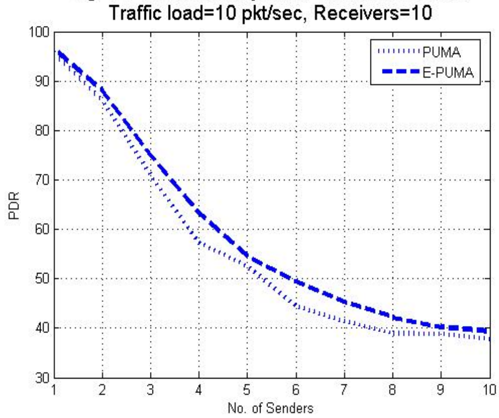
This phenomena can be explained by the fact that wireless channel is saturated around senders which prevent them to receive further control packets. It also explains why packet delivery ratio decreases so quickly from 10 senders. As a result of using "Node-Load" parameter which equally distribute the traffic load E-PUMA is almost better Than PUMA.

B. The impact of number of senders

In case the number of multicast senders increased from 1 to 10, traffic load at 10 pkt/sec, node mobility speed is moderate with maximum speed 5 m/s, and the number of receivers is 10.

Fig. 3 shows the packet delivery ratio as a function of the number of senders for the two protocols, it is clear that PDR in case of E-PUMA is more than in case of PUMA.

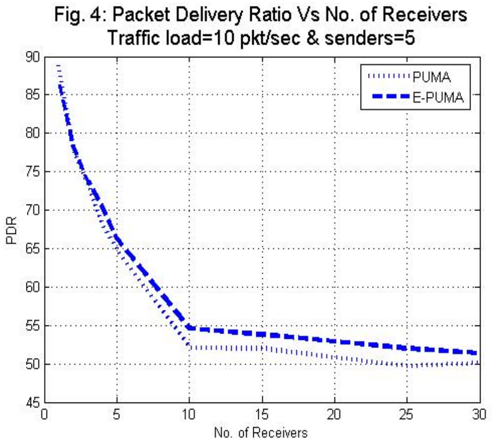
Fig. 3: Packet Delivery Ratio Vs No. of Senders
Traffic load=10 pkt/sec, Receivers=10



C. The impact of number of receivers

In this experiment, node mobility speed is moderate with maximum speed 5 m/s. The number of multicast receiver increases from 1 to 30.

Fig. 4 illustrates the packet delivery ratio as a function of the number of receivers for the two protocols. E-PUMA presents higher performance than PUMA for all number of receivers. So E-PUMA is more reliable when the number of receivers augments



.

D. The impact of node mobility

In this aspect, the maximum movement speed of nodes range in the set {0, 1, 5, 10, 15, and 20} m/s, the number of senders is fixed to 5, the number of receivers is fixed to 5 and the traffic load is 10 pkt/sec.

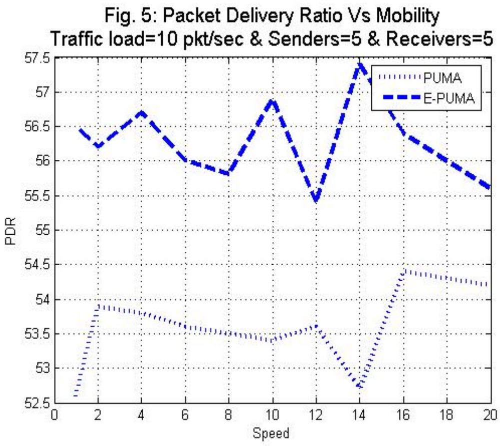


Fig. 5 shows the packet delivery ratio with different maximum speed of the two protocols. The results show that E-PUMA is reliable against frequent topology changes: mobility has nearly no impact on the performance of E-PUMA while frequent topology changes degrade the performance of PUMA multicast routing protocol. PUMA gives a worse performance than E-PUMA.

E-PUMA is more reliable facing to topology changes and deliver data packets than PUMA in all mobility cases. This protocol is also scalable in the sense that it distributes the load all over the nodes and prefers the route which has the maximum available time.

6. Conclusion

This paper presented a new multicast routing protocol for Ad Hoc networks which is useful for real-time / disaster environment applications.

The new routing protocol enhances the performance of reliable multicast and reduces the bandwidth utilization overhead in maintaining the network topology.

The first key concept is to fair the distribution of the data packets among nodes according to the states of nodes load, the second key is to use the most stable route while preserving the network robustness.

The new protocol is compared with the conventional protocol.

The Real-Time/|disaster applications are probably the most difficult ad hoc applications to handle when it comes to mobility management and mobile communication, since some issues should be taken into concern:

- **Hostile enemy** – If the enemy can get the communication in the network to stop function properly or be able to tamper with the messages, the enemy can get great advantages.
- **Trust models** – How to deal with the level of trust and compromised nodes.
- **Quality of service control** – Not all nodes and packets are equal.
- **Radio power usage restrictions** – Battery, reveal location, time and importance of the node.

So, the future work is looking forward to embedding a security algorithm in the proposed protocol to resist the passive and active attacks.

References

[1] Charles E. Perkins, AD HOC Networking", Addison-Wesley, 2001.

[2] C. Siva Ram Murthy, B. S. Manoj "Ad Hoc Wireless Networks Architectures and Protocols", PRENTICE HALL, 2005.

[3] P. Sinha, R. Sivakumar, and V. Bharghavan, "MCEDAR: Multicast Core Extraction Distributed Ad Hoc Routing," Proceedings of IEEE WCNC 1999. pp. 1313-1317, September 1999.

[4] E. Bommaiah. M. Liu, A. McAuley, and R. Talpade, "AMRoute: Ad Hoc Multicast Routing Protocol," Internet draft (work in progress), draft-talpade-manet-amroute-OO.txt, August 1998.

[5] T. Ozaki, J. B. Kim, and T. Suda, "Bandwidth Efficient Multicast Routing Pro-tocol for Ad Hoc Networks," Proceedings of IEEE ICCCN 1999, pp. 10-17, October 1999.

[6] V. Devarapalli. A. A. Selcuk, D. Sidhu, "MZR: A Multicast Protocol for Mobile Ad Hoc Networks."' Internet draft (work in progress), draft-vijay-manet-mzr-Ol.txt. July 2001.

[7] P. Sinha, R. Sivakumar, and V. Bharghavan, "CEDAR: A Core-Extraction Distributed Ad Hoc Routing Algorithm," Proceedings of IEEE INFOCOM 1999, pp. 202-209, March 1999.

[8] C. K. Toh, G. Guichala, and S. Bunchua. "ABAM: On-Demand Associativity-Based Multicast Routing for Ad Hoc Mobile Networks," Proceedings of IEEE VTC SOOO, pp. 987-993, September 2000.

[9] L. Ji and M. S. Corson, "Differential Destination Multicast (DDM) Specification," Internet draft (work in progress), draft-ietf-manet-ddm-OO.txt, July 2000.

[10] L. Ji and M.S. Corson, "Differential destination multicast - a manet multicast routing protocol for small groups," in Proceedings of IEEE INFOCOM, April 2001.

[11] S. K. Das, B. S. Manoj, and C. Siva Ram Murthy, "Weight-Based Multicast Routing Protocol for Ad Hoc Wireless Networks," Proceedings of IEEE GLOBE-COM2002, vol. 1, pp. 17-21, November 2002.

[12] R. S. Sisodia, I. Karthigeyan, B. S. Manoj, and C. Siva Ram Murthy, "A Preferred Link-Based Multicast Protocol for Wireless Mobile Ad Hoc Networks," Proceedings of IEEE ICC 2003, vol. 3, pp. 2213-2217, May 2003.

[13] E. M. Royer and C. E. Perkins, "Multicast Operation of the Ad Hoc On-Demand Distance Vector Routing Protocol," Proceedings of ACM MOBICOM 1999, pp. 207-218, August 1999.
[14] E. M. Royer and C. E. Perkins, "Multicast Ad Hoc On-Demand Distance Vector (MAODV) Routing," Internet draft (work in progress), draft-ietf-manet-maodv-OO.txt, July 2000.
[15] C. E. Perkins and E. M. Royer, "Ad Hoc On-Demand Distance Vector Routing," Proceedings of IEEE WMCSA 1999, pp. 90-100, February 1999.
[16] C. W. Wu, Y. C. Tay, and C. K. Toh, "Ad Hoc Multicast Routing Protocol Utilizing Increasing id-numberS (AMRIS) Functional Specification," Internet draft (work inprogress), draft-ietf-manet-amris-spec-OO.txt, November 1998.
[17] C. C. Chiang, M. Geria, and L. Zhang, "Adaptive Shared Tree Multicast in Mobile Wireless Networks," Proceedings of GLOBECOM 1998, pp. 1817-1822, November 1998.
[18] S. J. Lee, M. Geria, and C. C. Chiang, "On-Demand Multicast Routing Protocol," Proceedings of IEEE WCNC 1999, pp. 1298-1302, September 1999.
[19] S. K. Das, B. S. Manoj, and C. Siva Ram Murthy, "A Dynamic Core-Based Multicast Routing Protocol for Ad Hoc Wireless Networks," Proceedings of ACM MOBIHOC S008, pp. 24-35, June 2002.
[20] C. C. Chiang, M. Geria, and L. Zhang, "Forwarding Group Multicasting Protocol for Multi-Hop, Mobile Wireless Networks," ACM/Baltzer Journal of Cluster Computing: Special Issue on Mobile Computing, vol. 1, no. 2, pp. 187-196, 1998.
[21] S. Lee and C. Kim, "Neighbor Supporting Ad Hoc Multicast Routing Protocol," Proceedings of ACM MOBIHOC 2000, pp. 37-50, August 2000.
[22] J. J. Garcia-Luna-Aceves and E. L. Madruga, "The Core-Assisted Mesh Protocol," IEEE Journal on Selected Areas in Communications, vol. 17, no. 8, pp. 1380-1994, August 1999.
[23] J.G. Jetcheva and David B. Johnson, "Adaptive demand-driven multicast routing in multi-hop wireless ad hoc networks," in Proceedings of the ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc), October 2001.
[24] Kevin Fall, and Kannan Varadhan, editors, "ns notes and documentation," The VINT Project, UC Berkeley, LBL, USC/ISI and Xerox PARC, Nov. 1997. Available at http://www-mash.cs.berkeley.edu/ns/.
[25] "The CMU Monarch Projects wireless and mobility extensions to ns", The CMU Monarch Project, August 1999. Available at http://www.monarch.cs.cmu.edu/.
[26] K. Fall, K. Varadhan, editors, "The ns Manual", The VINT Project, UC Berkeley, LBL, USC/ISI, and XEROX PARC, August, 2006 . Available at http://www-isi.edu/nsnam/ns/.
[27] S.J. Lee, W. Su and M. Gerla, Internet Draft, "On-Demand Multicast Routing Protocol(ODMRP) for Ad Hoc Networks", Jan. 2000. draft-ietf-manet-odmrp02.txt
[28] R. Perlman, "An algorithm for distributed computation of a spanning tree in an extended lan," in *ACM Special Interest Group on Data Communication(SIGCOMM)*, 1985, pp. 44–53.
[29] Ravindra Vaishampayan, J.J. Garcia-Luna-Aceves], "Efficient and Robust Multicast Routing in Mobile Ad Hoc Networks," Proceedings of IEEE 2004, pp. 304-313, 2004.