Immediate Family ACK Tree for Reliable Multicast in Mobile Ad Hoc Networks

Tzu-Chiang Chiang, Chan-Yu Hsu, and Jia-Lin Chang

***Abstract*—In this paper, an efficient protocol, called Immediate Family ACKTree (IFA), is proposed to report correct data reception or a negative acknowledgment to request retransmissions for feedback.. Each reliable multicast protocol needs a recovery mechanism to deal with occasional losses, duplications and failed delivery of datagram. IFA use a directly-related based recovery mechanism that localizes ACKs and retransmissions to avoid forward and feedback flooding. Our approach constructs an immediate ACK tree to three generations of nodes and uses efficient Fast-Upstream recovery algorithm to maintain reliable multicast and reduce overhead. The results show that IFA achieves better performance a high reliability in a mobility mode, a scalability performance in a large ad hoc network, and high delivery efficiency in RTQ multicast due to the speedy of recoverable service.**

***Index Terms*—Immediate family ACK tree (IFA), mobile ad hoc network, reliable multicast.**

I. Introduction

Each reliable multicast protocol needs a recovery mechanism to deal with occasional losses, duplications and failed delivery of datagram. The recovery mechanism may use an acknowledgment (ACK) to report correct data reception or a negative acknowledgment to request retransmissions for feedback. It is a challenge for reliable multicast protocols to avoid feedback implosion. The number of feedback messages leads to a heavy burden on data sources and causes more packet congestion and losses. There has been much work on reliable multicasting, including ACK-based ender-initiated protocols [3], ring-based protocols [4], [5], and tree-based protocols [6]. As above, Some mobile nodes having joined a multicast group possess the service quality of reliable multicast if they have fixed hosts. Thus, a mobile node occurs packet losses or duplications seldom and receives data streams continuously and reliably due to roaming.

The challenge of reliable multicast for mobile ad hoc networks is to cope with node movements even while forwarding packets. Node movement may cause some nodes to be disconnected and miss some multicast packets, even if they will eventually be glued back to the tree.

Reference [7] studies multicast in ad hoc networks. This source ACK protocol essentially needs each member to send feedback (to acknowledge reception of packets or to request for retransmissions) directly back to the source. Thus, it suffers from the problem of feedback implosion. Reference [3] adopts a tree-based recovery mechanism to localize retransmissions to provide reliable multicast for mobile ad hoc networks, which requires each member of family group to maintain its subgroup when corresponding nodes move. It results in suffering from overhead of maintenance.

In our approach, to cope with node movements and retransmission fast recovery, IFA constructs an lineal path on which each node maintains reachable information to three generation of the directly-related member of its family. i .e., the node's parent, grandparent, the node itself and its children. When the tree is fragmented due to a departed node, the fragments will be glued back using the localize Fast-Upstream recovery mechanism to recover missed packets that have been multicast to the group during fragmentation. IFA is major concerned with the transport layer issue (i.e., reliable multicasting).

II. The Immediate Family ACK Tree (IFA) Protocol

The proposed IFA is a directly-related based reliable multicast protocol which runs on any multicast trees established by the underlying multicast routing protocol such as [8]. The typical approach of tree-based reliable multicast protocols is organized "region hierarchy" to manage acknowledgment (ACK)s and to localize retransmissions. Each node in a region acknowledges the reception of packets or requests retransmissions to the agent node. For reference [3], lost packets are recovered by a tree-based local recovery scheme, but it suffers from overhead for each member of family group to maintain its subgroup when corresponding nodes move. In summary, IFA works as a typical tree-based mechanism when nodes are connected. IFA reconnects the fragments back to the tree based and recovers the loss by using the Fast-Upstream recovery mechanism for avoiding flooding when node movements.

A multicast tree has been constructed by an ad hoc multicast routing protocol such as [2]. A immediate family is retrieved from the underlying multicast tree. On the IFA tree, the parent node (i.e., immediate upstream node) serves as the reliable agent for its child nodes. Each child node cumulatively acknowledges the reception of packets or requests retransmissions to their parent node. Once the parent node has moved away, the new parent found by the Fast-Upstream recovery mechanism will become the new reliable agent. The Fast-Upstream mechanism limits the flooding of messages to the upstream and the immediate

Manuscript received August 14, 2012; revised September 28, 2012. This work was supported by the National Science Council of Republic of China under grant NSC 100-2221-E-029 -016.

Tzu-Chiang Chiang is with the Department of Information Management, Tunghai University, Taiwan, ROC (e-mail: steve312kimo@thu.edu.tw)

Chan-Yu Hsu is with the Department of Information Management, National Kaohsiung First University of Science and Technology.

Jia-Lin Chang is with the Department of Transportation Technology and Management, Kainan University.

vicinity of the nodes by underlying multicast routing protocol. To recover the packet losses as soon as possible, the node may relay retransmissions to its neighbor nodes along the upstream ACK tree to find a new grant agent which is able to recover by roll back process. In the worst case, this retransmission node may be the root (i.e.,the source). As such, we can avoid sending retransmission requests all the way back to the source. Once a node departed, all the children of the departed node became orphans. Each orphan node with its descendants attempts to glue back to the tree independently. Therefore, using RTQ messages communicate vicinity nodes for reconnected back to the tree.

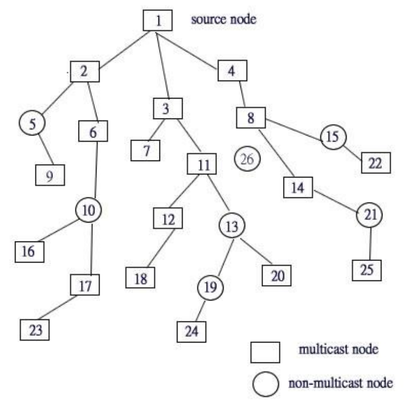

Fig. 1. An example of a family ACK tree.

TABLE I: IMMEDIATE FAMILY ACK TABLE

node	7	12	13	14	18	19	20
GID	1	3	8	4	7	14	14
PID	3	7	14	8	12	13	13
CIDset	12	18	19,20	13	18	24	-

A Retransmission request (RTQ) is used by a node to report loss packets to its reliable agent and to request a retransmission. In the normal state (i.e., no node is disconnected), the child node unicasts an RTQ to its parent node. When a node is disconnected, each child of the departed node tries to find the reliable parent to be glued back. The selected parent which should be its neighbors transmits the lose packets from the rollback buffer to the child. Once detecting the lost packets not reparable by the receiving node, it probably exists two situations. It should relay the RTQ to find the new grant node along its upstream of the ACK tree. Another is that the RTQ message forwarding is terminated with a time-to-live (TTL) value. The TTL value limits the range of forwarding messages. An RTQ is resent if the node has not received the requested packets within a predefined timeout. The TTL value is incremented by one to broaden the range of message forwarding and to increase the probability of reaching a node that can grant the RTQ and retransmit the packets. The grant node is able to recover the loss for the node by a rollback process. Thus, the Fast-Upstream recovery algorithm will be adopted to limit downstream seeking.

III. THE FAST-UPSTREAM RECOVERY ALGORITHM

In this section, the proposed IFA protocol is described in details. IFA is a directly-related based reliable multicast protocol for mobile ad hoc networks. It runs on any multicast trees established by the underlying multicast routing protocol. The typical approach of ACK tree reliable multicast protocols works as follows. The system maintains an acknowledgment (ACK) tree organized in a way of "region hierarchy" to manage ACKs and to localize retransmissions. Each node in a region cumulatively acknowledges the reception of packets or requests retransmissions to the agent like a parent. Each node may be in one of two states: normal state and repairing state.

1) Usually, a node is in the normal state.
2) Upon detecting the departure of the parent, the node enters the repairing state.

If a node in the normal state receives a multicast packet reliably, the packet is cached with a predefined timer.

When a node departs, The nodes notified may include the parent node, the child nodes. The departing node forwards the packets in its cache buffer to its children to continue running for its descendant nodes for a period of time. Once a node has departed, all the children of the departed node became orphans. Each orphan node with its descendants attempt to glue back to the tree independently. During the "glue back" process, the children of an orphan node keep its descendant nodes from leaving the group reluctantly.

The orphan node sending the RTQ starts a predefined timer. If the orphan node has not received the grant packets upon expiry of timer, a new RTQ with a larger TTL is resent. This repairing process continues until a repairing node enters the normal state. An orphan node has reconnected to the trees, it reconfigures its ACK table and identifies the missing packets received.

The RTQ including source id, parted node id it's PID and packet sequence etc. was multicast to its neighbors. The node received the RTQ, it will compare its GID with PID of RTQ. If the receiving node's grandparent is the parted node, the receiving node should be the descendants of the parted node and is located in the downstream of the parted node. The multicast transmissions are usually transmit from up (source node) to down (multicast nodes). For reduce the bandwidth congestion, the node received RTQ will discard it for avoiding forwarding to downstream. If the receiving node is in the repair state, then it suspends the RTQ until it became to normal state. If the receiving node is a multicast node and could grant the RTQ (could rollback buffer to meet sequence number of RTQ), then stops forwarding RTQ and replies back to the request node by following the reverse path. The requesting node receive the grant message and assent reconnected to the reply node, then the connection is constructed. Some corresponding ACK table will be modified. It illustrates the connection was connected as soon as possible for reduce the time of the packets lost. If the node is not a multicast node and is not belongs to ACK tree. it only forwards the RTQ to its neighbor multicast nodes. If it is in ACK tree, it set the PID value to its PID value for avoiding forwarding to its descendants and relay RTQ to its neighbors. Continue these step until the grant node is found. In the worst case, the request will go to the root. The algorithm is used to backtrack up searching one more level of the ACK tree until a node able to repair is found.

For avoiding feedback implosion, the TTL value limits the range of forwarding messages. If no repair is received within the timeout period, the requesting node will resend a request with a larger TTL. Increasing TTL has the effect of widening the scope of multicasting. As the number of nodes involved in request multicasting increases, the possibility of reaching a node that is able to grant the RTQ and retransmit the packets also increases, at the expense of unnecessary bandwidth consumption.

Upon the granting node reply back, the requesting node or the medium node just assent the first return grant message. After the grant node is found, the child of the departed node with its descendants reconnects back to the multicast tree, from where the loss can be recovered.

```
The Fast Upstream Recovery Agorithm as follows:
Mtree : the set of nodes belong to multicast tree combined by the source node
Mgp : the set of nodes which is a multicast node and is the number of the
      multicast group
PKTseq : packet sequence number
RTQ : retransmission request message
S: source node

Request Algorithm (RTQ(PID(rq), S, rq,TTL,PKTseq))
   //A node send a RTQ(PID(rq), S, rq,TTL) to it neighbors.//
The node(i) receives the RTQ message.
If TTL = 0 then stop //time out then stop//
If node(i) is < repair state>
 Then suspend RTQ till node(i) is <normal state> within TTL time.
 Else
   If Gid(i) = PID(rq) then discard RTQ
  //the grandchildren of the departed node//
  //avoid forwarding to its descendants//
If node(i)  $ Mgp
   then
      If PKTseq <= top(buffer) of node(i)
         Then return grant(i) to node(rq)
                        by reverse path
    Else PID(rq)=PID(i); backtracking
 // backtracking level by level up toward the sourceuntil a node able to
answer the request is reached//
 //avoid sending RTQ to its descendants
 else
   if node(i) $ Mtree
      then PID(rq)=PID(i); send RTQ(i)
      ////request is encapsulated //
     else
       if neighbor(i) $ Mgp
      then only send RTQ to neighbor(i)
      else discard RTQ
endif
 // If no node grant the request
 Then increase the value of TTL
 Continue Run requesting algorithm//
Repairing algorithm
Repair (GRANT(i,RTQ(PID(rq), S, rq,TTL)
 Node(j) return GRANT(j,i) message to node(i)
 If node(i)=node(rq)
   glue the multicast tree by following the reverse path from the grant node.
 Else the GRANT is the first grant message
   then return back by following reverse path
 endif
 modify the ACK table.
```

Our algorithm performs well in mobile ad hoc networks. This approach is fast dynamically recovery and less maintenance overhead.

The repair will also be multicast to its descendents for modifying the ACK table. For nodes sending requests or repairs, this recovery scheme provides a mechanism to prevent duplicate grant from duplicate repairs (i.e., only the first received grant is used to repair the connection). As a result, the fast recovery approach make the packet loss not be reinforced by delaying recovery. This mechanism allows local loss recovery and reduces unnecessary use of bandwidth (avoiding downstream process for any receiving request node) by limiting the scope of multicasting requests and repairs.

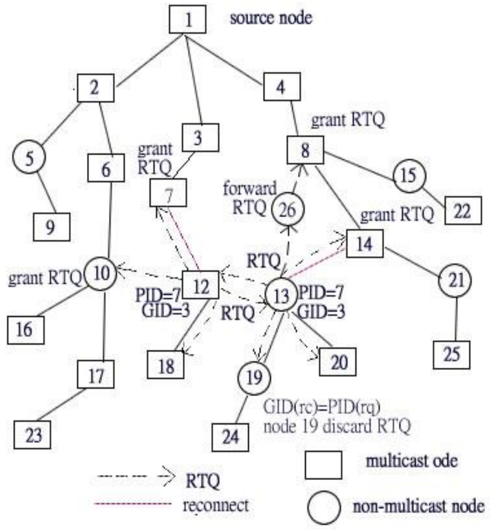

Fig. 2. The original ACK tree.

Fig. 2. Shows node 11 is gone from the original ACK tree. Node 12 and node 13 become orphan node. Node 7 and node 14 grant the RTQ and reconnect the node 13 with its descendants and node 12 with its descendants respectively. After then, modify the corresponding ACK table. Upon detecting the departure of node 11, node 12 and node13 with its descendant needs to reconnect back to the tree. Node 7 and node 14 identifies the packet sequence numbers in buffer offset, then it grant the RTQ back to node 12 and node 13 respectively. Node 12 and node 13 received the first acknowledgement, then they glue back to node 7 and node 13 for packets loss and multicast transmission after then. For node 19 and node 20, Once receiving the RTQ, check the RTQ with PID is equal to its GIDs. They discard the RTQ to prevent downstream seeking.

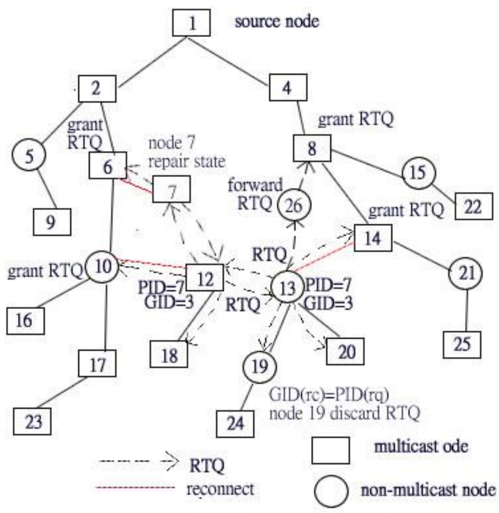

Fig. 3. The ACK tree maintenance of IFA.

Fig. 3. Shows the ACK tree maintenance of IFA. Both parent node 11 and grandparent node 3 are gone, node 7

become orphan node and is in the repair state. If node 3 departs at the same time as shown in Fig. 2, node 7 enters the repairing states and suspends the RTQ. Upon receiving an RTQ, node 10 (multicast tree node but not multicast node) updates the PID of RTQ to node 6 for prevent it children like node 16 and node 17 from continuing to forwarding the RTQ. Upon receiving the RTQ, compare PID of RTQ with its GID, node 17 discard the RTQ. Here, node 6 grant the request by roll back the buffer, after it receiving node 12's acknowledgement, then retransmits packets back to node 10 and relay to node 12.

IV. Performance Evaluation

This section describes the simulation conducted to comparison of IFA(our approach) , FAT(family Ack tree)[3] and source ACK approach. We investigate the behavior of IFA and compare IFA with other protocols in terms of reliability, scalability, and delivery efficiency.

In the simulation, we assume that packet transmissions are error-free and all losses are caused by node movements. We consider a 100-by-100 mesh in which nodes are roaming in the mesh during the simulation. Initially, we spread mobile nodes in the mesh randomly, where varies from 100 to 400 parted by two types of nodes. One is multicast nodes are about 90 to 360, the rest of nodes are non-multicast nodes. Each node has a transmission range of 10 m. Actually; there exist some nodes not to belong to the multicast tree. To simplify the type of nodes, a multicast delivery tree is rooted at the source and spans over all other nodes. The sender generates data packets at a constant rate of one packet per second.

A node moves at a speed between 10 and 20 m/s. We use six different pause times in the simulation: 10, 13 16, 20, 40, and 80 s. We define "mobility" as the inverse of the mean time that a node stays at a location. The shorter the pause time, the higher the mobility is.

To simplify the comparison, a parameter called the reliability index is defined as the ratio of the number of the granted requests (retransmitted successfully) to the total number of the received requests during a period of time. The larger the reliability index, the larger the number of lost packets that can be recovered, and the better the performance. The RTQs granted are regarded for calculation of the reliability index.

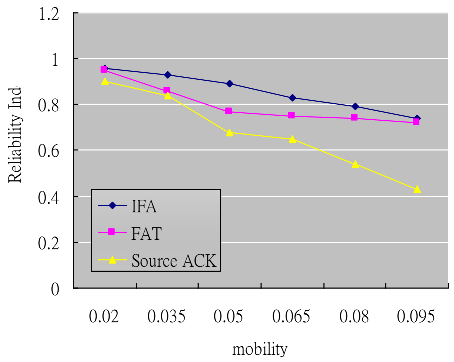

Fig. 4. Reliability index versus mobility.

In Fig. 4 show that source_ACK protocol has low reliability when node mobility is higher. Because in the source ACK, all mobile node sends an RTQ directly back to the source, causing the source to become a bottleneck and disrepair during a period time. In the FAT protocol, the further the node backtracks the former parent, the much higher the packets loss. It is better than the source ACK. But compare with our approach, the mobile nodes still backtrack to its former grandparent and uncle (disconnected with the node) will spend seeking time. In our approach, we use fast and dynamic Fast-Upstream repair mechanism to seek nearer multicast node to grant the RTQ and construct the new connection between it and the request node. Thus our approach is litter better than FAT approach. This simulation is performed to measure the bandwidth consumption which index is defined as the total number of retransmission requests multiplied by the average path length (the number of hops) by RTQ packet, normalized by the total number of mobile nodes. The pause time is fixed at 20 s.

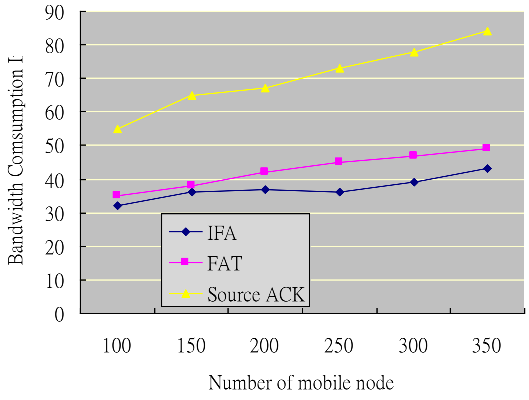

Fig. 5. Bandwidth consumption.

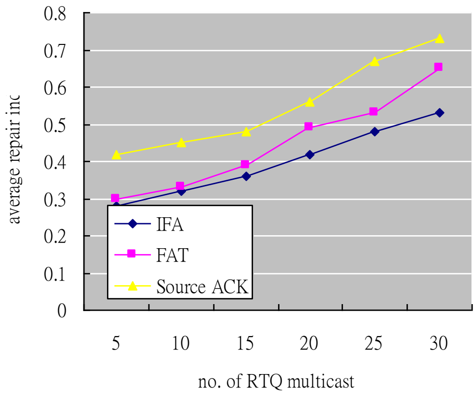

Fig. 6. Average repair index versus no. of RTQ multicast

Fig. 5 shows that the bandwidth consumption index a little increases as the number of nodes increases, thanks to retransmissions being localized by each request node. Thus, FAT and IFA are provided reliable multicast services even in a large ad hoc network. In the IFA, our approach fast looking for nearer and upstream multicast node for retransmission results in lower overhead than FAT. This below experiment is conducted to observe the relationship between the repair time index and the number of RTQ. The average repair index is defined as the ratio of the number of the granted requests (retransmitted successfully) multiple the period of time, 10,20,30,40 s moralized by the total number of the received requests. To measure the repair time of RTQ multicasting, we count the different times that an RTQ message is granted. The repair time shown in the figure, As the TTL time value increases, the coverage area of RTQ multicast increases and the time of affected grant time increases.

In Fig. 6 show the average repair time of the source ACK is higher than FAT and IFA because the RTQ directly to the source will spend more time than others. The performance of FAT is between the source ACK and IFA because it send RTQ backtracking to the former parent spend more time than IFA which immediate repaired by nearer multicast node.

V. Concluding Remarks

In this paper, we have proposed an efficient directly-related based protocol, the immediate family ACK protocol, to support reliable multicast service for mobile ad hoc networks. IFA always performs well in ad hoc networks. The proposed protocol of the immediate family ack, each node maintains an ACK table to store the reachability information to three generations of nodes on the tree, i.e., a GID, a PID, and a CID. When a tree is fragmented due to a departed node, the fragments will be glued back to the tree using the Fast-Upstream recovery algorithm, and a new ACK tree will be formed accordingly. IFA then adopts a localize recovery mechanism to speed up retransmissions and prevents the downstream forwarding packets. It obviously save repair time and reduce bandwidth of packets forwarding. We compare the difference approach like Source ACK and FAT. The results show the advantages of the IFA over other two mechanisms in terms of reliability, scalability, and delivery efficiency by simulation. In this paper, we mainly focused on the protocol description, provided a Fast-Upstream recovery algorithm and demonstrated the performance of the IFA protocol by simulation.

Acknowledgment

This work was supported by the National Science Council of Republic of China under grant NSC 100-2221-E-029 -016.

References

[1] R. Ramanathan and J. Redi "A brief overview of Ad hoc networks challenges and directions," *BBN Technologies, IEEE Communications Magazine*, May 2002, pp. 20-21.

[2] S. Basagni, I. Chlamtac, V. Syrotiuk, and R. Talebi "On-demand location aware multicast (OLAM) for Ad hoc networks," *U of Texas at Dallas*, 2000.

[3] W. Liao and M.-Y. Jiang, "Family ACK tree (FAT): Supporting reliable multicast in ad hoc networks," *IEEE Trans. On Vehicular Technology*, vol. 52, no. 6, Nov. 2003.

[4] B. Whetten, T. Montgomery, and S. Kaplan, "A high performance totally ordered multicast protocol," in *Theory and Practice in Distributed Systems. Berlin, Germany: Springer-Verlag, LNCS* vol. 938, 1994.

[5] J.-M. Chang and N. F. Maxemchuk, "Reliable broadcast protocols," *ACM Trans. Comput. Syst*., vol. 1, no. 3, pp. 151–173, Aug. 1984.

[6] S. Paul *et al*., "Reliable multicast transport protocol (RMTP)," *IEEE J. Select. Areas Commun.*, vol. 15, pp. 407–421, Apr. 1997.

[7] S. K. S. Gupta and P. K. Srimani, "An adaptive protocol for reliable multicast in mobile multi-hop radio networks," in *Proc. IEEE Workshop Mobile Computing Systems and Applications.*

[8] E. M. Royer and C. E. Perkins, "Multicast operation of the ad-hoc on-demand distance vector routing protocol," in *Proc. ACM/IEEE Mobicom* vol. 99, 1999.

[9] B. L. Su and Y. M. Huang, "Adaptive path finding for localized cluster in ad hoc networks," in *Proc. IASTED WNET 2004*, vol. 7, 2004.

[10] L. J. Hai, X. Liu, and Y. D. Xia, "Research on multicast routing protocols for mobile ad-hoc networks," *Computer Networks*, vol. 52, no. 5, pp. 988-997, 10 April 2008.

[11] K. Chen and K. Nahrstedt, "Effective location-guided overlay multicast in mobile ad hoc networks," *International Journal of Pervasive Computing and Communications*, vol. 5, no. 4, pp. 388 - 410, 2009.

[12] J. Luo, D. Ye, X. Liu, and M. Fan, "A survey of multicast routing protocols for mobile Ad-Hoc networks," *Communications Surveys and Tutorials, IEEE*, vol. 11, no. 1, pp. 78-91, 2009.