RELIABLE MULTICAST FOR AD HOC NETWORK WITH INTERMEDIATE NODE SUPPORT

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

Abstract—**Many applications of ad-hoc networks are based on the availability of a reliable multicast protocol. In this paper, we address the problem of reliability and propose a bandwidth-saved method to improve packet delivery of multicast protocol against failures caused by topology changes, congestion, air-interface interference and so on. Due to broadcast capacity, neighbors of multicast tree can help to store and re-transmit multicast data traffic. This protocol contains two parts. In the first part, a normal multicast protocol delivers data packet in group, while in the background, the intermediate nodes including multicast tree members and their neighbors selectively stores packets passing around them. In the second phase, a NACK-based re-transmission protocol tries to recover lost messages. This mechanism reduces the charge of source and congestion around source for re-transmission and saves bandwidth since some work is done by intermediate nodes. Simulation results show that the packet delivery of multicast is significantly improved and in most cases, nearly one hundred percent delivery ratio can be got.**

keywords: Ad hoc networks, multicast routing, reliable multicast, neighbor help

I. INTRODUCTION

Mobile Ad-hoc NETworks (MANET) are a collection of wireless mobile nodes forming a dynamical temporary networks without the use of any existing network infrastructure or centralized administration. Their portability and fluidity make them ideal for applications such as voyage group traveling and data distribution in conference. Many of these applications demand guarantees of data delivery. Different from traditional wireless networks, nodes have to use other mobile nodes in the network as relay if the destinations cannot be reached directly. Because of node mobility and wireless environment, changes of these network topology are frequent and unpredictable limited bandwidth and restrained resource of mobile nodes. These properties make the existing multicast routing protocols very unreliable. In this paper, we focus our interests on this question.

Several protocols have been proposed in recent years to give multicast routing in ad hoc networks. These protocols construct or maintain multicast trees or meshes to establish connectivity among group members. Among them, AMRIS [12], LAM [6], MAODV [10], MRDC[13] and MZR [2] are tree-based protocols while CAMP [5], NSMP [8] and ODMRP [7] are protocols that use mesh structure. Mesh-based approach introduces redundant routes to increase packet delivery ratio against network topology changes, but with the price of bandwidth. However these protocols do not attempt to ensure packet delivery and packet loss is a problem during mesh reconfiguration, a frequent repair activity.

This work is supported in part by HITACHI EUROPE

In wired network, several protocols such as SRM [4] RMTP [9] and PGM [11] have been proposed. Generally, there are two principal methods to give guarantees of packet delivery: source initiated and receiver initiated. In source initiated method, when a data packet arrives at receiver, receiver sends an ACK message back to source, source counts reception of ACKs and re-transmits data packet if there are some ACKs missing. In this method, source should manage group receiver dynamic. This mechanism has famous problem: ACK implosion. To solve this problem, ACK aggregation is employed to reduce the waste of bandwidth. But this optimization requires tree member maintain a downstream group receiver list. Routing message is needed to keep this list recent and acknowledge. In the case of network partition, this method will still try to re-send packets to the receivers while in fact there is no path to them. After network merge, source needs special mechanism to help receivers get lost packets.

On the contrary, in receiver initiated protocols, it is receivers who keep track of multicast data packets. Receivers send request to source or any other receivers for re-transmission once loss detected. In the case of ad hoc network, this is more appreciable. Firstly, source do not need to know which receivers are in the network. Messages are used only for recovering transmission failures. When underlying multicast protocol can deliver the most of packets, receiver initiate method creates less routing overhead than source initiate method.

Ad hoc network could use a special mechanism to provide packet delivery guarantee for multicasting. Retransmission should be shared among all possible nodes in order to avoid congestion around source and to save bandwidth. The broadcast capacity permits neighbor nodes of multicast tree members to hear traffic pass around them. So these neighbor nodes can also help re-transmission. Based on these ideas, this paper proposes a receiver initiated reliable multicast with intermediate nodes support (RMIS). Here, intermediate nodes are multicast traffic conveyors and their neighbors. Intermediate nodes randomly store packets for future re-transmission because it is neither necessary nor possible to store each packet. RMIS would proceed in two phases: data delivery and data re-transmission. In the data delivery phase, any unreliable multicast protocol delivers packets to the group. At the same time, every node, which relays or hears traffic packet, stores traffic packet in their buffer with some probability. In the data re-transmission phase, receivers send re-transmission request toward source once they detect loss. Each node is free to reply if it has some requested packet in its buffer. In this paper, we discuss our implementation of reliable multicast using MRDC [13].

In the rest of this paper, Section II gives a general description of MRDC. Then, Section III describes in detail RMIS. The performance analysis is presented in Section IV. Our conclusions are given in Section V.

II. Overview of MRDC

Our protocol for reliable multicast, RMIS, uses MRDC [13] to deliver data. MRDC is designed to provide a tradeoff between data transmission efficiency and routing overhead. This protocol constructs on-demand a group-shared tree. The core of the tree is the first source of a multicast session.

Each node in MANET possesses a *multicast routing table* (MRTable) which stores multicast routing information. The existence of an entry in the table, which corresponds to a multicast session tree, is the indication of the traffic presence in the group.

The control part of MRDC consists of two aspects: Tree construction and Tree maintenance. Tree construction is the aspect in which a core is selected and advertised to the network. Group members join the tree. Tree maintenance is the aspect where tree members detect broken branches and repair the failure to continue multicast traffic delivery in multicast tree (MTree). Tree maintenance aspect also takes care of departure of group members.

MTree construction is based on the following mechanism: core, the first source of a multicast session, broadcasts a *Core Advertisement* (CA) message $< CA, @group, @core, seq., @up >$ to the network, which constructs reverse path to core at the same time. A node that is interested in the multicast group executes a **RAR/RAA** procedure to join MTree. In this procedure, the node sends a Route Active Request (RAR) message $< RAR, @group, @core, seq., @down >$ to its potential upstream node, which is the next hop on reverse path toward core. The first MTree member, which receives RAR message, replies a Route Active Acknowledge (RAA) $< RAA, @group, @core, seq., @up >$ to activate all potential downstream nodes. Then a branch is added into MTree.

The tree maintenance aspect in MRDC contains: link failure detection, local route recovery, periodical Mtree refreshing and group members departures. A link failure is detected if a MTree member cannot forward packet to its MTree neighbor. Then local route recovery is immediately executed in order to avoid blocking data delivery. The upstream Mtree member of the broken branch locally broadcasts a *Joining Invitation* (JI) message $< JI, @group, @core, seq., @down >$ to invite the downstream node to rejoin the MTree. Upon receiving JI, the downstream node takes part in Mtree by running RAR/RAA procedure. The aim of periodical Mtree refreshing is to improve date transmission efficiency according to new topology. Core is in charge of initiating periodical Mtree refreshing. During each period, core computes a new reference and broadcasts a CA, which also refreshes reverse path for the core. Once a node receives a non-duplicated CA, it updates the information of the route entry, empties the field of downstreams and sets the entry to inactive. Group members run RAR/RAA procedure to reconstruct MTree. Group members could leave the group implicitly without execution RAR/RAA procedure. They can also explicitly leave tree by sending a message to their upstream nodes. When core wants to leave group, it checks whether there is another source in MTree that can become a new core. If this core is the only source, it dismisses the MTree. Otherwise, the new core is in charge of sending periodical CA.

III. RMIS Protocol Description

A. System model

This paper supposes that mobile nodes share a common wireless media for communication. When one node sends a packet, all other nodes in its radio coverage can detect the signal, and some of them can correctly calculate this packet. Nodes possess a FIFO buffer for cache multicast traffic packet. Topology change, radio interface interference and congestion provoke not only data packet loss but also routing packet loss. Before sending data packet to group, source assigns a consecutive sequence number into packet, with which receivers can detect loss. Then MRDC delivers multicast packet to group receivers. Receivers check packet arriving and send a list of packets toward source if they think there are some packets lost. Tree members look for the packets firstly in their buffers, and then query their neighbors. They send the found packets to receivers and report rest list to their upstream node. In the worst case, receivers can get lost packets from the source. Receivers keep multicast traffic packets that they correctly receive during the multicast session.

B. Data delivery phase

MRDC constructs and maintains a multicast tree for multicast traffic delivery. When a tree member sends traffic packets to its tree neighbors, its other neighbors may also receive these packets. These intermediate nodes insert multicast traffic packets, which they receive, into theirs FIFO buffer with a certain probability (said p). There are some reasons why we use such a probability. The first reason is the memory capacity of mobile nodes is limited. If nodes store every data packet they receive, they can only keep the newest packets. The second reason is no need to store all passed packets. The simulation results of MRDC shows this protocol can distribute greater part of traffic. Storing successfully delivered packet wastes memory capacity. The third reason is to avoid caching same packet in neighborhood.

Figure 1 shows an example of data caching. A multicast

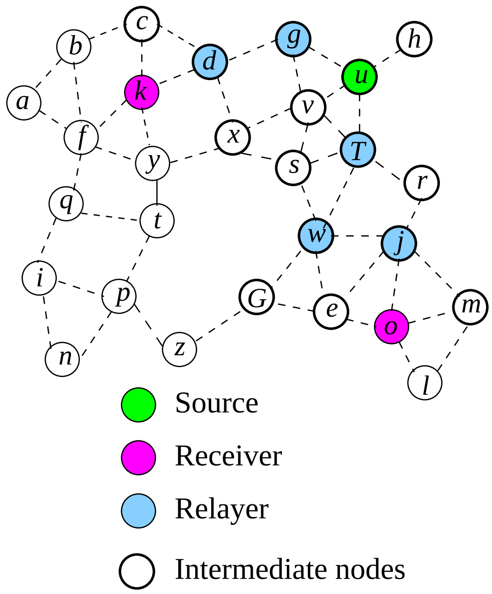

Fig. 1. Cache and deliver data packets

tree is constructed by MRDC. The root of tree, node u, is the source of group, it sends traffic with assigned sequence number to multicast tree. Node g, d, T, W and j are tree members, which relay multicast traffic to group receivers node k and o. The tree members and all their neighbors (node c, x, v, s, r, k, h, G, e and m), store packets in their cache with a probability p. For example, node W has three neighbors. When node W relays a traffic packet from node T to node j, three tree neighbors (node e, s and G) hear this packet. Supposing all these three nodes correctly receive this packet, the possibility of at least one of these nodes and node W itself storing this packet is $1-(1-p)^4$.

C. Data Re-transmission phase

Checking the sequence number of last received packet and that of new arrived packet, a receiver can tell whether there is loss. Once a loss detected, receiver sends a Re-Transmission Request (RTR message) to its direct upstream. A RTR message has the following five fields:

- Group Address: the address of the multicast group
- Source Address: the address of the source node of the traffic
- Receiver Address: the address of the node sending the RTR message
- Number Lost: the size of the Lost Array
- Lost Array: an array of the sequence numbers of traffic packets that the group receiver believes it has lost

Upon receiving a RTR message, tree member firstly checks whether its buffer contains the required packets. It also inquires its neighbors, excluding the downstream node that forwards RTR message and the upstream node, for the packets that it does not find in its buffer. Exclusion the downstream node is because there's no request packet in its buffer. Exclusion the upstream node is because its buffer will be checked if the node does not find packets. After that, this node sends all packets that have been found to the receiver. If there are still some packets not found, this node re-organizes the lost packet list and relays rest request array to its upstream node. Request aggregation is employed to avoid asking same packets to upstream node.

In the same example of figure 1, the possibility that a recent packet can be found in node W and its neighborhood, except node l and T, is $(1-(1-p)^4)$. As a result, the possibility of re-transmission request pass to the upstream node of node W for this packet is $(1-p)^4$. There are totally 9 intermediate nodes between group source $node u$ and group receiver $node o$. Compared to pure source re-transmission mechanism, our idea can reduce at least $(1-(1-p)^9)$ work of source and usage of source's bandwidth in this example.

A RTR message can take maximum L sequence numbers for re-transmission request. Node chooses the L biggest sequence numbers if it has more than L data packets. Because a newer packet has a bigger sequence number, and there is a greater possibility to find this packet in the buffers of intermediate nodes. After sending a RTR message, a node keep silent for a moment. That permits upstream nodes looking for and re-transmit packets. After a while, the node review whether there are still some requests. If so, node repeats the above procedure.

IV. Performance Analysis

To analyze the performance of RMIS, we use a network simulator, $ns-2$ [3]. $ns2$ is a discrete, event-driven simulator developed by the University of California at Berkeley and the VINT project. MRDC is implemented as described in [13]. RMIS used MRDC as underlying multicast protocol.

A. Simulation Environment and Implementation Decision

$ns-2$ provides support for simulating multi-hop wireless networks complete with physical, data link layer and IEEE802.11 MAC protocol (for detail description, see [3] and [1]). The radio interface worked like the 914MHz Lucent WaveLAN DSSS radio interface, the bandwidth of the wireless medium was 2Mbps and transmission range was 250m.

The simulation models use 50 wireless nodes forming an ad hoc network. These nodes move in a 1000m x 1000m flat space. The movement model of nodes is the random waypoint model [1] without pause time. A number of movement scenarios are used as input to simulations. Each movement scenario file determines movements of 50 nodes and the speeds of mobile nodes are uniformly distributed up to a maximum speed. Each data point in figures presents an average of 10 movement scenarios of a maximum speed since the performance results are sensible to node mobility and network topology. The maximum node speed is varied from 0m/s to 30m/s. Network partition and merger sometimes appear in movement scenarios.

There are two multicast groups with ten group members. Group members join the multicast session at the beginning of the simulation and remain as members throughout the simulation. Constant Bit Rate (CBR) traffic source is used in simulations. One members is selected as the CBR source in each group. Multicast sources generate 4 512-byte data packets per second. In the 900 seconds simulation time, each source starts to send packets at 120sec and generates 3000 data packets.

The period of tree refreshment of MRDC is 5 seconds. The capacity of buffer is set to permit a node store 64 data packets. To aggregate NACK at receiver, the time between two RTR message sent by a receiver should be greater than 10 seconds. The probability p of nodes storing a packet is 10%. What we want is to get the greatest possibility of a packet is store only once by a router and its neighbors. This probability can be calculated by $n*p*(1-p)^{(}n-1)$, where n is number of nodes including router and its neighbors. When $p = 1/n$, this formula gets maximum value. One node has on an average $N * \pi * R^2/(x * y)$ nodes in its radio coverage region, where N is the number of nodes in the network, R is coverage radius and x, y is the size of square. With the value defined above, in our simulation scenario, one node have about 9 neighbors. As a result, we choose 10%. In the real world, a node can detect how many neighbors it has by some mechanism as hello message or beacon to decide the probability.

We compare the packet delivery ratio of RMIS and MRDC. Packet delivery ratio is the number of data packets correctly delivered to receivers over the number of data packets sent by sources.

B. Performance Analysis

Figure 2 shows packet delivery ratio with different maximum speed respectively for both MRDC and the RMIS. The results show that RMIS consistently performs better than the underlying multicast protocol. Not only is the average packet delivery ratio higher but the variation in the number of packets received by different group member is also significantly lower. In all cases, our protocol gives nearly 100% packet delivery. Even when the network is table, MRDC cannot send packets to all receivers. In fact, these packet delivery failures are caused by packet collision in MAC layer. RMISC resolves these problems perfectly. However, in certain case, this protocol cannot recover all failures. For example if network partition presents till the end of transmission, receivers do not hear new data packet and consequently cannot detect loss.

Figure 3 and 4 present the improvement of variation of packet receive ratio by the group members in detail. There are in total 180 (9 receivers per group * 2 groups * 10 simulations) samples of packet receive ratio for each maximum speed. If we do not consider the extreme situation, RMRDC can give a guarantee of delivering 99% packets to all receivers even when nodes move very quickly (30m/s). While MRDC delivers about 85% or less packets to some receivers with the same maximum movement speed.

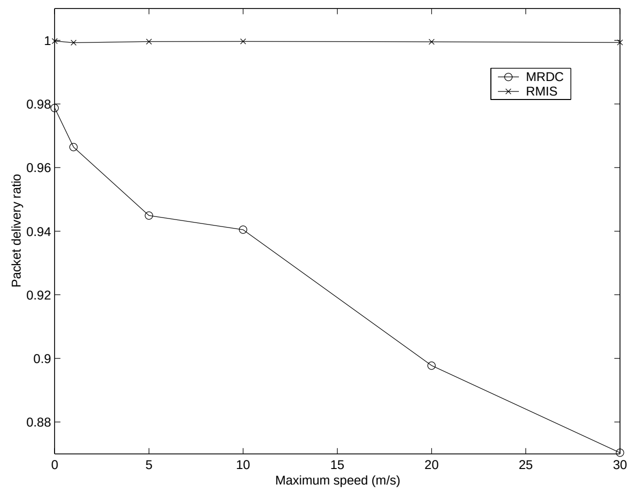

Fig. 2. Packet delivery ratio vs Maximum Speed

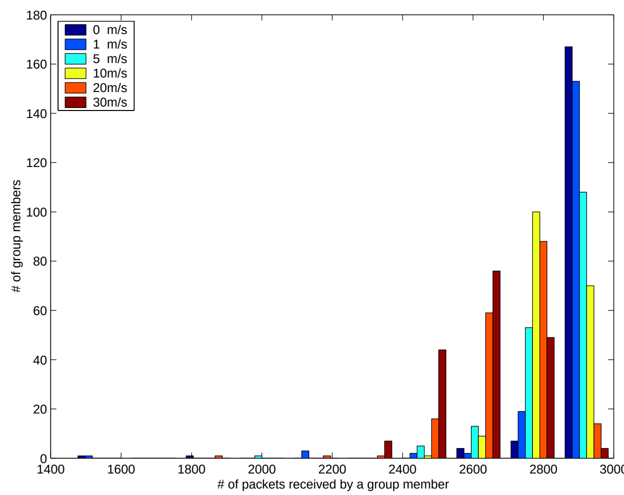

Fig. 3. Variation of packets received by the group member in MRDC

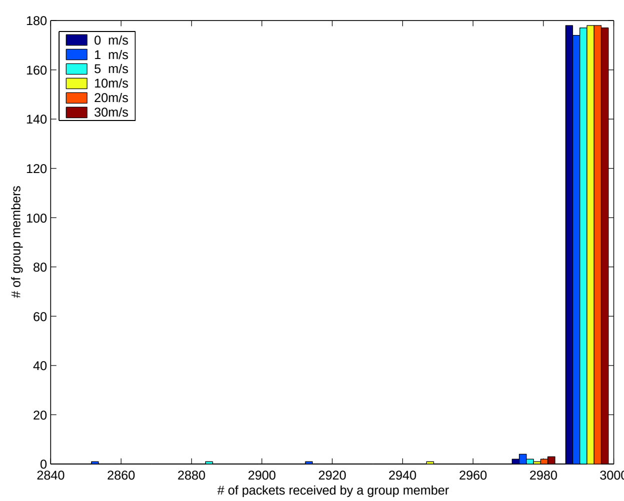

Fig. 4. Variation of packets received by the group member in RMIS

V. CONCLUSIONS

In this paper, we proposed a reliable transmission mechanism by using buffer of intermediate nodes, RMIS. Intermediate nodes are multicast traffic conveyors and their neighbors. Nodes store packets passed around them with a certain probability as memory of mobile nodes in ad hoc network is limited. This protocol is aimed to save bandwidth and reduce the charge of multicast source. The idea is to make intermediate nodes do some re-transmission work, which can consequently alleviate congestion around source. The simulation results show that it is reliable; it greatly improves data packet delivery ratio to nearly 100 % even when nodes move very quickly.

REFERENCES

[1] Josh Broch, David A. Maltz, David B. Johnson, Yih-Chun Hu, and Jorjeta Jetcheva. A performance comparison of multi-hop wireless ad hoc network routing protocols. In *The 4th International Conference on Mobile Computing and Networking(ACM MOBICOM'98)*, pages 85–97, October 1998.

[2] V. Devarapalli, A. A. Selcuk, and D. Sidhu. Mzr: A multicast protocol for mobile ad hoc networks. Internet-Draft, Nov. 2000.

[3] 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.

[4] Sally Floyd, Van Jacobson, Steve McCanne, ChingGung Liu, and Lixia Zhang. A reliable multicast frame work for light weight sessions and application level framing. In *'95 Symposium on Communication Architectures and Protocols (SIGCOMM). ACM.*, Cambridge MA, August 1995.

[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] L. Ji and M.S. Corson. A lightweight adaptive multicast algorithm. In *IEEE GLOBECOM'98*, pages 1036–1042, Sydney, Nov. 1998.

[7] 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.

[8] Seungjoon Lee and Chongkwon Kim. Neighbor supporting ad hoc multicast routing protocol. In *the First Annual Workshop on Mobile Ad Hoc Networking & Computing, MobiHOC 2000*, pages 37–44, Boston, August 2000.

[9] Sanjoy Paul, K. K. Sabnani, J. C. Lin, and S. Bhattacharya. Reliable multicast transport protocol (rmtp). *IEEE Journal on Selected Areas in Communications, Special issue on Network Support for Multipoint Communication*, 15(3), April. 1997.

[10] E. M. Royer and C. E. Perkin. Multicast ad hoc on-demand distance vector (maodv) routing. Internet-Draft, Jul. 2000.

[11] T. Speakman, D. Farinacci, S. Lin, and A. Tweedly. Pretty good multicast. Internet-Draft.

[12] C. W. Wu, Y. C. Tay, and C. K. Toh. Ad hoc multicast routing protocol utilizing increasing id-numbers (amris) functional specification. Internet-Draft, Nov. 1998.

[13] Shiyi Wu and Christian Bonnet. Multicast routing protocol with dynamic core. In *IST 2001*, pages 274–280, Tehran, Iran, Sep. 2001.