E-ODMRP:Enhanced ODMRP with Motion Adaptive Refresh ⋆

Soon Y. Oh [a,*], Joon-Sang Park [b], Mario Gerla [a]

[a] *Department of Computer Science, University of California, Los Angeles*
Los Angeles, CA 90095,USA

[b] *Department of Computer Engineering, Hongik University*
Seoul, Korea 121-791

Abstract

On Demand Multicast Routing Protocol (ODMRP) is a multicast routing protocol for mobile ad hoc networks. Its efficiency, simplicity, and robustness to mobility render it one of the most widely used MANET multicast protocols. At the heart of the ODMRP's robustness is the periodic route refreshing. ODMRP rebuilds the data forwarding "mesh" on a fixed interval and thus the route refresh interval is a key parameter that has critical impact on the network performance. If the route refresh rate is too high, the network will undergo too much routing overhead wasting valuable resources. If it is too low, ODMRP cannot keep up with network dynamics resulting in packet losses due to route breakages. In this paper, we present an enhancement of ODMRP with the refresh rate dynamically adapted to the environment. Simulation results show that the Enhanced ODMRP (E-ODMRP) reduces the packet overhead by up to a half yet keeping a packet delivery ratio comparable that of original ODMRP. E-ODMRP compares favorably with other published multicast schemes.

Key words: MANET, Multicast, Routing, Simulation.

1 Introduction

A mobile ad hoc network (MANET) is a self-organized and dynamically reconfigurable wireless network without central administration and wired infrastructure. Nodes in the MANET can instantly establish a communication structure while each node moves in an arbitrary manner. Thus MANET is useful for people working in groups to achieve the given task where preexisting infrastructure cannot be accessed or no infrastructure is installed. Accordingly, applications in this environment such as group conferencing, data dissemination, disaster relief and battlefield require multicast routing. Developing a multicast routing protocol for MANET, however, is a different challenge than for wired networks due to characteristics of MANET such as usages of wireless broadcast medium, dynamic topology, limited bandwidth, high packet error rate, etc.

Many MANET multicast routing protocols have been proposed in the literature (e.g., [1, 2, 4, 5, 7-14, 17-27, 29]) and several of them operate in an on demand fashion [1, 2, 8-13, 19 ,27] in which routing information is exchanged only when it is needed. In general, on demand routing protocols employ two-

[⋆] The material in this paper was presented in part at the 2nd International Symposium on Wireless Communication System (ISWCS 2005).

[∗] Corresponding author. (Phone:1-310-825-1888, Fax:1-310-825-2273)
Email addresses: soonoh@cs.ucla.edu (Soon Y. Oh), jsp@hongik.edu (Joon-Sang Park), gerla@cs.ucla.edu (Mario Gerla).

way handshaking to find a path for a sender/receiver pair. The sender floods the network with a Route Request packet and the receivers respond with Route Replys. To limit the scope (and overhead) of flooding, the local recovery approach is introduced. Namely, an alternative route to the destination is searched locally upon detection of the disconnection. Adaptive Demand-Driven Multicast Routing (ADMR) [9] and Multicast Ad hoc On-demand Distance Vector protocol (MAODV) [19] are two examples of an on demand multicast protocol following this approach. They first build a multicast tree between a source and receivers and on detection of a broken link try to repair the route locally. Another popular on demand multicast routing protocol, On Demand Multicast Routing Protocol (ODMRP) [11] relies instead on periodic network-wide flooding for route discovery and maintenance. This design is intended to ensure robustness against mobility and unreliable wireless link propagation.

ODMRP periodically reconstructs the “Forwarding mesh” on a fixed short interval. Thus the route refresh interval is one of the most important performance parameters since it has critical impact on the protocol overhead and thus efficiency. If the refresh interval is too short, the protocol generates more control packets than needed for mesh construction/maintenance wasting valuable resources such as channel bandwidth and if the interval is too long, ODMRP cannot keep up with network dynamics resulting in packet losses due to link breakages. To find the “right” refresh interval, a mobility prediction scheme [12] was previously proposed trying to adapt the refresh interval to nodes’ mobility. However, it requires additional hardware/software supports: a location service such as GPS (Global Positioning System) from which nodes can get their location information; and all nodes’ clock time to be synchronized using NTP (Network Time Protocol) or the GPS clock. Moreover, it assumes the location service to be error-free and free space radio propagation model [18] in which received signal strength solely depends on its distance to the transmitter. Considering its unrealistic assumptions such as an error-free location service and the free space radio propagation, it is unlikely to work as intended in a real deployment even with the additional hardware/software supports. Another problem of the scheme is that since it chooses the minimum link life-time in the mesh as the route refresh interval, unnecessarily frequent network-wide flooding is inevitable when only a small part in the multicast group has high mobility or unstable connection which may admit more efficient solutions such as local recovery. In fact, it generates more control packets than ODMRP as Lee et al. [12] showed when nodes' speed is higher than 50 km/h.

In this paper, we present E-ODMRP an enhanced version of ODMRP with adaptive refresh. Adaptation is driven by receivers' reports on link breakages rather than mobility prediction. And the adaptive refreshing mechanism is seamlessly integrated with a simple and "unified" (i.e., combined) local recovery and receiver joining scheme. As the time between refresh episodes can be quite long, a new node or a momentarily detached node might lose some data while waiting for the route to it to be refreshed and reconstructed. Upon joining or upon detection of a broken route, a node performs a local route recovery procedure instead of flooding to proactively attach itself to a forwarding mesh or to request a global route refresh from the source. Compared to ODMRP, a slightly lower packet delivery ratio might be expected in E-ODMRP in light load since the new scheme uses packet loss as a indicator of a broken link. The major advantage is reduced overhead (by up to 90%) which translates into a better delivery ratio at high loads.

The rest of the paper is organized as follows: Section 2 overviews existing MANET multicast protocols, Section 3 describes our protocol, Section 4 presents simulation results, and finally Section 5 concludes the paper.

2 Background and Related Works

In this section, we introduce ODMRP, the base protocol of E-ODMRP, and ADMR. Also we reivew other MANET multicast routing protocols very briefly.

2.1 ODMRP

ODMRP consists of Query and Reply phases. While a source has packets to transmit, it periodically broadcasts a member-soliciting packet, called Join Query. Upon receiving a non-duplicate Join Query packet, every node in the network stores the upstream node address (i.e., reverse path learning) into the route table and rebroadcasts the packet to its neighbor nodes. When the Join Query packet reaches a multicast receiver, the receiver creates and broadcasts a Join Reply to its neighbors. This Join Reply packet is relayed all the way back to the source following the learned reverse path and the nodes on the reverse path become the forwarding group. Data is delivered by the forwarding group nodes as following: The source broadcasts data packets and on reception of non-duplicate packets nodes in the forwarding group rebroadcast so that packets can be propagated toward the receivers. If the group consists of only two nodes (a sender and a receiver), ODMRP flags nodes along the shortest (delay) path as forwarding nodes. These nodes will then deliver packets from a source to a destination virtually implementing unicast routing as a special case of multicast. More formally, the forwarding group is the set of nodes responsible for forwarding multicast data, essentially forming a mesh structure between all senders and receivers. When a new node wants to join a multicast group as a receiver, it waits a next Join Query and responds with a Join Reply packet. If a receiver is disconnected from the forwarding mesh due to mobility, it should wait for a next Join Query flooding, too. ODMRP's frequent route refresh and redundant forwarding keep high packet delivery ratio without a local route recovery scheme. In ODMRP, a soft-state approach maintains multicast group members; no explicit control message is required to join or leave the group. When a source node has no more packets to send, it simply stops sending a Join Query. If a receiver wants to leave a particular multicast group, it does not respond to the Join Query for the multicast group. A forwarding node becomes a non-forwarder if it reaches a timeout which is a multiple of the refresh interval without receiving the Join Reply.

2.2 ODMRP Variants

PatchODMRP [10] is one of the first derivatives of ODMRP. It uses the same procedure to construct the initial forwarding mesh structure, but it differs from ODMRP in that it take local repairing approach on detection of mesh destruction (i.e., link breakage) to avoid frequent mesh reconstructions. To detect a link breakage, however, every node sends BEACON signal in MAC layer periodically, e.g., every 3 seconds, extra overhead is inevitable. When a forwarding node detects a link breakage, it floods a limited hop ADVT packet. Upon receiving an ADVT packet, a node responds with a PATCH packet if it is closer to a multicast source node than the ADVT sender node. PoolODMRP [1] enhances PatchODMRP's route repair cost using so called pool nodes. The pool nodes are defined to be neighbor nodes of forwarding nodes and they collect route information by overhearing data transmission. PoolODMRP reduces local route repair overhead with pool nodes' aid. In order to reduce more overhead, the Passive Data Acknowledge ODMRP (PDAODMRP) [2] suggests to use passive acknowledge scheme in PoolODMRP. It removes MAC layer BEACON signal and detects a route breakage by the passive acknowledge while data transmitting. Due to the use of the passive acknowledge scheme, PDAODMRP local route recovery is initiated by a upstream node whereas a downstream node starts a local recovery in previous two protocols.

The Performance Enhanced On-Demand Multicast Routing Protocol (PEODMRP) [22], and the On-Demand Multicast Routing Protocol with Multipoint Relay (ODMRP-MPR) [29] are multicast protocols based on ODMRP, too. PEODMRP reduces control packet overhead via limiting the transmission area of Join Query flooding. If forwarders are set within a time, it responds with Join Reply without Join Query relay. PEODMRP has advantage when multiple sources exist in the same multicast group. ODMRP-MPR minimizes the broadcasting overhead by reducing duplicated packet-forwarding. Only designated one hop neighbor nodes relay packets while packet transmitting. It significant reduces duplicated packet transmission but packet delivery ratio decreases in high mobility network.

2.3 ADMR

ADMR bears resemblance to ODMRP in many aspects. ADMR constructs and refreshes the data forwarding mesh structure by periodic data packet flooding. The route construction procedure is the same as ODMRP: A source floods the network with a data packet and each receiver sends back a Receiver Join packet to the source following the shortest path learned through the data packet flooding. Nodes along the path set up forwarding states and transmit non-duplicated data packets from the source to receivers cooperatively. The differences are, first, ADMR constructs a forwarding structure for each sender while ODMRP constructs a group shared mesh, second, ADMR keeps the route refresh rate rather low, e.g., every several tens of seconds, and tries to detect and repair path breakage locally to cope with frequent but locally contained topological changes, and, third, ADMR prunes unnecessary multicast tree branches using the passive acknowledgement scheme: A node recognizes itself as a valid forwarder by overhearing packets transmitted from its downstream nodes. Each data packet serves as a passive ACK. If a forwarding node does not receive several consecutive passive ACKs, the node prunes itself from the forwarding tree since no sub-tree node exists.

To detect a broken link, the source monitors traffic pattern and sends expected inter-packet time recording in the ADMR header of each data packet. If a node in the tree does not receive a data packet within a multiple of the inter-packet time, it initiates a local sub-tree repair process. Before launching the local repair, a node transmits a Repair Notification packet toward the downstream nodes to inform starting the local route repair. If it receives no packet within a short repair delay, a disconnected node floods a hop-limited Reconnect packet. As receiving a Reconnect packet, a node which is a forwarder in the multicast tree unicasts the packet toward the source. Upon receiving the Reconnect packet, the source sends back a Reconnect Reply packet to the disconnected node through a reverse path that the Reconnect packet passed. Nodes on the reverse path become forwarders for the multicast tree. ADMR's local recovery is two-way handshaking between a disconnected node and a multicast source. It is able to rebuild a new path but ADMR generates high overhead and long latency. Since packets are relayed through multiple hops, the probability of packet error increases in wireless ad hoc network environment. Therefore, ADMR's local recovery is weak in high traffic and high dynamic network. If the local repair has failed, the disconnected node tries to re-join the multicast group using a three-way handshaking method which is the same procedure to a new receiver joining a multicast tree. The disconnected node floods a Multicast Solicitation packet, the source responds it, and the disconnected node returns a Receiver Join packet to build a forwarding state.

2.4 Other Multicast Schemes

Several other MANET multicast routing protocols with their own unique features have been proposed as well in the literature (e.g., [5, 7, 8, 14, 17, 24, 25]). Similar to ODMRP, the Core-Assisted Mesh Protocol (CAMP) [7] uses a mesh structure for data forwarding. However, as differences, an underlying unicast routing protocol is required and "core" nodes are introduced to limit control traffic. The Reservation-Based Multicast (RBM) [5] routing protocol use similar "core nodes" concepts but its uniqueness comes from the fact that it is a combination of multicast, resource reservation, and admission control protocol where users specify requirements and constraints. The Lightweight Adaptive Multicast (LAM) [8] algorithm is another core-based and group shared tree protocol. Similar to other core-based protocols, it suffers from disadvantages of traffic concentration and vulnerability to core failure. The Adaptive Core Multicast Routing Protocol (ACMRP) [17] is an on demand core-based multicast protocol using a mesh structure. It adaptively elects the core node based on the current network topology and group membership. The Protocol for Unified Multicasting through Announcements (PUMA) [24] establishes and maintains a shared mesh by a core node. PUMA uses a receiver initiated approach. Receivers store the list of alternative routes and switch the route when route breakage recognized. The Robust Multicasting in Ad Hoc Network using Trees (ROMANT) [25] is similar to PUMA. A group leader node periodically transmits "Join Announcement" packet and receivers store alternative routes toward the group leader. It, however, uses a tree structure instead of a shared mesh. The Adhoc Multicast Routing Protocol (AMRoute) [14] is a shared-tree protocol which allows dynamic core migration based on group membership and network configuration.

3 Enhanced ODMRP with Motion Adaptive Refresh

In this section we describe details of E-ODMRP, an ODMRP enhancement for mobility adaptive refresh.

3.1 Creating a Forwarding Mesh by Source Initiation

Same as the original ODMRP, a forwarding mesh structure between sources and receivers is initiated by a source. When a new source has data to transmit to a multicast group, it starts with flooding the entire network with the first data packet piggybacking the control/signaling information. We refer to the first data packet as the Join Query packet for convenience hereafter. Upon reception of the first, non duplicate, Join Query packet, every node sets pointers to its upstream node, i.e. the sender of the Join Query packet, and rebroadcasts it. Once the Join Query reaches a receiver, the receiver sends a Join Reply packet back towards the source. The Join Reply is relayed by the intermediate nodes all the way to the source following the pointers set when the Join Query was propagated through the network. The intermediate nodes which have relayed the Join Reply become the forwarding group (or mesh). All nodes in the forwarding mesh are collectively in charge of delivering multicast data to receivers and achieve such goal by transmitting non-duplicate data packet once. A source refreshes the forwarding mesh, i.e., floods the Join Query, on variable-interval schedules and the interval can vary from the prefixed minimum to maximum values. How to adapt the schedule is to be explained in Section 3.4.

The initial creation of the forwarding mesh is the same for ODMRP and E-ODMRP but the nodes' behavior in the mesh is quite different due to the difference in the mesh maintenance mechanism. All nodes in the E-ODMRP mesh, intermediate and leaf nodes, forward received non-duplicated data packets. The leaf nodes' data forwarding is to implement the passive acknowledge-

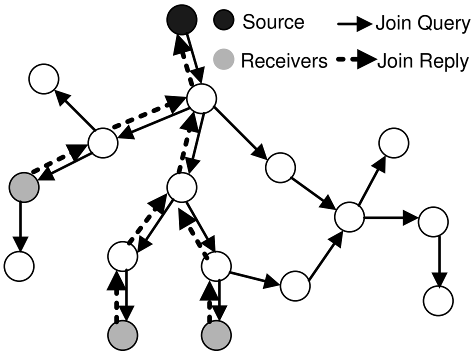

Fig. 1. **E-ODMRP mesh construction: Join Query and Join Reply flow.**

ment (ACK) which is a general mechanism widely used in various MANET protocols for various reasons. In the data packet's header, there is a field indicate the packet sender's upstream node. By overhearing every data packet transmission and from the field in the packet, a node can know whether its transmission was a success and/or whether it is a valid forwarder, that is, some node is actually receiving data from it. Forwarders in the E-ODMRP mesh do not have the forwarder life-time whereas ODMRP's forwarder has a timeout which is a parameter usually set to 3 times the refresh interval. In ODMRP forwarder nodes discharge themselves when the forwarding state expires by a timeout. In E-ODMRP, intermediate nodes forward data packets as long as downstream receivers exist otherwise they prune themselves. Nodes realize whether receivers exist in the downstream sub-tree using the passive ACK mechanism.

3.2 Receiver Joining

When a receiver wants to join a multicast group, it performs a local search to graft onto the existing multicast mesh. The receiver broadcasts a Receiver Join packet first with limited Time-To-Live (TTL), e.g., 1. When a Listener node, defined to be a neighbor of any forwarder or receiver nodes, receives a Receiver Join packet, it sets itself up as a Temporary Forwarder and immediately starts forwarding data packets. While Temporary Forwarders forward next several non-duplicate packets, the receiver chooses one of them as a regular forwarder being part of the forwarding group. Other Temporary Forwarders clear their status and go back to Listeners. The Receiver Join packet's TTL is 1 in E-ODMRP, but disconnected node can grab into a forwarding mesh that is 2 hops away due to the definition of the Listener. In Figure 2 (a), node A wants to join the multicast mesh and transmits a Receiver Join packet. Upon receiving the Receiver Join, Listeners, node B and C, relay next several packets. In Figure 2 (c), node B becomes a Forwarder and node A is connected to the forwarding mesh. Therefore, E-ODMRP's Local Recovery scheme performs the same effect as other protocols' local recovery that a recovery control packet travels up to 2 hops. If such a local search fails, the disconnected receiver floods a Refresh Request packet. Sources, if exist, will receive the packet and refresh the multicast forwarding group by flooding with a Join Query packet. When multiple receivers simultaneously issue Refresh Request floods, huge traffic overhead occurs through the network. It may degrade protocol performance to waste resources and block other traffics. To prevent this harmful network wide flooding, E-ODMRP nodes relay only one of such Refresh Request packets when multiple receivers broadcast Refresh Request packets in a short time frame, i.e., a minimum refresh time. In other words, a node relays the first Refresh Request packet and drops any other Refresh Request packets arrived within a short time period even though they have been generated by different nodes. If timeout occurs without receiving any data packet, the receiver re-floods the Refresh Request. If timeout occurs again, it waits for the next Join Query without flooding since it means that the network is completely divided

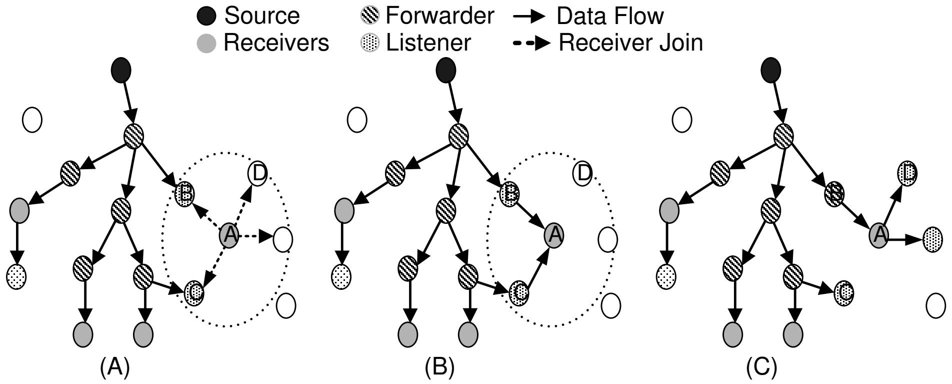

Fig. 2. **(a) Illustration of receiver join/local recovery process. (Dotted circles denote the same distances.) Node A wants to join the forwarding mesh, so node A floods the Receiver Join packet. Node B, and C hear it and set themselves up as Temporary Forwarders. (b) Now node B and C transmit a packet to the node A, but node D cannot. (c) Node A chooses the node B as a upstream node. Node C goes back to a Listener due to Temporary Forwarder timeout. The node D becomes a Listener since it overhears data transmission from the node A.**

or there is no source. If a forwarder or a listener wants to join a multicast group, it simply updates its status as a receiver without the joining process.

3.3 Detecting a Link Break and a Local Recovery

An intermediate node or a receiver can be disconnected from the mesh due to mobility. For unicast transmission, detection of a broken route is fairly easy and provided by the MAC layer. If a node does not return a MAC layer ACK, the link incident on the node is considered to be broken. But in multicast, a link break should be detected in different ways since MAC broadcast has no ACK. ADMR monitors the traffic to detect malfunctioning links. We take a similar approach. Assuming that traffic is frequent enough to serve as indicator for any route break, each source estimates its own inter packet arrival time and informs receivers by recording it in Join Query packets. Based on source's value, each node calculates and updates own inter packet arrival time until receiving the next Join Query. If a node in the mesh does not receive any data during a multiple of the packet arrival interval e.g., 5 times arrival interval in our simulation, the node considers itself to be detached from the mesh and performs the recovery procedure. It is the same as the receiver join process except sending a Dummy packet. When the node receives a Receiver Join packet from a parent node, it generates a Dummy packet and transmits to a sub-tree to prevent recovery explosion. Nodes received the Dummy packet wait for a next packet without a local recovery. However, they start the local recovery, if they have timeout without receiving a new packet. A source generates the Dummy packet when no packet is coming from the application. All nodes in the mesh wait without the local recovery. If timer expires again, the source re-sends the Dummy packet. Upon receiving the second Dummy packet from the source, all nodes in the multicast group realize that the data transmission ends and they remove information related to the multicast group by the next timeout.

3.4 Adaptive Refresh

If the local search fails in recovery, a Refresh Request flooding occurs. The initiator of the Refresh Request estimates the "route lifetime" which is the time difference between the two events: route establishment, e.g., reception of Join Reply, and detection of the route breakage. The route lifetime is the maximum time period that an established forwarding mesh stays functioning for all the receivers under the current network dynamics. The value is recorded in a Route Refresh packet and is passed to the source via network wide flooding. The multicast source adjusts its Join Query flooding rate when it receives a Refresh Request packet. The multicast source changes the refresh rate to the inverse of route lifetime less a very small constant value and floods a Join Query packet. The rationale is that E-ODMRP should refresh the forwarding mesh before it breaks. Since the refresh rate is adjusted to the “route lifetime” estimated by the node that actually has experienced disconnection, the forwarding mesh will be reconstructed before it breaks unless the node mobility condition changes. A sender linearly and slowly increases, say by a half of the refresh interval, the refresh interval if the interval is not adjusted, i.e., not received any Refresh Request packet, by the next “scheduled” route refresh. In essence, this is a linear increase, sudden decrease refresh scheme where the source attempts to reduce overhead by slowing down the refresh updates. If there is no maximum limit in the refresh rate, the sudden decrease scheme leads short refresh interval in high mobility situation. The unnecessary short interval wastes channel bandwidth and degrades network performance. In E-ODMRP, network dynamics can be kept up by the local recovery in general if the node mobility is not too high. The local recovery, however, cannot keep the shortest path from the source to receivers and it may increase overhead while decreasing efficiency. Thus mesh rebuilding helps network efficiency in case of low mobility as well as high mobility. We recommend the minimum refresh interval to be 3 seconds the same as the ODMRP’s refresh interval and maximum interval to be 30 seconds.

3.5 Passive ACK and Pruning

During the route refresh and the recovery period, the forwarder mesh becomes larger since new forwarders are emerged. Though redundant data forwarding leads to high delivery ratio, it also generates high overhead that may degrade performance. Pruning removes unnecessary data forwarding using the passive ACK scheme. As mentioned earlier, in the every data packets, the address of the next-hop to the upstream direction is written. Each node records upstream and downstream node's addresses in its Multicast Routing Table to be explained in the next section that is updated and maintained during the route refresh and the recovery process. Intermediate nodes overhear packet transmission from the downstream nodes so that they can confirm whether their transmission is valid by checking the recorded address in the packet. Thus each sent packet serves as a passive ACK eliminating any explicit control packet. If a forwarder misses several passive ACKs continuously, it prunes itself from the mesh. Though the passive ACK removes unnecessary forwarding, the overhead may be still high since all nodes including the leaf nodes in the mesh forward packets. To reduce the overhead, a passive ACK suppression technique is employed in the leaf nodes. The leaf nodes forward packets after short delay whereas intermediate nodes forward as soon as receiving packets. If a leaf node receives duplicated packets during the short delay, it skips sending a passive ACK for this packet since another receiver may send a passive ACK or the leaf node may change a upstream forwarder due to mobility.

3.6 Multiple Source Case

If there are multiple sources, nodes including sources in the mesh calculate and update own packets arrival interval based on received packet from multicast group sources. For example, a node receives 4 packets per second from a source A and 6 packets per second from a source B, it considers that the multicast group sends 10 packets per second and inter packet time is 0.1 second.

If a node does not receive packets during a multiple of packet arrival interval, it tries to reconnect to the mesh. Most likely, it has become detached from the mesh since a node does not receive a packet from ALL Senders. If the receiver receives a packet from subset of Senders (but not all), it does nothing. This situation is detected and recovered by the senders. Senders detect a broken route in the same way as receivers doing. If a sender times out on ALL other senders, it assumes that it has become detached and performs the Join Query flood. If the sender times out on a subset of the Senders, it assumes the network has become partitioned. The highest number sender in the partition is responsible for issuing the Join Query. This will suffice to repair the mesh. When there is one sender, it does not try to detect a broken route since receivers and forwarders do local recovery for link break. However, in the multiple sources case, sources have to monitor the route breakage to find detaching themselves from the mesh since receivers and forwarder do not start the local recovery as long as time out from all senders.

In original ODMRP, the overhead increases rapidly as the number of sources increasing since each source independently floods the Join Query. However, since all sources and receivers in the same multicast group share the same mesh structure, it can be created and maintained if only one source floods the Join Query and other sources and receivers respond with a Join Reply. E-ODMRP employs this flooding aggregation technique to prevent overhead increasing. When the source listens the flooding from another source in the same group, it responds with a Join Reply and resets the route refresh schedule. When a new source has data to transmit to a multicast group, it starts sending packets if it involves in the multicast group as a receiver, a forwarder, or a listener. Otherwise, it floods Join Query. When sources receive a Refresh

Request, they try to flood after random backoff.

3.7 Data Structure

Nodes in the network running E-ODMRP need to maintain the following data structures.

Multicast Routing Table : Each node creates and maintains a Routing Table. It stores the multicast group address, the destination address (i.e., the source of the multicast group), the addresses of the upstream and downstream nodes, and minimum hop count from the source. A node updates or enters a new entry into the Routing Table when receiving a non-duplicated Join Query packet or sensing a new parent node (i.e., the next hop to the source).

Membership Table : The Multicast group information is stored in the Membership Table that is created and maintained by each node. The Membership Table contains the multicast group address, node's status (source, receiver, forwarder, and listener), inter-packet time, and the time when the last Join Query received.

Message Cache : The message cache is generated and maintained by each node to detect duplicated packets. When a node receives a control or data packets, it records multicast group address, source address and sequence number of the packet in the Message Cache. If the entry of received packet exists in the Message Cache, the node drops the packet; otherwise processes it. Multicast group and source address is 4 bytes and the sequence number is only 2 bytes due to the IPv4 header. Total size of the Message Cache is small since each entry is 10 bytes big. Since entries in the message cache are not stored permanently, old entries are removed to keep the table size small. FIFO (First In First Out) or LRU (Least Recently Used) can be employed to prevent the size of the Message Cache growing.

4 Simulation Results

In this section, we study the performance of E-ODMRP and compare it with ODMRP, PatchODMRP, and ADMR. To this end, we conducted a set of ns-2 [16] simulations.

4.1 Simulation Settings

We implemented E-ODMRP and PatchODMRP and used existing ADMR and ODMRP codes publicly available from Monarch Project homepage [15]. Simulation settings are as follows: 100 nodes randomly distributed over the 1200m by 800m field; 1 multicast group with 1 source and 20 receivers unless otherwise specified; constant bit rate traffic of 4 packet/s and 512 byte/packet; 900 seconds of simulation time; no packet/channel error model. We use five metrics: Packet Delivery Ratio is the fraction of packets received averaged over all receivers; Total Packet Transmitted per Data Packet Delivered is the total number of data and control packets generated by the network divided by the total number of data packets actually received and we refer to this metric as Packet Overhead for convenience hereafter; Total Byte Transmitted per Data Byte Delivered is the total byte of data and control packets generated by the network divided by the total byte of data packets actually received and we refer to this metric as Byte Overhead for convenience hereafter; Total Control Packets Transmitted is the total number of transmitted control packets from all nodes during simulation time; Average Route Refresh Interval is the averaged period between route refresh. All numbers are averaged over 10 simulation runs. Protocol parameter settings are as follows. ODMRP's refresh interval is 3 seconds and forwarder's lifetime is 3 times the refresh interval, i.e., 9 seconds. PatchODMRP's refresh interval is 240 seconds and forwarder's lifetime is 3 times the refresh interval, i.e., 720 seconds. MAC layer BEACON interval in PatchODMRP is 3 seconds and a local recovery control packet, an ADVT packet, travels up to 2 hops. ADMR's periodic data flooding interval is 30 seconds. When bootstrapping, the interval steps up from 5 seconds to 10 seconds and then 30 seconds. The number of lost packets triggering local repair is 3, local repair packet's TTL is 2, and the Receiver Join packet retransmission limit is 3. E-ODMRP's minimum and maximum refresh interval is 3 seconds and 30 seconds respectively and the initial interval is 30 seconds. The number of lost packets triggering local recovery is 5, local recovery TTL is 1, and retransmission limit is 2. The source changes the refresh rate to the inverse of route lifetime if it receives a Route Request packet; otherwise the source linearly increases the refresh interval by a half of the refresh interval until reach the maximum value. Finally for the mobility model, the Random Waypoint model [3] was used. Nodes in the area select some destination and move to there at a random speed uniformly chosen from the minimum speed, 1 m/s, to the selected maximum speed. An average node speed in the Random Waypoint model is a half of the maximum node speed since node speeds are uniformly distributed. However, Yoon et al. [28] has presented that the average speed may lower than a half of the maximum node speed since a

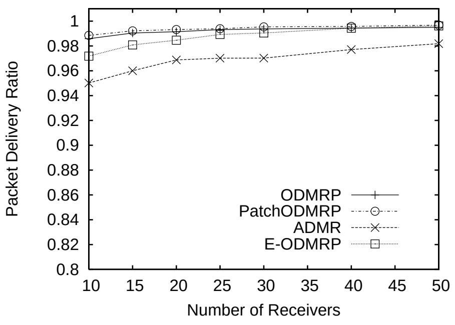

Fig. 3. **Packet Delivery Ratio with Varying Number of Receivers (1 group, 1 source, 0 second pause time, 20 m/s maximum speed).**

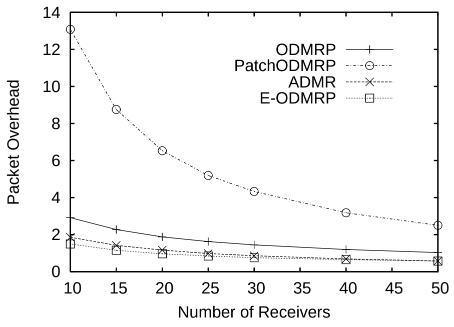

Fig. 4. **Packet Overhead with Varying Number of Receivers (1 group, 1 source, 0 second pause time, 20m/s maximum speed).**

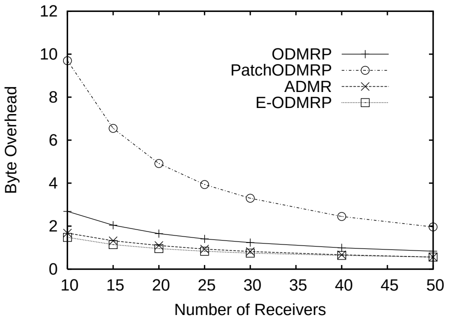

Fig. 5. **Byte Overhead with Varying Number of Receivers (1 group, 1 source, 0 second pause time, 20 m/s maximum speed).**

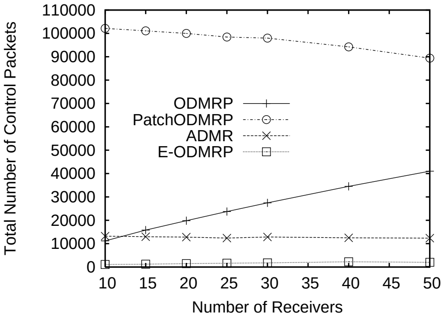

Fig. 6. **The Number of Control Packets with Varying Number of Receivers (1 group, 1 source, 0 second pause time, 20 m/s maximum speed).**

node that chooses a far-away destination with a slow speed needs a long time to reach the destination. As the simulation time passes, more nodes may be trapped to slow trips and thus the average speed is slowly decreasing. Therefore, we mention the maximum node speed instead of the inaccurate average node speed.

4.2 Varying Number of Multicast Receivers

Figure 3, 4, 5, and 6 illustrate E-ODMRP's performance with varying number of receivers in the multicast group. As predicted, in E-ODMRP the delivery ratio degrades (but only minimally) with respect to ODMRP while the overhead is drastically reduced. E-ODMRP shows the best delivery ratio with 50 receivers. ADMR has the lowest packet delivery ratio among four protocols. PatchODMRP's delivery ratio is as high as ODMRP's, but its overhead is by far worse than others.

In figure 4, the packet overhead decreases in the number of receivers in all four protocols. In the original ODMRP, the number of data packet forwarded is higher than ADMR and E-ODMRP due to periodic flooding with a short interval and a long forwarder lifetime. However, PatchODMRP's data forwarding is much higher than ODMRP's since the forwarder lifetime is overly longer than ODMRP's. In PatchODMRP, more than 90% of nodes in the network participate data forwarding, but it is not over 40% in ODMRP and less than 30% of nodes forward data packets in ADMR and E-ODMRP. Even though this excessively high overhead keeps a high delivery ratio, PatchODMRP wastes bandwidth and the network may collapse when data traffic increases. E-ODMRP's overhead is inherently low with the long refresh interval and pruning based on passive ACKs. Similarly, ADMR executes periodic network wide flooding for forwarding tree reconstruction at a slow rate and pruning. But a more elaborate recovery process results in ADMR's higher overhead than E-ODMRP's (It is difficult to distinguish E-ODMRP and ADMR overhead in figure 4 and figure 5, but E-ODMRP overhead is less than ADMR's at most 0.4.). As mentioned in section 2.3, the a Repair Notification packet is transmitted first toward the downstream tree and then the disconnected node floods a hop-limited Reconnect packet for two-way handshaking with the source. Thus at least several copies of the control packets are transmitting during the local recovery in ADMR. On the other hand, E-ODMRP broadcasts one Receiver Join packet only to its one hop neighbors. Therefore, the number of control

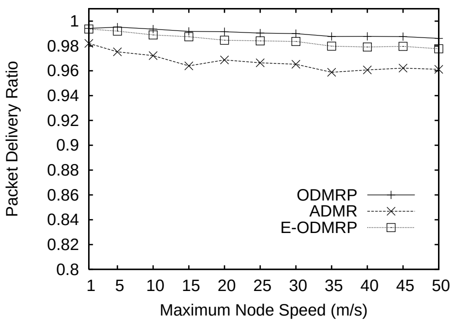

Fig. 7. **Packet Delivery Ratio with Varying Maximum Node Speed (1 group, 1 source, 20 receivers, 0 second pause time).**

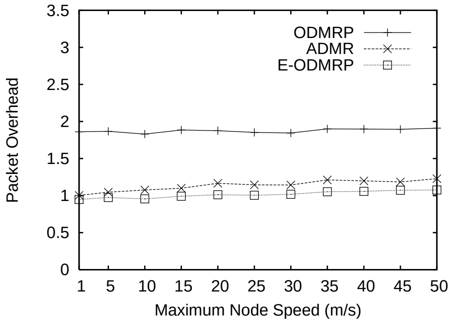

Fig. 8. **Packet Overhead with Varying Maximum Node Speed (1 group, 1 source, 20 receivers, 0 second pause time).**

packets in E-ODMRP is much smaller than that of ADMR even if they are about the same in terms of the total number of data packet transmissions. As the number of receivers increases, the ratio of the number of received packets over the number of generated packets increase. If the number of receivers increases, then receiver density becomes higher and thus the number of receivers that receive a packet transmitted from the same node increases due to broadcast nature of wireless medium. Therefore, the overhead decreases and E-ODMRP's and ADMR's overhead are lower than 1 when there are more than 30 receivers. The packet overhead shown in Figure 4 and figure 5's byte overhead look similar to each other. The values are different but the superiority and inferiority among the protocols is the same in the graphs. The byte overhead values are lesser than packet overhead's because the size of a control packet is smaller than a data packet. Since the two graphs show similar patterns and the packet overhead is claimed to show protocol's channel access efficiency in wireless contention medium in the RFC 2501 [6] document, we only show the packet overhead graphs hereafter.

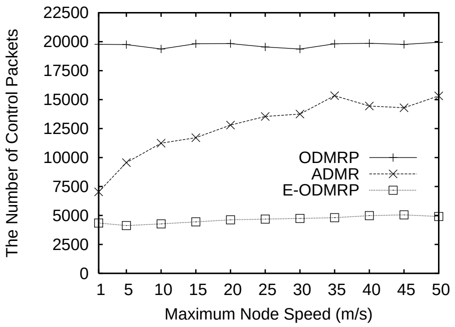

Fig. 9. **The Number of Control Packets with Varying Maximum Node Speed (1 group, 1 source, 20 receivers, 0 second pause time).**

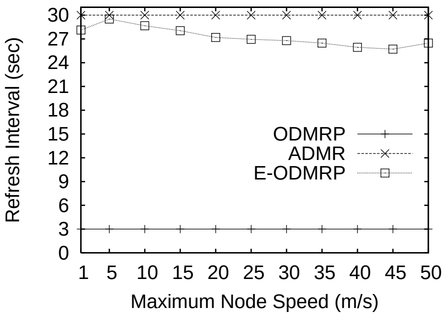

Fig. 10. **Refresh Interval with Varying Maximum Node Speed (1 group, 1 source, 20 receivers, 0 second pause time).**

4.3 Varying Node Speed

Figure 7 and 8 compares E-ODMRP's, ODMRP's, and ADMR's performance in different mobility conditions, i.e., varying maximum node speed. The maximum node speed varies from 1 m/s (3.6km/h) to 50 m/s (180 km/h). We exclude PatchODMRP's results from the graphs since its overhead is too high such that if it were included in the graphs other protocols' performance cannot be distinguished from each other due to the scaling effect. The packet delivery ratio of PatchODMRP is as high as ODMRP's but PatchODMRP's packet overhead is more than 3 times bigger than ODMRP's. Compared to ODMRP, E-ODMRP achieves just about the same delivery ratio with minimal difference while incurring only a half of overhead. ADMR's packet delivery ratio diverges from the other two as the mobility gets greater, lower than 95% when the maximum node speed is 50 m/s. The gap between ADMR's and E-ODMRP's packet overhead is not big but stays similar in all cases.

Figure 9 shows the total number of control packets transmitted in the network and figure 10 compares protocol's route refresh intervals with varying node speed. Fixed refresh interval, e.g., 3 seconds, and no local recovery in ODMRP lead the number of control packets to stay flat in figure 9. ADMR's refresh interval is fixed as 30 seconds but it generates several control packets when a local recovery process, e.g., Repair Notification, Reconnect, Reconnect Reply packets and as network dynamic escalates, the number of control packets increases rapidly due to frequent route breakages. On the other hand, E-ODMRP induces significantly less control packets compared to the other two protocols in all cases. Its dynamic refresh reduces the network wide flooding and a simple local recovery scheme cuts down control packet transmissions. It is interesting to see that in figure 9, E-ODMRP's number of control packets stays nearly constant from 1 m/s to 50 m/s maximum node speed. In figure 10, we can find that E-ODMRP route refresh adaptively changes based on node mobility. As node speed escalates, Refresh Request increases and as consequences E-ODMRP's dynamic refresh interval declines.

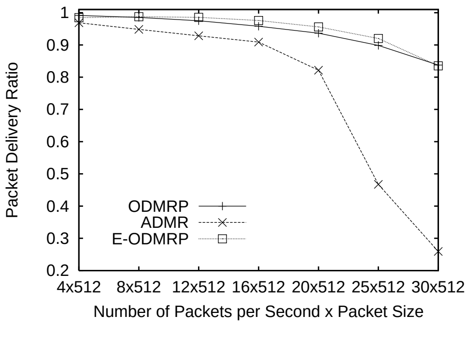

Fig. 11. **Packet Delivery Ratio with High Data Rate (1 group, 1 source, 20 receivers, 0 second pause time, 20 m/s maximum speed).**

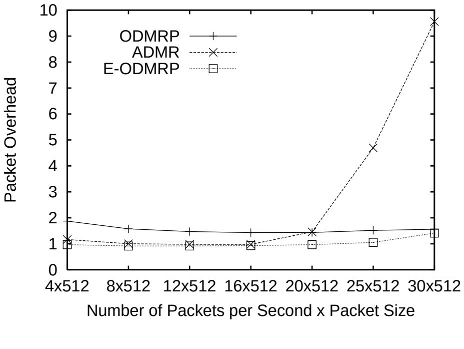

Fig. 12. **Packet Overhead with High Data Rate (1 group, 1 source, 20 receivers, 0 second pause time, 20 m/s maximum speed).**

4.4 Varying Data Traffic

Figure 11 and 12 present protocols' performance with varying traffic pattern. The packet sending rate varies from 4 packets per second to 30 packets per second and each packet is 512-byte sized. Interestingly, E-ODMRP outperforms other three protocols in high packet sending rate situation. Four protocols' delivery ratios decrease by significantly different factors with increasing sending rate. There is a crossing point at 8 packet/s point between E-ODMRP and ODMRP. As mentioned previously, PatchODMRP delivery ratio significantly decreases as data rate increases. ADMR's delivery ratio and packet overhead slopes suddenly change at 16 packet/s point, which is because the number of control packets in ADMR starts to increase rapidly at that point. As traffic gets intense, congestion occurs and packets are dropped but since ADMR cannot distinguish between route breakage and congestion, ADMR recognizes the situation as route breakage and initiates the local route recovery by transmitting control packets for recovery which makes the situation worse and the network to collapse. E-ODMRP cannot distinguish between route breakage and congestion either; but it does not incur the control packet implosion in case of congestion owing to its simple recovery scheme. E-ODMRP's low control overhead provides more network bandwidth to applications. This extra bandwidth makes packet delivery ratio stay high.

4.5 Random Packet Error

In Figure 13 and 14, we compare three protocols in presence of random packet errors, i.e., lossy wireless channels. We exclude PatchODMRP's results in the graphs again (and thus the discussion of the results in this section) since the trend is the same as figure 3 and 4 and we intend to focus on the performance differences among E-ODMRP, ODMRP, and ADMR in the graphs. To simulate random errors, every node is enforced to drop the packets randomly and uniformly with probability 0.1. The number of receivers varies from 10 to 50. As predicted, the packet delivery ratios are decreased compared to the case without random errors shown in figure 3. Among the three protocols, ODMRP shows the least degradation due to its rich forwarding redundancy and ADMR is the worst: its ratio drops from 95% to 91% when the number of receivers is 10. E-ODMRP maintains performance degradation within 2%. The packet overhead is a little higher than the overhead without random packet drop as shown in figure 4, but the difference is not significant, less than 1%, for all three protocols.

4.6 Varying Number of Multicast Sources

Figure 15 and 16 show four protocols' performances with varying number of sources in one multicast group. The number of sources changes from 1 to

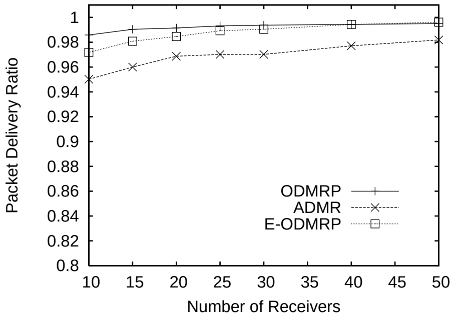

Fig. 13. **Packet Delivery Ratio with Random Probabilistic Packet Drop (Probability 0.1, 1 group, 1 source, 0 second pause time, 20 m/s maximum speed).**

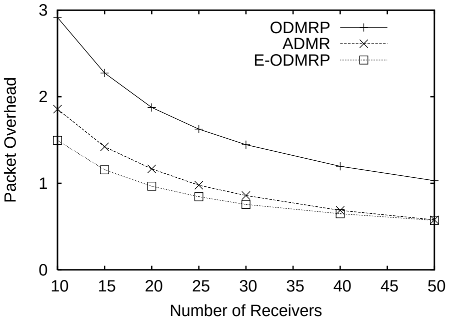

Fig. 14. **Packet Overhead with Random Probabilistic Packet Drop (Probabilty 0.1, 1 group, 1 source, 0 second pause time, 20 m/s maximum speed).**

6 in the multicast group and 20 nodes out of 100 nodes are multicast receivers. Nodes move at the speed of at most 20 m/s. We observe that, in figure 15, the delivery ratios decrease by different factors for the different protocols, as the number of sources increases. The different trends of the four protocols' delivery ratios are explained by the differences among the four protocols' overhead trends shown in Figure 16. ODMRP's packet delivery ratio gradually decreases as the overhead increases in the number of sources. In ODMRP, since each source independently floods a Join Query packet periodically for route refreshing, overhead increases as the number of sources increases. Higher overhead, meaning more in-transit packets, clearly makes destructive contribution to packet delivery since it causes more collisions and congestions. PatchODMRP's delivery ratio decreases steeply as the number of sources increases due to its inherent high overhead. E-ODMRP outperforms other three protocols with respect to packet delivery ratio when there are more than three sources. This is again explained by E-ODMRP's near-flat line in the overhead graph, see figure 16. ADMR's performance suddenly starts dropping after four sources and it shows the worst performance with six sources. Previously, we pointed out ADMR's susceptibility to network congestion. The

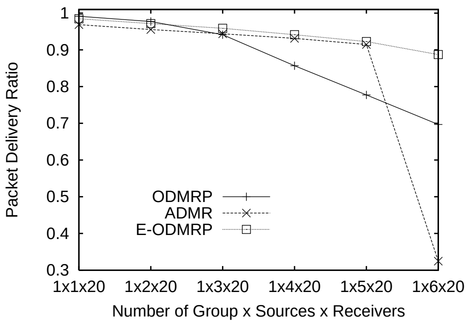

Fig. 15. **Packet Delivery Ratio with Multiple Sources (1 group, 1 ∼ 6 sources, 20 receivers, 0 second pause time, 20 m/s maximum speed).**

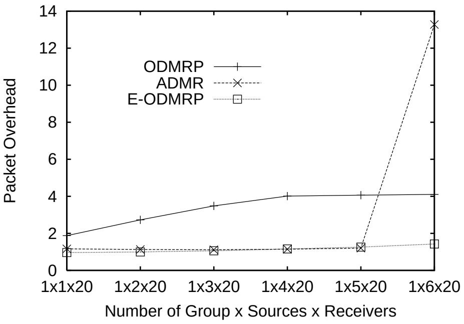

Fig. 16. **Packet Overhead with Multiple Sources (1 group, 1 ∼ 6 sources, 20 receivers, 0 second pause time, 20 m/s maximum speed).**

results in Figure 15 and 16 shows that this problem is worsened when there are multiple sources. The increased overhead is due to the fact that ADMR maintains a multicast tree for each source. Since each multicast member has to recover a broken link for each source, in the worst case a receiver initiates the local recovery as many times as the number of sources.

5 Conclusion

In this paper, we have presented an enhanced version of ODMRP with motion adaptive refresh, namely E-ODMRP. It performs the periodic refresh at a rate dynamically adapted to the nodes' mobility. Another feature is the "unification" of local recovery and receiver joining process. On joining or upon detecting a broken route, a node performs a local search to graft to the forwarding mesh proactively. Simulation results show that E-ODMRP reduces the packet overhead by up to 50% yet keeping similar packet delivery ratio as the original ODMRP. The simulation results also confirm that E-ODMRP outperforms ADMR. The utility of the linear increase, sudden decrease refresh scheme is not mathematically analyzed nor proven in the paper. One of our immediate future works is to give such a theoretical analysis or proof. One of our approaches is to view the refresh rate adjustment scheme as a feedback system and apply techniques of Control Theory.

Acknowledgement

Research was sponsored by the U.S. Army Research Laboratory and the U.K. Ministry of Defence and was accomplished under Agreement Number W911NF-06-3-0001. The views and conclusions contained in this document are those of the author(s) and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Army Research Laboratory, the U.S. Government, the U.K. Ministry of Defence or the U.K. Government. The U.S. and U.K. Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

References

[1] S. Cai, X. Yang, The performance of PoolODMRP, in: Proc. The 6th IFIP/IEEE International Conference on Management of Multimedia Networks and Service, MMNS 2003, Belfast, Northem Ireland, 2003, pp. 90-101.

[2] S. Cai, X. Yang, L. Wang, PDAODMRP: An extended PoolODMRP based on passive data acknowledgement, Journal of Communications and Networks, 6(2004) (4), pp.362-375.

[3] T. Camp, J. Boleng, V. Davies, A servey of mobility models for ad hoc network research, Wireless Communication and Mobile Computiong (WCMC): Special Issue on Mobile Ad Hoc Networking, 2(2002) (5), pp.483-502.

[4] C.-C. Chiang, M. Gerla, L. Zhang, Forwarding group multicast protocol (FGMP) for multihop, mobile wireless networks, Cluster Computing, 1(1998) (2), pp.187-196.

[5] M. Corson, S. Batsell, A Reservation-Based Multicast (RBM) routing protocol for mobile networks: initial route construction phase, ACM/Baltzer Wireless Networks, 1(1995) (4), pp.61-81.

[6] S. Corson, K. Macker, Routing protocol performance issues and evaluation considerations, RFC 2501, 1999.

[7] J.J. Garcia-Luna-Aceves, E. Madruga, The core assisted mesh protocol, IEEE Journal on Selected Areas in Communications, 17(1999) (8), pp.1380-1394.

[8] L. Ji, M. Corson, A lightweight adaptive multicast algorithm, in: Proc. 1998 IEEE Global Telecommunications Conference, GLOBECOM 1998, Sydney, Australia, 1998, pp.1036-1042.

[9] J. Jetcheva, D. Johnson, Adaptive demand-driven multicast routing in multihop wireless ad hoc networks, in: Proc. The 2nd ACM international symposium on Mobile ad hoc networking & computing, MobiHoc '01, Long Beach, CA, USA, 2001, pp. 33-44.

[10] M. Lee, T.Kim, PatchODMRP: an ad hoc multicast routing protocol, in: Proc. The 15th International Conference on Information Networking, ICOIN '01, Oita, Japan, 2001, pp. 537-543.

[11] S. Lee, W. Su, M. Gerla, On-Demand multicast routing protocol, in: Proc. IEEE Wireless Communications and Networking Conference, WCNC '99, New Orleans, LA, USA, 1999, pp.1298-1302.

[12] S. Lee, W. Su, M. Gerla, Ad hoc wireless multicast with mobility prediction, in: Proc. Eight International Conference on Computer Communications and Networks, ICCCN, Boston, MA, USA, 1999, pp.4-9.

[13] S. Lee, W. Su, M. Gerla, On-Demand multicast routing protocol in multihop wireless mobile networks, ACM/Kluwer Mobile Networks and Applications, 7(2000) (6), pp.441-453

[14] M. Liu, R. Talpade, A. McAuley, Amroute: Adhoc multicast routing protocol, University of Maryland CSHCN Technical Report 1999-1, College Park, MD, 1999.

[15] The Monarch Project. Monarch Project: mobile networking architectures. Available at *http://www.monarch.cs.rice.edu/*.

[16] Network Simulator ns-2. *http://www.isi.edu/nsnam/ns/*.

[17] S. Park, D. Park, Adaptive Core Multicast Routing Protocol, Wireless Networks, 10(2004) (1), pp.53-60.

[18] T.S. Rappaport, Wireless Communications: Principles and Practice, Prentice Hall, Upper Saddle River, NJ, 1995.

[19] E. Royer, C. Perkins, Multicast operation of the ad-hoc on-demand distance vector routing protocol, in: Proc. The 5th annual ACM/IEEE international conference on Mobile computing and networking, MobiCom '99, Seattle, WA, USA, 1999, pp.207-218.

[20] P.M. Ruiz, A.F. Gomez-Skarmeta, Reducing data-overhead of mesh-based ad hoc multicast routing protocols by steiner tree meshes, in: Proc. 2004 First Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks, SECON '04, Santa Clara, CA, USA, 2004 pp.54-62.

[21] P. Sinha, R. Sivakumar, V. Bharghavan, MCEDAR : Multicast Core-Enxtraction Distributed Ad hoc Routing, in: Proc. IEEE Wireless Communications and Networking Conference, WCNC '99, New Orleans, LA , USA, 1999, pp.1313-1317.

[22] B. So, H. Jeon, J. Lee, PEODMRP: Performance Enhanced On-Demand Multicast Routing Protocol, in: Proc. The 8th Joint Conference on Communications & Informations, JCCI '04, MooJoo Resort, Korea, 2004,.

[23] C-K. Toh, G. Guichala, S. Bunchua, ABAM: On-demand associativity-based multicast routing for ad hoc mobile networks, in: Proc. IEEE Vehicular Technology Conference, VTC 2000, Tokyo, Japan, 2000, pp.987-993.

[24] R.Vaishampayan, J.J. Garcia-Luna-Aceves, Efficient and robust multicast routing in mobile ad hoc networks, in: Proc. 2004 IEEE International Conference on Mobile Ad-hoc and Sensor Systems, MASS, Fort Lauderdale, FL, USA, 2004, pp.304-313.

[25] R.Vaishampayan, J.J. Garcia-Luna-Aceves, Robust tree-based multicasting in ad hoc networks, in: Proc 2004 IEEE International Conference on Performance,

Computing, and Communications, IPCCC, Phoeniz, AZ, USA, 2004, pp.647-652.

[26] K. Viswanath, K. Obraczka, G. Tsudik, Exploring mesh and tree-based multicast routing protocols for MANETs, IEEE Transaction on Mobile Computing, 5(2006) pp.28-42.

[27] C. Wu, Y. Tay, C. Toh, AMRIS : A multicast protocol for ad hoc wireless networks, in: Proc. of IEEE Military Communications, MILCOM '99, Atantic City, NJ, USA, 1999, pp.25-29.

[28] J. Yoon, M. Liu, B. Noble, Random waypoint considered harmful, in: Proc. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications, INFOCOM 2003, San Francisco, CA, USA, 2003, pp.1312-1321.

[29] Y. Zhao, L. Xu, M. Shi, On-deman multicast routing protocol with multipoint relay (ODMRP-MPR) in Mobile Ad-Hoc Network, in: Proc. International Conference on Communication Technology, ICCT 2003, Beijin, China, 2003, pp.1295-1300.