A 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 properties of mobile ad hoc networks, such as limited bandwidth, low memory capacity and high degree of mobility, make reliable multicast protocols for wired networks unsuitable for this kind of networks. To guarantee message delivery to all multicast receivers, the active reliable multicast protocol with intermediate node support (ARMPIS) is introduced. Our contribution is that ARMPIS distributes data cache and retransmission tasks among group members and intermediate nodes, which are not only multicast traffic conveyors but also their neighbors, to improve network throughput even in high mobility cases. 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

Mobile Ad-hoc NETworks (MANET) are multi-hop networks 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 or data distribution during a conference. Many of these applications require error-free many-to-many data delivery. The properties of MANETs, such as limited bandwidth, low memory capacity and high degree of mobility, increase the difficulty of designing a reliable multicast protocol for this kind of networks.

Recently, several reliable multicast protocols have been proposed for MANETs. In [1], the authors suggested an adaptive scheme to support reliable multicast to a set of predefined group members against topology change. A core based shared multicast tree is constructed to delivery messages reliably. In case of fragmentation due to node movement, a ”forward region” is introduced to glue together the fragmented tree and messages are flooded in this region. However, this protocol needs that each recipient sends feedback directly to the source and get recovery messages from the source. Thus, this protocol is not efficient as the size of the multicast group increases. To solve the scalability problem of source-based retransmission, Family ACK Tree (FAT) [2] proposes a hierarchical system based on a tree structure. Each node on the tree is responsible for the reliable transmission of packets to its downstream nodes so that the reliability charge is distributed. For this purpose, each node on the tree temporarily caches the packets and keeps track of on-going traffic. However this protocol becomes inefficient in high mobility networks due to the difficulty of ACK tree maintenance.

In this paper, we propose a receiver initiated active reliable multicast protocol with intermediate node support (ARMPIS) to guarantee message delivery to all multicast receivers in MANET. This protocol requires intermediate nodes to share the message cache and retransmission tasks in order to keep scalable. Considering node's limited memory and node's mobility, the intermediate nodes defined in this protocol refer to all nodes which overhear multicast messages. These nodes thus include not only group members and multicast message conveyors but also their neighbors. When a node receives a NACK packet, it firstly runs local recovery by looking for the requested message in its cache and those of its neighbors. If it finds the message, node sends the message to the corresponding receiver(s). Otherwise, node forwards the NACK message to source. This mechanism makes recovery messages travel a shorter route than original ones and consequently obtains a higher recovery success rate. This protocol needs no other control packet than NACK and independent of unicast routing protocols. The route to source is established by on-going traffic and retransmission tree is constructed during NACK forwarding. ARMPIS may be implemented on either tree-based or mesh-based multicast routing protocols, such as [3–6]. [1]

The rest of this paper is organized as follows. Section II is our design assumption. Section III describes in detail active reliable multicast protocol with intermediate nodes support (ARMPIS). The performance analysis is presented in Sec-

---

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

[1] We use multicast delivery structure to indicate multicast tree and mesh in the rest paper.

tion IV. Finally, section V concludes this paper.

II. SYSTEM MODEL

In this paper, we make 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 group 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 structure (tree or mesh).

III. ARMPIS PROTOCOL DESCRIPTION

Nodes in ARMPIS are active. They not only aggregate and/or suppress feedback to control NACK implosion problem, but also cache multicast messages and perform local recovery for reliable multicasting. The broadcast nature of wireless channel permits neighbor nodes of multicast traffic conveyors to overhear multicast messages passing around them. Thus, these neighbor nodes can help to cache data packets for future retransmission so that network load can be reduced. For example, Figure 1 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.

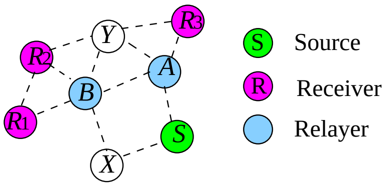

Fig. 1. Multicast packet delivery

In ARMPIS, nodes store packets with a certain probability (denoted by p) to realize distribute 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 multicast message they receive, they can only keep the newest messages.
2. It is unnecessary to store all messages. Simulation results ([3] and [4]) show that multicast routing protocol can deliver safely most of the traffic. Storing successfully delivered multicast messages 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.

Each node in ARMPIS reserves a memory space as multicast message caching buffer. When this buffer is plain, node deletes the oldest packet. Nodes possess a table called relayed packet list to detect multicast duplication. This table contains three fields: group identification, source identification and sequence number. These information identifies a multicast message in the network. Before forwarding a multicast message, node stores the relevant information in the relayed packet list. Packets listed in the table will not be forwarded 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. During data forwarding, a header is added in traffic packets in which there is a field called last hop in their packet header, which carries the last conveyor's identification.

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. When nodes receive a non-duplicate data packet, they fill their relayed packet list to avoid processing the same packet next time and cache received packets with probability p. Nodes also update the reverse path to the source in their route caches. Multicast traffic conveyors update the last hop field of packet header and forward the packet.

In the data repair phase, receivers detect losses primarily by sequence gap in data packets. Nodes aggregate NACKs and delete duplicate requests. Then nodes search missing packet locally by checking their cache and/or sending a broadcast NACK to check the caches of their neighbors. In case of local repair failure, nodes delete the relevant packet information from relayed packet list. Then a unicast NACK is forwarded to the next hop on the reverse path toward the source. These steps repeat until a recovery packet is found. The source is the ultimate responder of retransmission requests. The recovery 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. The way that a node decides to store a passing packet in the simulation is as following. When node receives a non-duplicate data packet, it uses a uniform random function to generate a random number between 0 and 1. If the number is smaller than p, node will store this packet. When p equals to 1, which means nodes store all packets they hear. The result is only the newest packets are stored in cache. On the contrary, when p is set to zero, nodes do not cache any packet.

In the simulations, ARMPIS was integrated into MRDC [4]. MRDC was implemented as described in [4]. This multicast routing protocol constructs on-demand a core based shared tree with the choice of the first source of a multicast session as core to avoid single node failure problem. Multicast tree members send multicast packets on broadcast. Two other multicast routing protocols, Adaptive Demand-Driven Multicast Routing protocol (ADMR) [8]and the On-Demand Multicast Routing Protocol (ODMRP) [6], are developed by RICE MONARCH project for ns2. [2]. Both ADMR and ODMRP are ”source based” in the sense that delivery structure contains direct path from source to receivers. While MRDC is ”group shared” because messages from other source should go through the core, which complicates NACK forwarding and message retransmission. Cooperating with MRDC, we believe that ARMPIS should provide a better performance with ODMRP and ADMR.

We studied the protocol’s performance from three parts, the impact of cache probability, node’s mobility and traffic load by varying three corresponding parameters: the probability p (from 0 to 100%), the maximum node speed (from 0m/s to 20m/s) and the number of sources (from 2 sources to 8 sources). The number of groups was the mode 2 of the number of sources. In node’s mobility and traffic load simulations,we compared ARMPIS with a protocol (denoted by ARMP1) in which feedbacks were sent directly to the source as in [1]. 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.

A. The impact of cache probability p

First, we set the number of source to 6, which means 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.

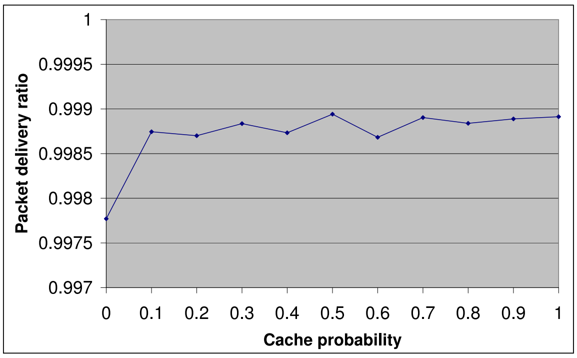

Fig. 2. Packet delivery ratio v.s. cache probability

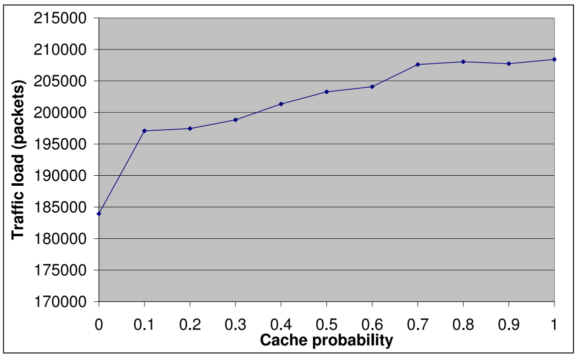

Fig. 3. Total traffic load v.s. cache probability

Figure 2 shows that packet delivery ratio is improved when cache probability passes from 0 to 0.1 then keeps stable. Thus increase cache probability cannot enhance packet delivery ratio. When cache probability increases, the distribution of multicast messages becomes worse and NACK should go further to find the request message. On the other hand, total traffic load in the network (illustrated in Figure 3), which includes original multicast messages and recovery messages, rises along with the increase of cache probability. For the goal of 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 simulation, we choose this value as cache probability

B. The impact of node mobility

Figure 4 shows packet delivery ratio with different maximum speed of these three protocols. The results show that ARMPIS is reliable against frequent topology change and there is no impact on the performance of ARMPIS. MRDC delivers less data packets as node’s mobility increases because this protocol cannot adapt topology changes in time.

---

[2] http://www.monarch.cs.rice.edu/multicast_extensions.html

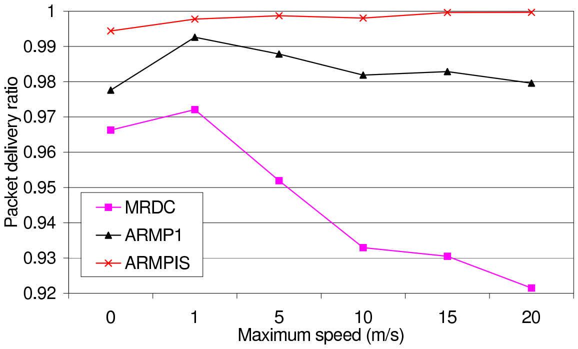

Fig. 4. Packet delivery ratio v.s. Maximum speed

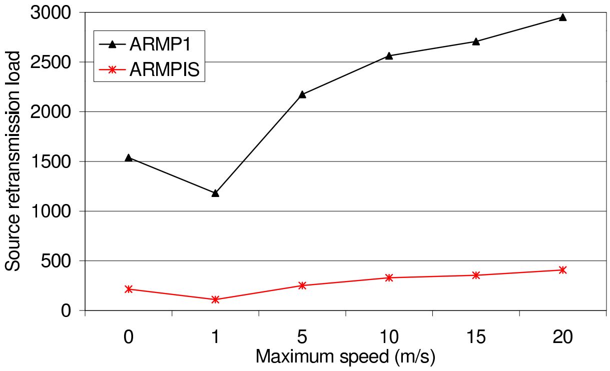

Fig. 5. Source retransmission load v.s. Maximum speed

In MANET, a packet has greater chance of suffering from congestion and topology changes if it goes through more hops. In ARMP1, only source can resend the lost packets, thereby the recovery packets have the same loss probability as the primary one. But local recovery mechanism can decrease this risk by shortening the path for retransmission. In a word, ARMPIS gives higher packet delivery ratio than ARMP1.

Figure 5 illustrates the number of packets retransmitted by source. ARMPIS makes source retransmit 5 times less packets than ARMP1 does. ARMP1 should retransmit more packets as node's mobility increase since MRDC deliver less packets. Compared with ARMP1, ARMPIS distributes retransmission works and have less retransmission failure. It's more meaningful if sources run on battery.

ARMPIS is reliable and scalable as node's mobility increases. This protocol can deliver approximate 100% data packets and can keep the low retransmission load of source in all mobility case.

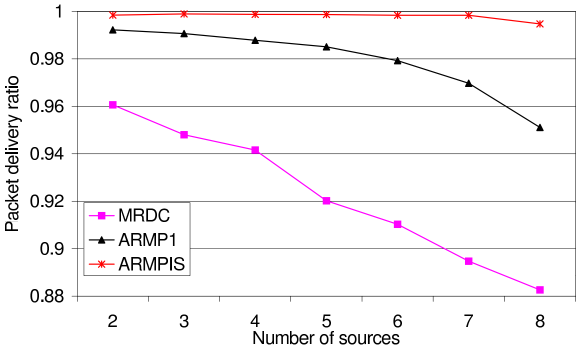

Fig. 6. Packet delivery ratio v.s. # of sources

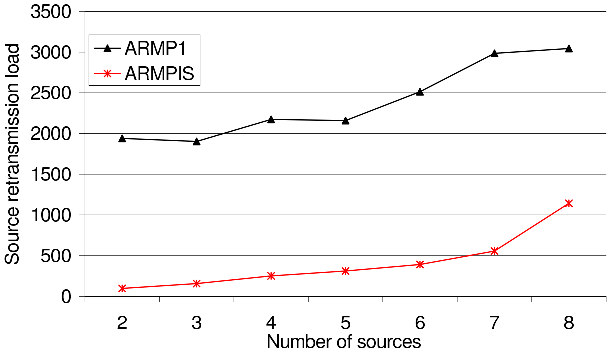

Fig. 7. Source retransmission load v.s. # of sources

C. The impact of traffic load

Figure 6 presents the packet delivery ratio of these three protocols when the number of sources increases. ARMPIS keeps nearly 100% packet delivery ratio till 7 sources and then appear a little degenerative. However, they can transfer more than 99% data packets to all receivers. This shows that they are reliable when traffic augments. The performance of ARMP1 degrades from the beginning and the degeneration becomes quicker and quicker. 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. IEEE 802.11 gives no guarantee delivery for broadcast and multicast packets. While MRDC broadcasts multicast traffic packets. These traffic packets suffer from the famous hidden terminal problem. And it becomes more and more grave when network load increases. In ARMP1, the packets retransmitted by source have the same collision risk as primary ones. Further more, retransmission initiated by the original source adds considerable extra traffic to the network (see figure 7), which raises collision risk and introduces congestion in some cases. That's why the packet delivery ratio decreases more quickly when there are more than 5 traffic flows. Local recovery mechanism tries to find the request packet as close as possible so that the recovery packets have less loss risk. As a result the retransmission traffic of ARMPIS is less important than that of ARMP1. So, ARMPIS outperforms ARMP1. Since there is no retransmission congestion control, when traffic becomes heavy in the network, the performance of ARMPIS finely degrades.

As demonstrated in Figure 7, 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 congestion around sources cause these nodes cannot get further NACKs. It also explains why packet delivery ratio of ARMP1 decreases so quickly from 7 sources to 8 sources when 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 proposed an active reliable multicast routing protocol ARMPIS, to support reliable multicast in mobile ad hoc network. In order to reduce source's retransmission load and achieve scalability in high link loss rate environment, ARMPIS distributes retransmission burden to intermediate nodes. A cache probability is employed to decide store or not a message in each node to reduce message cache duplication and stores as many as possible multicast messages 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 and source's retransmission load is greatly reduced.

REFERENCES

[1] 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.

[2] Ming-Yu Jiang and Wanjiun Liao. Family ack tree (fat): a new reliable multicast protocol for mobile ad hoc networks. In *IEEE International Conference on Communications,ICC 2002*, volume 5, pages 3393–3397, New york, USA, April 2002.

[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] 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.