Journal of Computer Science 1 (3): 424-428, 2005
ISSN 1549-3636

Adaptive Reliable Efficient Multicasting Protocol for Mobile Ad Hoc Networks

K. Baskaran, A. Ebenezer Jeyakumar, V. Sumathy and J. Lakshmana Perumal
Government College of Technology, Coimbatore-641013, India

Abstract: The Adaptive Reliable Multicast Protocol (ARM) with intermediate node support is introduced to guarantee message delivery to all multicast receivers in mobile ad hoc environment. ARM distributes multicast message cache and retransmission tasks among intermediate nodes to offer a scalable reliable multicasting. Due to limited memory and node's mobility, ARM defines intermediate node as all nodes which overhear multicast messages. Thus it includes not only multicast traffic conveyors but also their neighbors and group members. Simulation results show that ARM has a packet delivery close to 100% and maintains a low bandwidth consumption facing to frequent topology change. This protocol is also stable as traffic load increasing.

Keywords: Mobile Ad Hoc Networks, Adaptive Reliable Multicast

INTRODUCTION

A Mobile Ad-Hoc Network (MANET) is a multi-hop network formed by a collection of mobile nodes without using fixed infrastructure. Their portability and fluidity make them to be ideal choices for applications such as emergency rescue operations, group travel, instant Internet game or file distribution during a conference. Many of these applications require error-free one-to-many or many-to-many data delivery. The properties of MANETs, such as limited bandwidth, low memory capacity and high degree of mobility, make the design of a reliable multicast protocol for MANET a challenging task.

For this aim, we design a reliable multicasting protocol, called Adaptive Reliable Multicast protocol (ARM) with intermediate node support, which extends receiver-initiated Automatic Repeat Request (ARQ) mechanism [1, 2] to MANET. Our main contribution is that ARM distributes packet storage and retransmission responsibility to all nodes which overhear multicast packets. These nodes are called intermediate nodes. According to this definition intermediate nodes include not only group members and nodes which forward multicast packets but also the neighbors of multicast packet forwarders. In reliable multicast, retransmission load of source is a function of link loss rate, size of network and group. In MANET, link loss rate is relatively high due to wireless interface and node's mobility. Thus, we think it is necessary to make intermediate nodes share retransmission tasks. Retransmission made by intermediate nodes lead to recovery packets travel a shorter route than original ones traveled and consequently achieves a higher recovery success and lower bandwidth consumption. Intermediate nodes need to store multicast packets for retransmission while limited memory prevents them to store all packets.

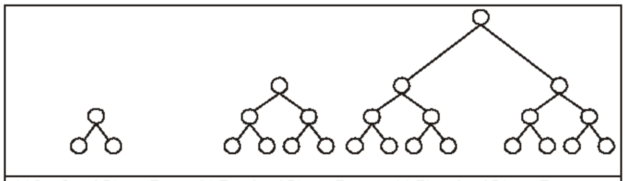
Our strategy is that in ARM, intermediate nodes randomly store overheard multicast packets to reduce duplicated cache among neighbors and a node queries its neighbors about the request packets before forwarding a retransmission request. Furthermore, this protocol needs no other control packet than negative acknowledge message (NACK) and independent of unicast routing protocols. The route to source is established by on-going traffic and retransmission paths are established during NACK forwarding. ARM can cooperate with either tree-based or mesh-based multicast routing protocols, such as [3-6]. We use multicast delivery structure to indicate multicast tree and mesh in the rest study.

The rest of this study is organized as follows. Section II analyzes the necessity of distributing retransmission task to some other nodes than source in MANET. Section III describes ARM in detail. Section IV presents performance analysis. And concluding remarks are made in Section V.

MATHEMATICAL ANALYSIS

In this section, we analyze the source retransmission load on binary trees to approve the necessity of distribution retransmission responsibility. We suppose that the tree is rooted at source and all leaves are receivers. The source sends multicast packets to these receivers. The packet loss rate on each link is and it is the source that takes the responsibility of retransmission.

First, we calculate the retransmission load of source (root) to guarantee the delivery to two receivers (leaves) in Fig. 1 (a). In this case, the retransmission load is



(a)One Level Binary Tree (b) Two level Binary Tree (c) Three level Binary Tree

Fig. 1: Binary Trees

$_{r\,=}[\ \ (2\ +\ {}^{2})\ /\ (1-\ \ {}^{2})]$ for level one. While, a two level binary tree (Fig .1 (b)) can be seen as a one level tree in which two leaves are sub one level tree. Thus, the retransmission load in two level tree is $_{r\,=\,3}\ [(2\ +\ {}^{2})\ /\ (1-\ \ {}^{2})]$. Consequently, as network and group size increase, the retransmission load of root in a three level tree (Fig. 1(c)) is $_{r=7}\ [(2\ +\ {}^{2})/(1-\ \ {}^{2})]$. These formulae demonstrate that retransmission load is a function of group and network size but also packet loss rate on links. In wired network, packet loss rate is relatively small and it is network size that plays a key role in retransmission overhead. Local recover/retransmission strategies are applied only to large scale networks to maintain the scalability of reliable multicasting. However it is not the case in MANET, where loss rate is relatively high due to wireless interface and node's mobility. Even in a small network, retransmission load might be important. For example when p equals to 0.034, the retransmission rate is 0.5. On the contrary, if every node in the tree takes the retransmission responsibility to its direct down-stream nodes, the retransmission load of source is always that in one level tree whatever the tree size is. Therefore, it is necessary to distribute retransmission tasks among source and some other nodes to reduce this overhead in MANET.

ADAPTIVE RELIABLE MULTICAST PROTOCOL WITH INTERMEDIATE NODE SUPPORT

System Model: In this study, we make the following assumption. Links between nodes are symmetric. Before sending data packet to group, source assigns a consecutive sequence number into packet. Then a multicast routing protocol delivers packet to the receivers. Receivers detect losses primarily by sequence gap in the data packets. During a multicast session, senders have all packets they sent and receivers have all packets they received. We consider a scenario where there are n sources and m receivers in the multicast group sharing the same multicast delivery structure.

ARM Protocol Design Principle: In ARM, intermediate nodes are group members, nodes which convey multicast traffic and the neighbors of these conveyors, in brief, all nodes that overhear multicast traffic. These nodes are active in the sense that they cache multicast packets and perform retransmission. When a multicast traffic conveyor forwards packets, the broadcast nature of the air interface permits its neighbors to overhear the packets. Thus, these neighbor nodes can help to cache data packets for future retransmission. For example, Fig. 2 illustrates a simple MANET where source sends packets to three receivers R_1, R_2, R_3. When node A forwards multicast packets, its neighbor node K can_ receive those packets at same time. Then node Y can store and participate in the retransmission if there is delivery failure to R_2 and R_3.

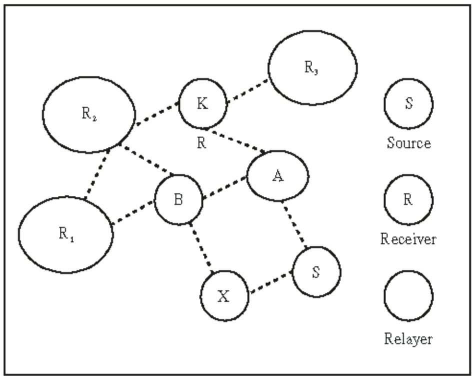


Fig. 2: Multicast Packet Delivery

Intermediate nodes store packets with a certain probability (denoted by p) to realize distributed multicast data cache. There are some further reasons why we use such a probability. (i) The memory capacity of mobile nodes is limited. If nodes store every data packet they receive, they can only keep the newest packets. (ii) It is unnecessary to store all packets. Simulation results [3, 4]) show that the multicast routing protocol can deliver safely most of the traffic. Storing successfully delivered packet wastes memory capacity. (iii) Nodes mobility causes frequent changes in their roles. A node can be multicast traffic conveyor at a given moment and becomes a neighbor at the next moment, or is far away from the structure.

ARM Protocol Description: Each node in ARM reserves a memory space as multicast packet caching buffer, which behaves in a FIFO fashion. Nodes dispose a table called relayed packet list for duplication detection. This table contains three fields: group identification, source identification and sequence number. These information identify a multicast packet in the network. Before forwarding a multicast packet, a node stores the relevant information in the relayed packet list. Packets listed in the table will not be processed second time. ARM defines two kinds of NACKs: local broadcast NACK which are sent to neighbors for local inquiry and unicast NACK which are addressed to the request packet's source. A NACK message contains group identification, source identification and a reference list. During data forwarding, a header is added in traffic packets in which there is a field to indicate that the packet is original one or retransmitted. ARM is a receiver-initiated, NACK-based scheme in which receivers are responsible for detecting and requesting lost packets. This protocol contains two phases: data delivery phase and data repair phase. In the data delivery phase, the source assigns consecutive sequence numbers into data packets before sending them. At the same time when a multicast routing protocol delivers this packet to group receivers, ARM caches packets and generates reverse path to the source. When intermediate nodes receive a non-duplicate data packet, they fill their relayed packet list to avoid processing the same packet next time. Routing protocol also uses relayed packet list during delivery for the same goal. Node uses the following way to achieve cache received packets with probability p. Node asks a uniform distribution random value generator to generate a random number between 0 and 1. Node stores this packet if the random number is smaller than p. Registering the node from which the original packet comes; nodes update the reverse path to the source. In the data repair phase, receivers detect losses primarily by sequence gap in data packets and initiate a negative acknowledge to request retransmit the lost packets. When receiving a NACK, nodes aggregate NACK with processed NACKs and delete duplicate requests. Before forwarding NACK to the source, nodes do local recovery beginning with looking for the requested packet in their cache. If the packet is not found, they delete the relevant packet information from relayed packet list to permit that multicast routing protocol forwards the packet a second time and then send a broadcast NACK to check the caches of their neighbors. In case of local repair failure, a unicast NACK is forwarded to the next hop on the reverse path toward the source. These steps repeat until the requested packet is found. Before transmission node marks in the packet header that this packet is a retransmitted one. Then, the requested packet is sent by multicast routing protocol as a normal packet and forwarded only by the conveyors which do not have the relevant packet information in their relayed packet list. In case of retransmission failure, data repair phase is re-executed.

PERFORMANCE ANALYSIS

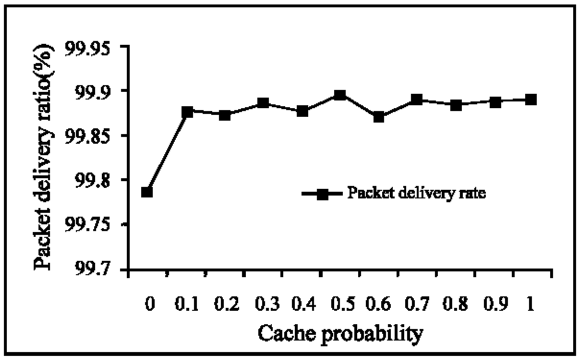
In the simulations, ARM was integrated into Multicast Routing protocol with Dynamic Core [4].

Our simulation modeled a network of 50 mobile nodes placed randomly within a 1100m * 1100m area. Radio propagation range for each node was 250 meters and channel capacity was 2 Mbits sec ^{1}. Each simulation executed for 900 seconds of simulation time. Collected data was averaged over multiple runs with different movement scenarios. For each multicast group, 10 nodes were randomly chosen as multicast member. These members join the multicast session at the beginning of the simulation and remain as members throughout the simulation. Multicast traffic was generated by constant bit rate sources. Each source transmitted 3200 packets during a simulation with a speed of 4 pkt sec ^{1}. The size of data payload was 512 bytes. These sources were attached to nodes which were arbitrarily chosen among multicast members. We studied the performance by varying three parameters: the probability p, the maximum movement speed and the number of sources. The number of groups was the mode 2 of the number of sources. Two metrics were used for performance analysis: Packet delivery ratio, which is the percentage of data packets correctly delivered to receivers over the number of data packets that should have been received and Source retransmission load, which is the number of data packets retransmitted by sources. The performance analysis contains three aspects, the impact of cache probability, node's mobility and traffic load. The rest of this section presents them in detail.

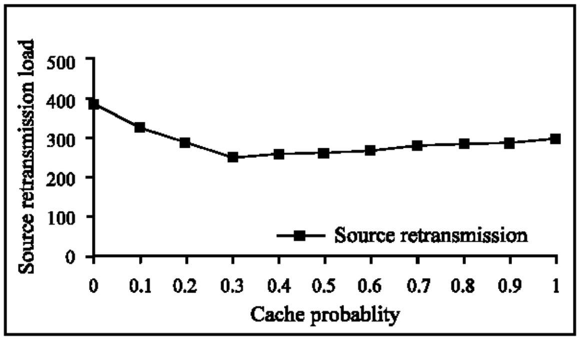
The Impact of Cache Probability p: First, we set the number of source to 6 (three groups and two sources per group) and maximum movement speed to 5 m s ^{1} while vary the cache probability from 0 to 1 to see the behaviors of ARM. When p equals to 1, nodes store all packets they overhear. This results to only the newest packets being stored in cache. On the contrary, when p is set to zero, nodes do not cache any packet.

Figure 3(a) shows that packet delivery ratio is improved when cache probability passes from 0 to 0.1 then remains stable. Thus, increase cache probability cannot enhance packet delivery ratio. On one hand, when cache probability increases, the duplicate storage among neighbor's increases while the number of total different packet in the cache of intermediate nodes decreases and NACK should go further to find the request packet(s). On the other hand, source re-transmission load (illustrated in Fig. 3 (b)) rises along with the increase of cache probability after reaches the minimum value at 0.3 cache probability. To achieve low source retransmission load and high network throughput, cache probability should keep small. ARM gives the best compromise between packet delivery ratio and bandwidth consumption when cache probability equals to 0.3. In the following simulations, we choose this value as cache probability.

The Impact of Node Mobility: In this aspect, the maximum movement speed of nodes range in the set { 0, 1, 5, 10,15,20} m s ^{1} and the number of sources is fixed to 4. Fig. 4(a) shows the packet delivery ratio with different maximum speed of these three protocols. The results show that ARM is reliable against frequent topology changes: mobility has nearly no impact on the performance of ARM while frequent topology changes

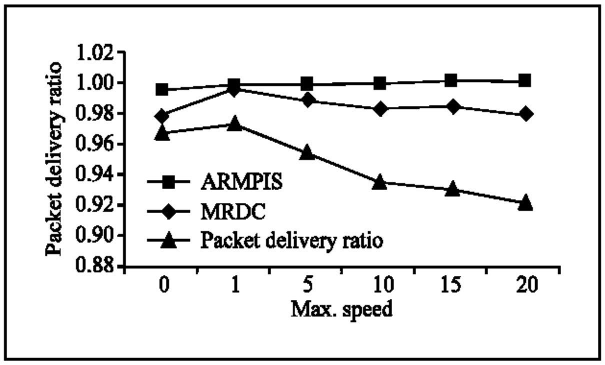


(a) Packet Delivery Ratio vs. Cache Probability

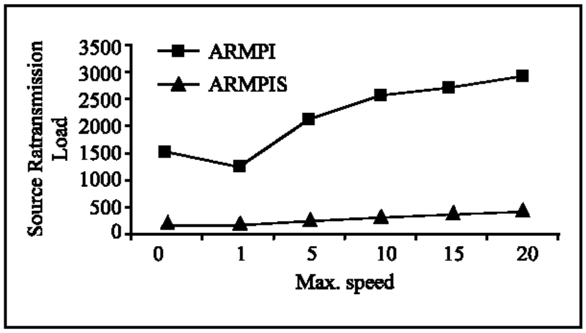


(b) Source Retransmission Load vs. Cache Probability

Fig. 3: The Impact of Cache Probability p



(a) Packet Delivery Ratio vs. Maximum Speed



(b) Source Retransmission Load vs. Maximum Speed

Fig. 4: The Impact of Node's Mobility

degrade the performance of underlying multicast routing protocol. ARM1 gives a worse performance than ARM. In ARM1, only source can resend the lost packets, thereby the recovery packets have the same loss probability as the primary ones. But local recovery mechanism can decrease this risk by proposing a shorter path for retransmission. Figure 4(b) illustrates the number of packets retransmitted by source. ARM makes source retransmit five times less packets than ARM1 does. ARM1 should retransmit more packets as node's mobility increase since MRDC delivers less packets. Compared with ARM1, ARM distributes retransmission works and have less retransmission failures. ARM is reliable facing to topology changes and can deliver nearly 100% data packets in all mobility cases. This protocol is also scalable in the sense that it does not generate significant retransmission load as node's movement speed increases.

The Impact of Traffic Load: In traffic load experiment, node mobility speed is moderate with maximum speed at 5 m s ^{1}. The number of multicast sources increased from 2 to 8. The number of groups was consequently increased from 1 to 4.
The packet delivery ratio as a function of the number of sources is presented. ARM maintains nearly 100% packet delivery ratio till seven sources and then appears a little degenerative. However, it can transfer more than 99% data packets to all receivers. This shows that this protocol reliable when traffic augments.

CONCLUSIONS

In this study, we introduced our adaptive reliable multicast routing protocol with intermediate node support, ARM, to support reliable multicast in mobile ad hoc network. Intermediate nodes in ARM are nodes that overhear multicast messages. These nodes store multicast messages in their buffers for future retransmission to enhance the performance of reliable multicast and reduce bandwidth utilization caused by retransmission. Instead of caching every multicast packet, nodes save them with a probability, called cache probability, to reduce the probability of cache same message among neighbors. The performance evaluations suggest a small cache probability since a high cache probability degrades message cache distribution among neighbors. The simulation results show that ARM is reliable in both low and high mobility cases when network load is moderate. The source's retransmission load is greatly reduced.

REFERENCES

1. Li-Wei, H.L., S.J. Garland and D.L. Tennhouse, 1998. Adaptive reliable multicast. In INFOCOM '98, San Francisco, CA, pp: 581-589.
2. Speakman, T., D. Farinacci, S. Lin and A. Tweedly. Pretty good mul-ticast. Internet-Draft.
3. Royer, E.M. and C.E. Perkin, 2000. Multicast ad hoc on-demand distance vector (maodv) routing. Internet-Draft.
4. Shiyi Wu and Christian Bonnet, 2001. Multicast routing protocol with dy-namic core. In IST 2001, Tehran, Iran, pp: 274-280.
5. Garcia-Luna-Aceves, J.J. and E.L. Madruga, 1999. A multicast routing protocol for ad-hoc networks. In IEEE INFOCOM'99, New York, NY, pp: 784-792.
6. Lee, S.J., M. Gerla and C.C. Chiang, 1999. On-demand multicast routingprotocol. In IEEE WCNC'99, New Orleans, LA, pp: 1298-1304.
7. Kevin Fall and Kannan Varadhan, Eds. ns notes and doc-umentation, 1997. The VINT project, UC Berkeley, LBL, USC/ISI and Xerox PARC, Avaible from http://www-mash. cs.berkely.edu/ns.
8. http://www.eurecom.fr/ ˜ wus.
9. Jorjeta, G.J. and D.B. Johnson, 2001. Adaptive demand-driven multicast routing in multi-hop mobile ad hoc networks. In the ACM Symposium on Mobile Ad Hoc Networking and Computing, Long Beach, CA, USA, pp: 33-44.
10. http://www.monarch.cs.rice.edu/multicast extensions.html.
11. Sandeep K.S. Gupta and Pradip K. Srimani, 1999. An adaptive protocolfor reliable multicast in mobile multi-hop radio networks. Technicalreport, the 2nd IEEE Workshop on Mobile Computing Systems and Applications.