ARMPIS: AN ACTIVE RELIABLE MULTICASTING PROTOCOL FOR AD HOC NETWORKS

Shiyi Wu and Christian Bonnet
Institut Eurécom *
2229, Route des Crêtes
06904 Sophia Antipolis, France
{Firstname.Name}@eurecom.fr

Abstract

The active reliable multicast protocol with intermediate node support (ARMPIS) is introduced to guarantee message delivery to all multicast receivers in ad hoc environment. ARMPIS distributes multicast message cache and retransmission tasks among intermediate nodes to offer a scalable reliable multicasting. Due to limited memory and node's mobility, ARMPIS 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 ARMPIS 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: Ad hoc networks, active reliable multicast

I. 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.

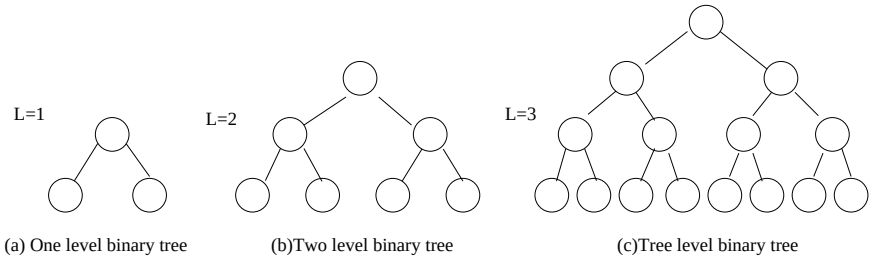
For this aim, we design a reliable multicasting protocol, called active reliable multicast protocol with intermediate node support (ARMPIS), which extends receiver-initiated automatic repeat request (ARQ) mechanism [1, 2] to MANET. Our main contribution is that ARMPIS distributes packet storage and retransmission 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 ARMPIS, 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. ARMPIS 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 paper.

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

II. MATHEMATIC 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 λ multicast packets to these receivers. The packet loss rate on each link is γ and it is the source that takes the responsibility of retransmission.

* EURECOM research is partially supported by its industrial partners: ASCOM, Swisscom, FranceTelecom, La Fondation Cegetel, BouyguesTelecom, Thales, STMicroeletronics, Hitachi Europe and Texas Instruments



(a) One level binary tree (b)Two level binary tree (c)Tree level binary tree

Fig. 1. Binary trees

First, we calculate the retransmission load of source (root) to guarantee the delivery to two receivers (leaves) in Figure1(a). In this case, the retransmission load is $\lambda_r(L=1) = \lambda\frac{2\gamma+\gamma^2}{1-\gamma^2}$. While, a two level binary tree (Figure1(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 $\lambda_r(L=2) = 3\lambda\frac{2\gamma+\gamma^2}{1-\gamma^2}$. Consequently, as network and group size increase, the retransmission load of root in a three level tree (Figure1(c)) is $\lambda_r(L=3) = 7\lambda\frac{2\gamma+\gamma^2}{1-\gamma^2}$.

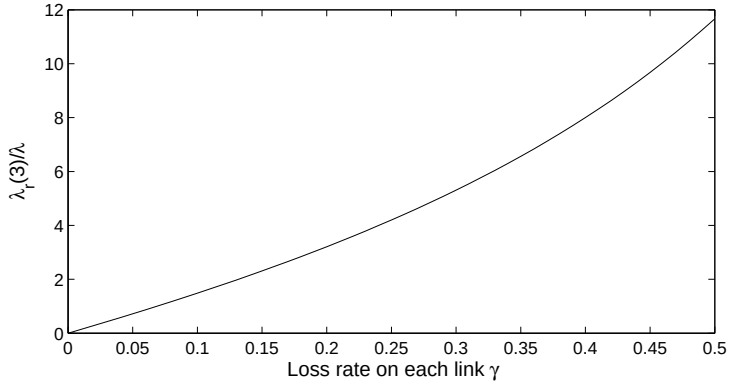


Fig. 2. Retransmission rate as a function of loss rate on each link p

The above formulas 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. Figure2 illustrates how source retransmission rate (retransmission load λ_r over original load λ) varies as loss rate changes in a three levels binary tree. 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 downstream 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 .

III. Active Reliable Multicast Protocol with Intermediate node Support

A. System model

In this paper, 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.

B. ARMPIS Protocol Design Principle

In ARMPIS, 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, Figure 3 illustrates a simple MANET where source S sends packets to three receivers R_1, R_2, R_3. When $nodeA$ forwards multicast packets, its neighbor $nodeY$ can receive those packets at same time. Then $nodeY$ can store and participate retransmission if there is delivery failure to R_2 and R_3.

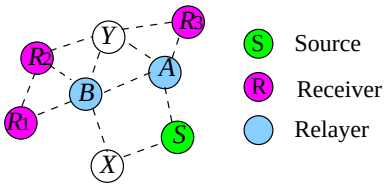


Fig. 3. Multicast packet delivery

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. (1) The memory capacity of mobile nodes is limited. If nodes store every data packet they receive, they can only keep the newest packets. (2) It is unnecessary to store all packets. Simulation results ([3] and [4]) show that multicast routing protocol can deliver safely most of the traffic. Storing successfully delivered packet wastes memory capacity. (3) Nodes mobility causes frequent changes in their roles. A node can be multicast traffic conveyor at a given moment and become a neighbor at the next moment, or is far away from the structure.

C. ARMPIS Protocol Description

Each node in ARMPIS 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 informations 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. ARMPIS 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.

ARMPIS 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 these packet to group receivers, ARMPIS 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 acknowledgment 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.

IV. Performance analysis

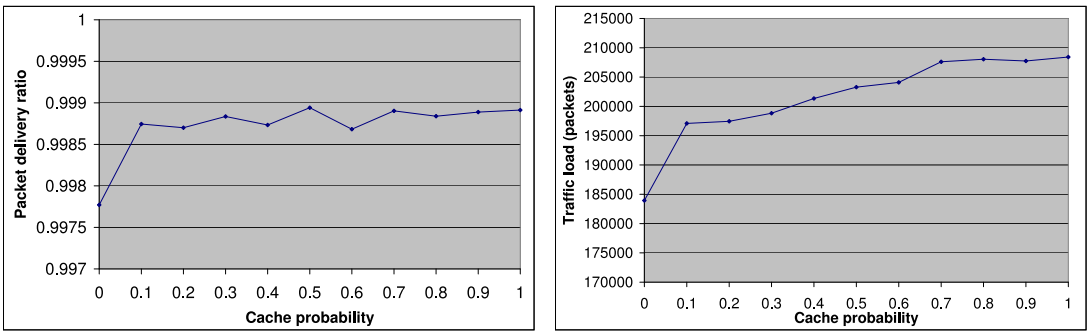
We used $ns2$ [7] to analyze the performance of ARMPIS. In the simulations, ARMPIS was integrated into Multicast Routing protocol with Dynamic Core (MRDC) [4]. MRDC constructs on-demand a group-shared tree rooted on the first source of a multicast session. Multicast tree members send multicast packets on broadcast. The source code can be found in the author's home page [8]. Two other multicast routing protocols, Adaptive Demand-Driven Multicast Routing protocol (ADMR) [9] and the On-Demand Multicast Routing Protocol (ODMRP) [6], are also available on the web site of RICE MONARCH project [10]. Both ADMR and ODMRP are "source based" in the sense that receivers receive packets directly from sources. While MRDC is "group shared" because packets from other sources should go through the core, which increases the difficulty of get recovery packet from sources. We believe that ARMPIS should provide a better performance with ODMRP and ADMR. We compared ARMPIS with a protocol (denoted by ARMP1) in which nodes did not cache packets and feedbacks were sent directly to the source as in [11].

Our simulation modeled a network of 50 mobile nodes placed randomly within a 1000m * 1000m area. Radio propagation range for each node was 250 meters and channel capacity was 2Mbits/sec. 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. 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.

A. 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 while vary the cache probability from 0 to 1 to see the behaviors of ARMPIS. 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.



(a) Packet delivery ratio v.s. cache probability

(b) Total traffic load v.s. cache probability

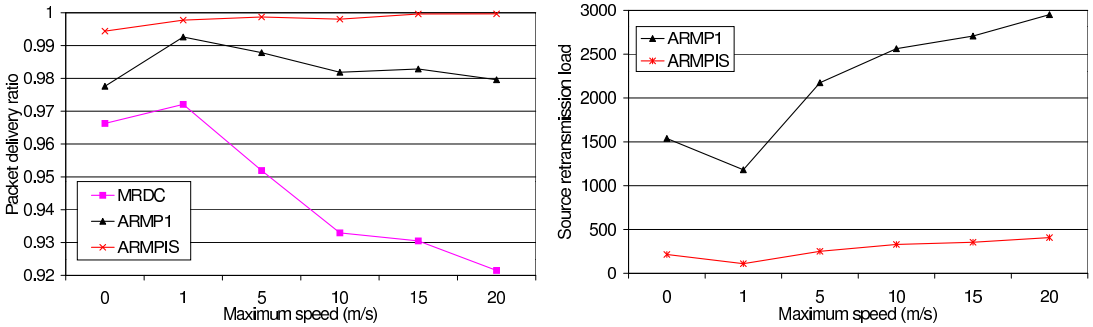
Fig. 4. The impact of cache probability p

Figure4(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 neighbors 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, total traffic load in the network (illustrated in Figure4(b)), which includes original multicast messages and recovery messages, rises along with the increase of cache probability. To achieve low bandwidth consumption and high network throughput, cache probability should keep small. ARMPIS gives the best compromise between packet delivery ratio and bandwidth consumption when cache probability equals to 0.1. In the following simulations, we choose this value as cache probability.

B. 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 and the number of sources is fixed to 4.



(a) Packet delivery ratio v.s. Maximum speed

(b) Source retransmission load v.s. Maximum speed

Fig. 5. The impact of node's mobility

Figure5(a) shows the packet delivery ratio with different maximum speed of these three protocols. The results show that ARMPIS is reliable against frequent topology changes: mobility has nearly no impact on the performance of ARMPIS while frequent topology changes degrade the performance of underlying multicast routing protocol. ARMP1 gives a worse performance than ARMPIS. In ARMP1, 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.

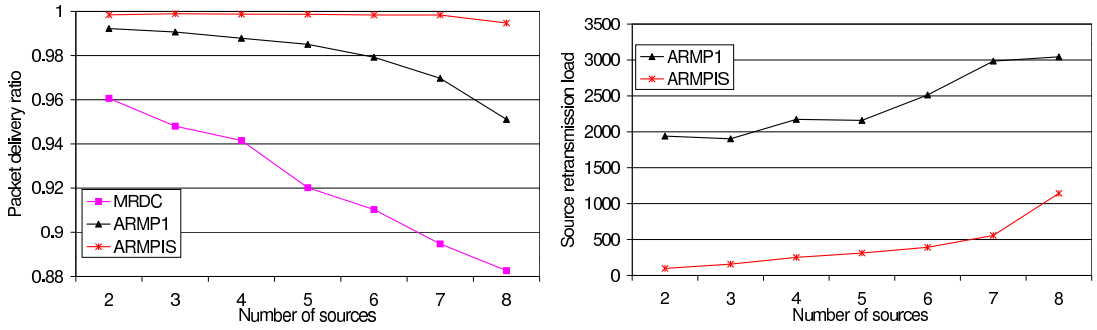
Figure5(b) illustrates the number of packets retransmitted by source. ARMPIS makes source retransmit five times less packets than ARMP1 does. ARMP1 should retransmit more packets as node's mobility increase since MRDC delivers less packets. Compared with ARMP1, ARMPIS distributes retransmission works and have less retransmission failures.

ARMPIS 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 generates significant retransmission load as node's movement speed increases.

C. The impact of traffic load

In traffic load experiment, node mobility speed is moderate with maximum speed at 5 m/s. 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 in Figure6(a). ARMPIS 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. The performance of ARMP1 exponentially degrades. MRDC has a linear degradation even when no congestion happens. This phenomenon is related to the data forwarding fashion employed by MRDC, which works on top of IEEE 802.11. This later does not offer delivery guarantee for broadcast and multicast packets. When MRDC forwards multicast packets, some packets are lost due to hidden terminal problem. And it becomes serious when network load increases. In ARMP1, retransmission initiated by the original source adds considerable extra traffic to the network (see Figure6(b)), which raises collision risk and introduces congestion. That's why the packet delivery ratio decreases more quickly after 5 sources. On the contrary, local recovery mechanism of ARMPIS tries to find the request packet as close as possible to the receivers. As a result, the retransmission load of ARMPIS



(a) Packet delivery ratio v.s. # of sources

(b) Source retransmission load v.s. # of sources

Fig. 6. The impact of traffic load

is less important than that of ARMP1 that makes ARMPIS outperform ARMP1. Since there is no retransmission congestion control, when traffic becomes heavy in the network, the performance of ARMPIS finely degrades.

As demonstrated in Figure6(b), the packets resent by sources in ARMPIS is much less than those in ARMP1. In the case of 7 and 8 sources, each source of ARMP1 retransmits nearly the same number of primary packets while retransmission load of sources nearly no change. This phenomena can be explained by the fact that wireless channel is saturated around sources which prevent them to receive further NACKs. These source do not consequently generate more retransmission load. It also explains why packet delivery ratio of ARMP1 decreases so quickly from 7 sources to 8 sources while at the same time, the degeneration of MRDC is not so significant. On the contrary, in ARMPIS much more NACKs arrive at sources in the case of 8 sources than that of 7 sources. Then, the retransmission load of source is doubled.

V. Conclusions

In this paper, we introduced our active reliable multicast routing protocol with intermediate node support, ARMPIS, to support reliable multicast in mobile ad hoc network. Intermediate nodes in ARMPIS 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 ARMPIS 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. Lehman, Stephen J. Garland, and David L. Tennhouse. Active reliable multicast. In *INFOCOM '98*, pages 581–589, San Francisco CA, March 1998.
[2] T. Speakman, D. Farinacci, S. Lin, and A. Tweedly. Pretty good multicast. Internet-Draft.
[3] E. M. Royer and C. E. Perkin. Multicast ad hoc on-demand distance vector (maodv) routing. Internet-Draft, Jul. 2000.
[4] Shiyi Wu and Christian Bonnet. Multicast routing protocol with dynamic core. In *IST 2001*, pages 274–280, Tehran, Iran, Sep. 2001.
[5] J. J. Garcia-Luna-Aceves and E. L. Madruga. A multicast routing protocol for ad-hoc networks. In *IEEE INFOCOM'99*, pages 784–792, New York, NY, Mar. 1999.
[6] S. J. Lee, M. Gerla, and C. C. Chiang. On-demand multicast routing protocol. In *IEEE WCNC'99*, pages 1298–1304, New Orleans, LA, Sep. 1999.
[7] Kevin Fall and Kannan Varadhan, editors. *ns notes and documentation*. The VINT project, UC Berkeley, LBL, USC/ISI, and Xerox PARC, Novembre 1997. Avaible from http://www-mash.cs.berkely.edu/ns.
[8] http://www.eurecom.fr/w̃us.
[9] Jorjeta G. Jetcheva and David B. Johnson. Adaptive demand-driven multicast routing in multi-hop wireless ad hoc networks. In *the ACM Symposium on Mobile Ad Hoc Networking and Computing*, pages 33 – 44, Long Beach, CA, USA, October 2001.
[10] http://www.monarch.cs.rice.edu/multicast_extensions.html.
[11] Sandeep K. S. Gupta and Pradip K. Srimani. An adaptive protocol for reliable multicast in mobile multi-hop radio networks. Technical report, the 2nd IEEE workshop on mobile computing systems and applications, 1999.