A Reliable Multicast Algorithm for Mobile Ad hoc Networks

Thiagaraja Gopalsamy
Altera Corp.
tgopalsa@altera.com

Mukesh Singhal
University of Kentucky
singhal@cs.uky.edu

D. Panda and P. Sadayappan
Ohio State University
{panda,saday}@cis.ohio-state.edu

Abstract: A reliable multicast algorithm, called RMA, for mobile ad hoc networks is presented that is based on a new cost criterion, called link lifetime, for determining the optimal path between a pair of nodes. The algorithm has the characteristics of using an undirected graph for its routing operations rather than a fixed structure like a tree or a mesh. Previously proposed routing metrics for mobile ad hoc networks were designed for use in wired environments, where link stability is not a concern. We propose a new metric, called the lifetime, which is more appropriate for mobile ad hoc networks. The lifetime metric is dependent on the predicted future life of the link under consideration. We developed a simulator for the mobile ad hoc networks, which is portable and scalable to a large number of nodes. Using the simulator, we carried out a simulation study to analyze the effectiveness of the routing metrics and the performance of the proposed reliable multicast algorithm. The simulation results show that the lifetime metric helps achieve better performance in mobile ad hoc environments than the hop count metric.

1. Introduction

Mobile ad hoc networks (MANET) are a special case of mobile network without any fixed backbone network to support them and provide connectivity or to perform state maintenance. The mobile hosts themselves perform all the routing and state maintenance operations. The nodes in the network also have lower processing capabilities than their stationary counterparts. The bandwidth of the wireless medium is less than wired media. Thus routing in ad hoc networks poses a challenging research problem. The standard routing protocols used in fixed networks or infra-structured mobile networks can't be used in mobile ad hoc networks. The main applications of mobile ad hoc networks are in emergency rescue operations and in battlefields. The most characteristic operation in these areas is multicast, where messages are sent from one node to multiple recipients. Thus multicast routing is a challenging research problem [2-12]. There are several requirements posed on the multicast algorithm by the mobile ad hoc network environment. The existing multicast algorithms do not satisfy all of the requirements, e.g., reliability of message delivery.

Routing is one of the most contentious and important issues in mobile ad hoc network environments. Routes are usually multi-hop and this necessitates the presence of a unified routing mechanism across the whole network. The routing mechanism has to maintain the routing structure and update it whenever changes occur in the system. The expectations on the routing mechanism are more, in that it is expected to be robust enough to handle all possible changes in the system topology and guarantee near optimal routes between any source destination pair. The resources provided to the routing mechanism on the other hand are limited, i.e., nodes with limited life and processing capabilities and the limited bandwidth of the wireless medium. These conflicting sets of resources and demands make routing in ad hoc networks a very challenging problem.

Multicast routing is a subset of the routing problem. The advent and development of multi-user applications have led to the need for reliable, cost effective multicast mechanisms. The majority of the ad hoc network applications are in the multi-user domain. Hence the issue of multicast routing becomes an object of interest as well as concern. All the problems associated with routing are applicable to multicasting as well, but the demands are much higher, in the sense that multi-point delivery should be guaranteed.

Multicast operations in mobile networks are generally used for dissemination of important and confidential information. The multicast operation is also used as a means for synchronization of operations between various mobile hosts. Multicast algorithms are hence expected to ensure a reliable message delivery and in most of the cases, the source of message transmission is to be informed of the success of the message delivery. In MANET environments, reliability is of much higher concern. Existing multicast algorithms for mobile ad hoc networks do not provide reliability. Our algorithm RMA concentrates on reliable message delivery.

RMA ensures reliability through the use of Acknowledge messages from the destination to the source. These acknowledge messages also contribute to the reverse path maintenance in the routing tables. The source is also entrusted with the task of ensuring a reliable message delivery, through retransmissions in case of failure to get an acknowledge message back within a pre-specified time. The novelty of our approach is that we propose a new cost factor (discussed in the next section) for determining the optimal path between a pair of nodes.

Proceedings of the 22 nd International Conference on Distributed Computing Systems (ICDCS'02)
1063-6927/02 $17.00 © 2002 **IEEE**

The remainder of the paper is organized as follows. In Section 2, we propose a new cost criterion for determining the optimal path and the changes that need to be made in the environment to incorporate these factors. In Section 3, the Reliable Multicast Algorithm (RMA) is presented. (A pseudo-code for the algorithm is given in the Appendix.) Section 4 presents a simulation study and a comparative performance analysis of the RMA and AODV [3] on RELSIM [1]. Finally, Section 5 concludes the paper with remarks on the future work.

2. A new cost criterion for determining the best path

Existing multicast algorithms assume that the best path between two nodes is the path with the minimum number of hops. This assumption may be true in wired networks where wires are fixed and also for infra-structured networks. But for a highly unstable network like MANET, the number of hops alone is not a good measure of the cost of wireless links connecting the hosts. The lifetime of links plays an important role in determining the cost associated with links. A link with a longer life and more hops is preferred over a link with shorter lifetime and fewer hops. This is because any change in the link topology results in the modification of the state of the whole system or at least a significant part of it, which results in the reconstruction of the links. Routing through such short-lived links result in frequent broadcast, which is costly in such a cost constrained network.

The lifetime of a link is thus a crucial factor in determining the best path in such networks. But a major challenge is how to determine the lifetime of a link. The MANET is highly unstable and prediction of the lifetime of a link in such a network is a challenging task. A simple prediction mechanism works as follows, each mobile host in the system maintains a list of all its neighbors, current and past, with their average time of stay. Whenever a new neighbor moves into the neighborhood or an old neighbor reenters the system or a neighbor moves out of the system, the neighborhood table is updated to take note of the changes. The Neighbor table has the following fields, for each neighbor (i) Neighbor ID, (ii) Average Time of Stay in the past, and (iii) Current Time of Stay

Whenever a new neighbor moves into the neighborhood, an entry is created for that neighbor with its time of stay set to zero, and a timer is started to measure the 'current time of stay'. When a neighbor moves out of the range of the host, the host modifies the status of the neighbor in the neighborhood table and updates its 'average time of stay' field and resets its 'current time of stay' field. When an old neighbor moves into the neighborhood again, its 'current time of stay' field is started.

Whenever there is a need to find the lifetime of a path between two neighbors, the neighborhood table is examined and the 'average time of stay' and 'current time of stay' fields are compared and based on that the lifetime is estimated. If the 'current time' is less than the 'average time of stay', then the difference is the estimated lifetime of the link. Otherwise, the lifetime of the link is estimated to have a minimum value, as we are not sure when the link is going to disappear.

The prediction mechanism can be improved by using neural networks to get more accurate predictions of the lifetime of links and also using better sample data space, like storing more than one past time of stay field.

3. The Reliable Multicast Algorithm (RMA)

The reliable multicast algorithm ensures a message delivery to set of destination nodes in a MANET provided the nodes are reachable. The underlying environment should be able to support multiple dynamic multicast sessions. Each mobile host can join or leave a multicast session in the system. Mobile hosts maintain connectivity with their neighbors by periodically broadcasting HELLO messages to their neighbors. These HELLO messages are used to indicate hosts of any changes in their neighborhood.

Each mobile host maintains a route table with the following fields: (i) Destination IP Address - Identifier of the destination, (ii) Next Hop IP Address - Next hop identifier, (iii) Bandwidth of the link, (iv) Lifetime of the link - Cost parameter of the link, and (v) Membership count of the link - It is defined to be the count of members and non-members of a particular multicast session in the link. This status field is used because in a cost constrained network like MANET, it is always preferable not to involve unnecessary nodes, which are not the intended recipients of a message. So a path mostly through the members of a multicast group will be preferred over a path through non-members, even though it may cost more.

Any mobile host, which is a member of a multicast session can put out messages to the multicast group at any time. For a particular multicast group, each host maintains a separate sequence number. This sequence number together with the source node's IP address and the multicast group number, uniquely identifies each message. A source that wants to transmit a multicast message gets the list of members of the multicast group. For each member of the multicast group, if a route is known to that member, the message is routed to the member along the known path. If a route to the member is not known, the message is broadcast. If two multicast destinations share the same next hop, both the control headers are packed together in only one data packet and that is sent forward. Similarly for broadcast messages, only one data packet is sent but coupled with all the control headers that need to be broadcast.

A message to a destination with a known route is posted through MKNOWN Message: MKNOWN < Source, Destination, Next hop, Lifetime of path, Bandwidth of path, Membership count of path, Data packet>.

- Lifetime of the path is the lifetime of the weakest link in the path.
- Bandwidth of the path is the bandwidth of the weakest link of the path.
- Membership count is a measure of the number of members and nonmembers in the path for a particular multicast session.

Message to a destination without a known route is posted through MUNKNOWN message: MUNKNOWN <Source, Destination, Lifetime of path, Bandwidth of path, Membership count of path, Status of message, Data packet>. Status of message is used to indicate whether the particular message is a retransmission or the original transmission.

Whenever a message is posted to a multicast group, a sequence table is created and is maintained with the sequence number of the message being posted. The source maintains a multicast table for each of the multicast sessions it is member of. It has the following fields: Session IP Address, Sequence Table, and Members of the Session. The sequence table has the following fields: Sequence Number, Message, and Members who have acknowledged.

The next hop node on the reception of a message checks whether it has a valid route to the destination. If so the packet is forwarded to the next hop, otherwise it is broadcast. Duplicate copies of the messages are discarded. For finding out duplicate copies, each mobile host maintains the key fields of the most recent messages it has received. The messages propagate and finally reach the destination.

A destination on the receipt of a message sends an acknowledge message back to the source through the path it received the message. The acknowledgement is posted through the MACK Message: MACK < Source, Destination, Next Hop, Bandwidth of path, Lifetime of path, Membership count of path>. The acknowledge message also maintains fields like bandwidth and lifetime so that reverse path is also updated in the route tables. The reception of every message triggers a route update in the route table. If the message had come through a better path from the same source, then the new route replaces the old route. Each intermediate node also performs a route update on the reception of a passing message and also updates the message with its information. The destination may receive multiple copies of the same message. If more than one copy of the same message is received by the destination, it checks up the new route through which the message has arrived with the old route through which acknowledge was sent. If this new route is better, an acknowledge message is sent on the new route.

The source on receipt of an acknowledge message, updates its sequence table with the information that the source of acknowledge has received the message. The source after it has sent the first list of messages, waits for a pre-specified period of time WAIT_TIME and then checks the sequence table. If all the members of the group have received the message, the sequence table entry is deleted. If not, then the message is posted again to the nodes with the special field set indicating it is a rebroadcast. All intermediate nodes receiving the message rebroadcast the message unless the destination is their next hop. The destination on receiving such a rebroadcast message, checks if it had already received this message. If so, it broadcasts an acknowledge back, which is rebroadcast until it reaches the source. The acknowledge is broadcast back using the BMACK Message: BMACK <Source, Destination, Bandwidth of path, Lifetime of path, Membership count of path>.

If the destination has never before received the message, it sends an acknowledge message back along the path on it which it received the message. This ensures that every message is delivered to the destination provided the network is connected. The procedure is repeated for a pre-specified number of times depending on whether all the nodes have been reached. If after the pre-specified number of tries, certain nodes are still unreachable, it implies that the nodes are unreachable and the process is aborted.

Any mobile host that wants to join a multicast session broadcasts a JOIN Message to the network, which is propagated across the network. All hosts, which receive the message, update their multicast membership with this new information and rebroadcast the message. The format of a JOIN message is JOIN < Node IP Address, Session IP Address>. Any mobile host that wants to leave a multicast session broadcasts a LEAVE to the whole network. Since there is no tree or mesh structure for maintaining the multicast session, a LEAVE message involves the removal of the host from the membership information and modification to the route table by changing the status of the routes that have this node as the Next hop, if possible. The format of a LEAVE message is LEAVE <Node IP Address, Session IP Address>.

No propagation of state information is carried out in the system. So each node maintains only the local information about its neighborhood. This does not create any inconsistencies in the system as every routing decision is taken at the local node. Whenever there is a change in the neighborhood of a node, i.e., whenever a neighbor moves out of range of the node, the node removes all routes with the neighbor as the next hop as they are now stale.

A pseudo-code for the algorithm is given in the Appendix.

Highlights of RMA

The RMA has the following interesting features:

- Reliable message delivery provided a node is reachable.
- New cost factors for determining the best path, which ensures useful utilization of the available limited set of resources.
- RMA doesn't depend on the existence of an underlying mechanism like Tree [3] or Mesh [4] to support the algorithm. These structures can be costly to construct, repair, and maintain. RMA dynamically constructs a graph, which can be a mesh or tree structure depending on the topology.
- Non-member nodes are rarely involved in transmission unless it is absolutely essential, preventing their unnecessary usage.
- RMA does not depend on the existence of a unicast routing protocol like LAM [2] or AMRoute [6] for its operation. RMA also performs unicast functionality. RMA can be made to work with any unicast algorithm with little modifications. Instead of RMA explicitly constructing and maintaining route tables, it can utilize the tables constructed by the underlying unicast protocol.

4. A Performance Study

A simulation of the RMA algorithm was implemented on RELSIM [1], a scalable distributed mobile ad hoc network simulator, developed to test the reliability characteristics of routing algorithms on mobile ad hoc networks.

CORBA provides the basic architecture for the simulator. The simulator, developed in java, uses the multithreading features of the java to simulate actions of individual mobile hosts in the environment.

RMA and Multicast AODV [3] multicast routing algorithms were implemented on the RELSIM and their performances over a variety of situations was studied and tabulated.

The simulation environment consisted of 50 nodes in an area of 1000x1000 meters. All nodes in the system have uniform power capabilities and they are distributed uniformly across the area initially. The power range of each node is approximately 200 m. All nodes follow the random way point model of motion, i.e., each node moves in a random direction with a fixed speed for a random period of time and then rests for a uniform time period and then restarts again. The propagation function used is the free space propagation function.

Table 1 Simulated parameter values used in the simulation of MAODV

Parameter name	Meaning	Value
allowed hello_loss	Number of allowed hello losses	2
Group_hello_interval	Frequency of group hello messages	5s
Hello_interval	Frequency of hello messages	1s
Max_retrans	maximum number of retransmissions	10
mtree_build	Time to wait to receive a MACT	2s
retransmit_time	Time to wait for data packet transmissions	1s
Reverse_route_life	Time to keep reverse route entries	3s
route_expiration	Lifetime of a route entry	3s
Req_retries	Max number of rreq retransmissions	2
rte_discovery_timeout	Max time to wait for a RREP	1s

Table 2 Simulated parameter values used in the simulation of RMA

Parameter name	Meaning	Value
hello_interval	Frequency of hello messages	1s
max_retrans	maximum number of retransmissions of data	2
b_retrans	number of broadcast retransmissions	1
retransmit_time	Time to wait for data acks before initiating retransmissions	10s
hello_wait_period	time to wait for hello before deciding that the neighbor has moved out	2s

The following three standard metrics were used for comparing the performance of the two multicast algorithms, MAODV and RMA: **PACKET DELIVERY RATIO:** The packet delivery ratio is the ratio of the number of data packets delivered to the number of packets originally sent. It is a measure of the algorithm's reliability characteristics. **DATA OVERHEAD:** Data overhead is the ratio of the total number of data packets in the system to the number of data packets delivered. In a comparison perspective, data overhead helps to distinguish the algorithm that injects fewer data packets in the system, normalized by the number of data packets delivered. **CONTROL OVERHEAD:** Control overhead is the ratio of the number of control bytes to the number of data packets delivered. Each of these metrics captures different characteristics of the algorithms, but they present a common means to draw comparisons between the algorithms.

4.1 Results for Packet Delivery Ratio

We analyze the behavior of the algorithms under the two routing metrics (lifetime and hop count) in terms of packet delivery ratio, i.e., the reliability of the algorithms.

Comparison Using Lefitime Metric: Figures 1 and 2 present a comparison between the algorithms under fixed rest times of 10s and 50s, respectively.

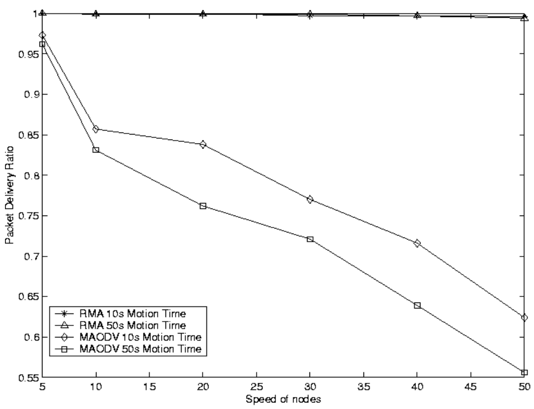

Figure 1. Packet delivery ratio vs. Speed at rest time 10s using lifetime metric.

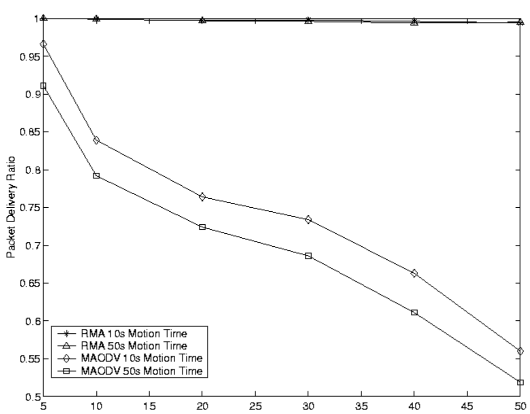

Figure 2. Packet delivery ration vs. Speed at rest time 50s using lifetime metric.

Under both the rest times, RMA has a packet delivery ratio of close to 1, which confirms the reliability of the algorithm. In the case of MAODV, at low speeds, it achieves a packet delivery ratio close to 95%, but as the speed increases, the delivery ratio drops significantly, going down to about 50%. Increase in motion time, under fixed rest times, does not have any impact on the delivery ratio of the RMA algorithm, but has an effect on the delivery ratio of MAODV. At low speeds, the impact of increase in motion time is not appreciable, but at high speeds, the difference in delivery ratio for MAODV gets as high as 8%.

Comparison Using Hopcount Metric: Figures 3 and 4 present a comparison between the algorithms under fixed rest times, using the hopcount metric. Use of hopcount as a means of determining optimal routes does not make any difference in the case of the RMA as it has a packet delivery ratio of close to 1. Variation in motion times does not make any difference in the behavior of RMA with respect to packet delivery. MAODV has a packet delivery ratio close to 90% at low speeds. But as the speed increases, the delivery ratio drops significantly down to about 50% at the higher speeds. An increase in motion time further brings down the delivery ratio, noticeably at higher speeds.

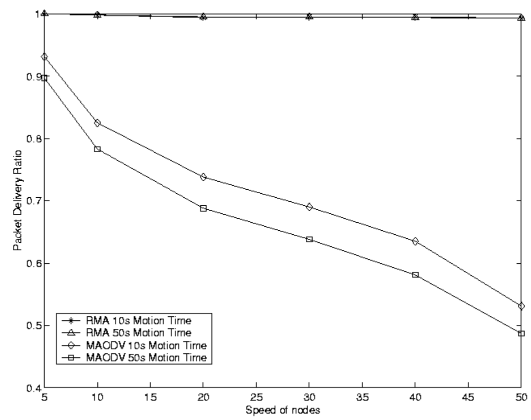

Figure 3. Packet delivery ratio vs. Speed at rest time 10s using hopcount metric

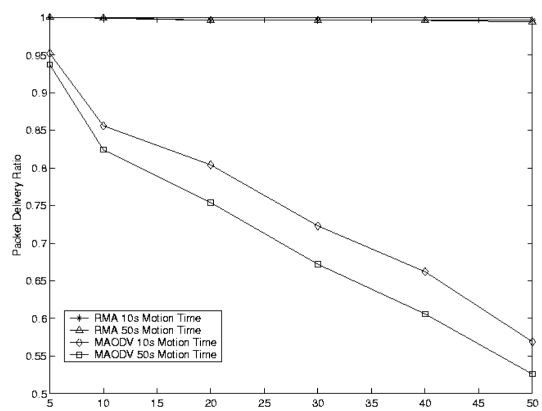

Figure 4. Packet delivery ration vs. Speed at rest time 50s using hopcount metric

A comparison with respect to packet delivery ratio clearly shows that RMA is a reliable multicast algorithm. MAODV has good delivery ratios only at low speeds and fares poorly at high speeds. The delivery ratio of MAODV is sensitive to the variations in the environment like change in rest time and/or change in motion time. But RMA is insensitive to the environmental characteristics in ensuring 100% delivery.

4.2 Results for Data Overhead

In this section, we analyze the behavior of the algorithms with respect to the metric of data overhead, which measures the packets injected into the system by the algorithm.

Comparison Using Lefitime Metric: Figures 5 and 6 present the behavior of MAODV and RMA under fixed rest times of 10s and 50s, respectively, using the lifetime as the routing metric. At low speeds of about 5m/s, MAODV has a lower data overhead than RMA, at times having a difference of about 0.8 units. As the speed increases, RMA shows a smooth and measured increase in the data overhead. But MAODV has an abrupt increase in the data overhead.

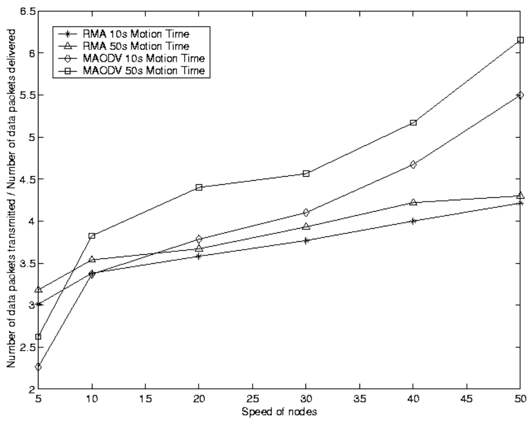

Figure 5. Data Overhead vs. Speed at rest time 10s using lifetime metric.

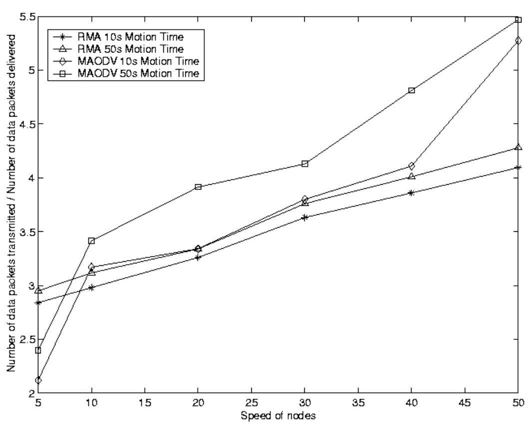

Figure 6. Data Overhead vs. Speed at rest time 50s using lifetime metric.

At higher speeds, RMA has a much lower overhead than MAODV. At speeds of about 50m/s, the difference is as high as 2.0 units, which is very significant. Under stable conditions, MAODV tends to get close to RMA with almost the same overhead. For example, when the rest time is 50s and motion time is 10s, until a speed of 40m/s, the difference in overhead between RMA and MAODV is quite small, being of the order of 0.1 units. RMA using the lifetime metric shows only small variations in overhead when the environmental variables like motion time change. But an increase in motion time has a very significant effect on the overhead of MAODV.

Comparison Using Hopcount Metric: Figures 7 and 8 present the comparison between MAODV and RMA using the hopcount metric under fixed rest times.

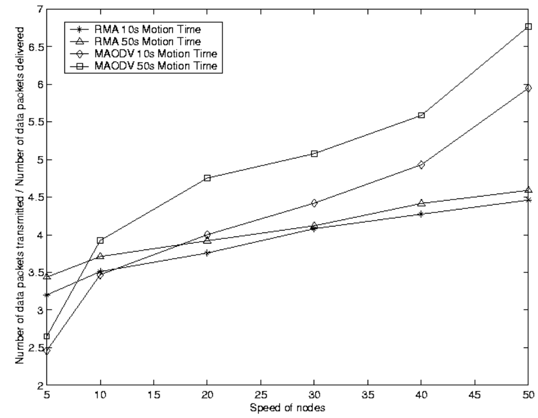

Figure 7. Data Overhead vs. Speed at rest time 10s using hopcount metric.

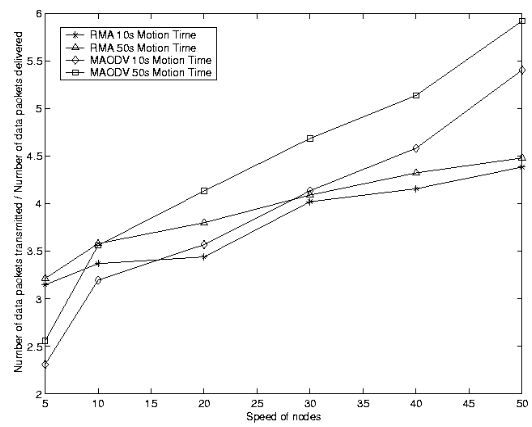

Figure 8. Data Overhead vs. Speed at rest time 50s using hopcount metric.

When hopcount is used as the routing metric, MAODV has a lower overhead than RMA at low speeds, but has a much higher overhead at higher speeds. Under stable conditions, MAODV has an overhead that is quite comparable to that of RMA till 40m/s speed. RMA has an almost flat curve with an increase in overhead of only about one unit when the speed increases from 5m/s to 50m/s. In contrast, MAODV has a very large change of almost 4 units when speed increases from 5m/s to 50m/s.

Analysis with respect to the data overhead metric indicates that MAODV performs well at lower speeds, In the case of RMA there is a slightly higher overhead. But at higher speeds, RMA achieves a lower overhead. MAODV on the other hand, fares poorly when the system is unstable.

4.3. Results for Control Overhead

In this section, we analyze the behavior of the multicast algorithms with respect to the control overhead metric.

Comparison Using Lifetime Metric: Figures 9 and 10 present a comparison for rest times of 10s and 50s, respectively.

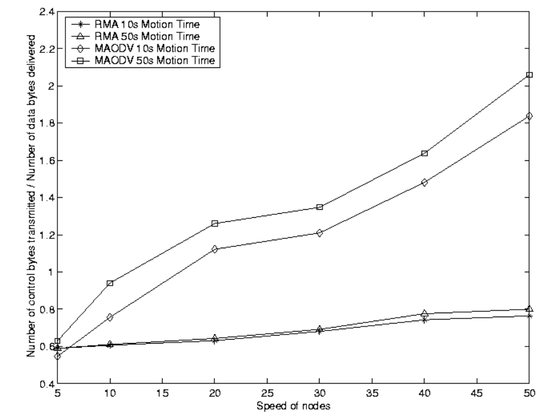

Figure 9. Control Overhead vs. Speed at rest time 10s using lifetime metric

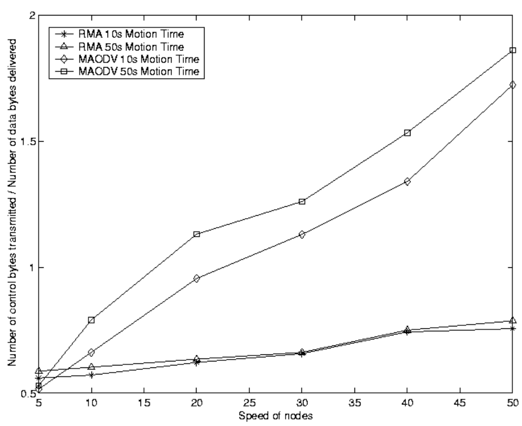

Figure 10. Control Overhead vs. Speed at rest time 50s using lifetime metric

RMA has an almost uniform control overhead at all speeds, varying by at the most 0.2 units. MAODV has a varying control overhead, which varies by as much as 1.5 units when speed goes up from 5m/s to 50m/s. At low speeds of 5m/s, both MAODV and RMA have almost the same overhead. But as speed increases the difference grows and becomes as high as 1.2 units at a speed of 50m/s. The variation in motion time has almost no effect on the control overhead for RMA. But increase in motion time has the effect of increasing the control overhead for MAODV.

Comparison Using Hopcount Metric: Figures 11 and 12 compare the behavior of the multicast algorithms using the routing metric of hopcount under fixed rest times.

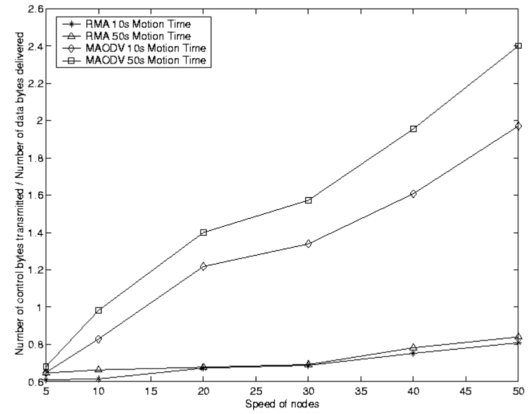

Figure 11. Control Overhead vs. Speed at rest time 10s using hopcount metric.

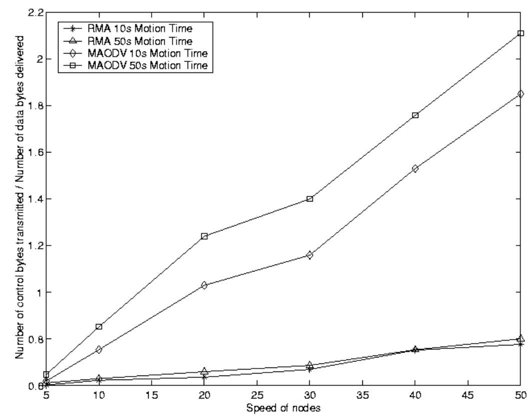

Figure 12. Control Overhead vs. Speed at rest time 50s using hopcount metric.

RMA has the same uniform flat curve and MAODV gets close to RMA only at lower speeds. Unlike the case of the lifetime metric, where at a speed of 5m/s RMA had somewhat poorer performance than MAODV, RMA does better than MAODV at all speeds. The difference between RMA and MAODV at a speed of 50m/s becomes as high as 1.4 units.

Comparison with respect to control overhead reveals that except for a very few cases, RMA clearly outperforms MAODV. At high speeds, the difference between the two algorithms is very significant. RMA achieves 100% delivery rate with little control overhead due to the effective packaging of data and control packets together. The standalone control messages, namely, MACK and BMACK are very small in the case of RMA unlike that of MAODV, namely, RREQ, RREP, and MACT.

5. Conclusion and Future Work

We presented a reliable multicast algorithm for MANET, which insures at least one message delivery at low cost. The algorithm is based on a new cost criterion, called link lifetime, for determining the optimal path between a pair of nodes. A comparison of the RMA with an existing algorithm, MAODV, with respect to the three performance metrics, packet delivery ratio, data overhead, and control overhead, suggests that RMA has a better performance than MAODV. At a low speed of 5m/s, MAODV does well with a good packet delivery ratio and low control and low data overheads. But as speed increases, the performance deteriorates significantly. RMA has almost unity delivery ratio, and low control and data overheads at all speeds. Thus RMA is a reliable multicast algorithm with a stable behavior. If the system under consideration is very stable and has low speeds of the order of 5m/s, then MAODV might be a better choice than RMA. But for mobile ad hoc network environments consisting of automated vehicles with irregular pattern of behavior, RMA would be the preferred choice.

We plan to enhance the RMA algorithm such that it works fine even when certain nodes are unreachable during initial stages of transmission but later become reachable, and delivers them all the undelivered messages. We plan to do a rigorous testing of the RMA on the RELSIM and come up with a comparison of the reliability of the RMA and other representative multicast protocols like ODMRP [10] and MCEDAR [4] and compare it with other recently proposed reliable multicast protocols like the adaptive protocol in [9].

References

[1] Thiagaraja B Gopalsamy, Multicasting in Mobile Ad Hoc Networks, Technical Report, 2001.

[2] L.Ji, M.S.Corson, *A Lightweight Adaptive Mutlicast Algorithm*, Pages 1036-1042, IEEE transactions in Mobile Computing 1998.

[3] Elizabeth M.Royer, Charles E. Perkins, *Multicast operation of the Ad-hoc On-Demand Distance Vector Routing Protocol*, In the proceedings of Mobicom'99, pages 207-218, Seattle, Washington.

[4] P.Sinha, R.Sivakumar, V.Bharghavan, *MCEDAR: Multicast Core Extraction Distributed Ad hoc Routing*. In the proceedings of Wireless Communications and Networking Conference, 1999.

[5] C. Ho, K. Obraczka, G. Tsudik, K. Visvanath, *Flooding for reliable multicast in multi hop ad hoc networks*, In the proceedings of DIAL M99, Seattle, Washington.

[6] E.Bommiah, M.Liu, A.McAuley and R.Talpade*, AMRoute: Adhoc Multicast Routing Protocol,* IETF MANET draft 1998.

[7] C.E. Perkins, E.M. Royer, *Ad hoc On Demand distance Vector routing,* Proceedings of 2nd IEEE workshop on mobile computing systems and applications, Feb 1999, pp 90-100.

[8] K.Obraczka, G. Tsudik, *Multicast Routing Issues in Ad Hoc Networks*, Proceedings of IEEE ICUPC '98 Oct. 1998.

[9] S.K.S. Gupta, P.K. Srimani, *An Adaptive protocol for reliable multicast in Multi-Hop radio networks*, 2nd IEEE workshop on mobile computing systems and applications 1999.

[10] S.J.Lee, W.Su, M.Gerla, *On-Demand Multicast Routing Protocol (ODMRP) for Ad Hoc Networks*, IETF MANET draft June 1999.

[11] K. Obraczka, G. Tsudik, and K. Viswanath, *Towards Reliable Multicast in Multihop Ad Hoc Networks, ACM/Baltzer Wireless Networks*, Vol 7, No 6, 2001.

[12] G. Anastasi, A. Bartoli, and F. Spadoni, A Reliable Multicast Protocol for Distributed Mobile Systems: Design and Evaluation, IEEE Trans. On Parallel and Distributed Systems, Oct. 2001, pp. 1009-1022.

APPENDIX: Pseudo-code for RAM

A Source Transmitting a Message:

- For each message assign a sequence number. For each destination, check the route table.
- If route is known to the destination, combine messages that have the same next hop. Post an MKNOWN message to the destination through the next hop.
- Else, combine all the headers of unknown destinations. Broadcast an MUNKNOWN message.
- Wait for an acknowledge message from all the destinations.
- If acknowledge is not received within pre specified wait time, for each destination from which acknowledge has not been received, put an MUNKNOWN message with RETRANSMIT field set indicating retransmission of message.
- Repeat previous step a pre-specified number of times or until all the nodes have acknowledged.
- If multiple copies of MACK are received from a destination, choose the path with the best cost factor as the path to the destination.

A Destination Handling a Message:

- On receiving a message, send the MACK along the path on which the message came.
- If multiple copies of the same message are received, then post a MACK along the new path if the new path is better than the already acknowledged path. Suitably update routing table.
- When a RETRANSMIT message is received, If message has not been received ever before, then post an acknowledge message along the best path as before, else, post a BMACK message.

An Intermediate Node Handling a Message:

- On receiving a message, check if it is a duplicate message.
- If not a duplicate message, update the route table with this new message information. Check the route table if there exists a path to the intended destination. Update the lifetime of the path.
- If a route exists, post a message along the route.
- If no route exists, broadcast the message.

Maintenance of the Environment

- Broadcast a Hello message to the neighbors. Wait for a pre-specified time and then repeat Hello transmissions.
- Check the route table and remove stale routes.
- On receipt of a Hello message from a neighbor, Update the neighbor table.
  - If the neighbor is new, add it to the list.
  - If the neighbor is old but has been out of contact, update its lifetime field.
- If a neighbor has failed to transmit a Hello message within a specified time, update its lifetime by incorporating the current lifetime accumulated so far.
- If neighbor has retransmitted a Hello message within the pre specified gap, reset the wait timer.

1063-6927/02 $17.00 © 2002 **IEEE**