Family ACK Tree (FAT): A New Reliable Multicast Protocol for Mobile Ad Hoc Networks*

Ming-Yu Jiang¹ and Wanjiun Liao²
Department of Electrical Engineering
National Taiwan University
Taipei, Taiwan

Abstract-This paper proposes a new protocol, called Family ACK Tree (FAT), to support reliable multicast service in ad hoc networks. FAT employs a tree-based recovery mechanism that localizes ACKs and retransmissions to avoid feedback implosion. To cope with node movements, FAT constructs an ACK tree in which each node maintains reachability information to three generations of family in the tree. When a tree is fragmented due to a departed node, the fragments will be glued back to the tree using the underlying multicast routing protocol. FAT then adopts an adaptive mechanism to recover missed packets that have been multicast to the group during fragmentation. We have conducted simulations to evaluate the performance of FAT and to compare FAT with the existing solution. The results show that FAT achieves better performance in providing reliable service in ad hoc networks, in terms of reliability, scalability, and delivery efficiency.

Keywords- reliable multicast, mobile ad hoc network

I. INTRODUCTION

A mobile ad-hoc network is an autonomous system comprised only of mobile nodes, which communicate with one another without the support of any wired infrastructure.

IP multicasting is an efficient group communications mechanism, which provides the same unreliable delivery as unicast IP transmission. Reliable multicast protocols ensure reliable end-to-end delivery of unreliable multicast datagrams for group members. A major challenge to this problem is avoiding feedback implosion. As the number of group recipients grows, the number of feedback messages increases dramatically. This leads to a heavy burden on data sources and causes more severe congestion and packet losses.

Reliable multicast in ad hoc networks adds the dimension of multihop nodal mobility within the scope of reliable multicast. Multicast datagrams are

still unreliably delivered to mobile group members. Reliability is ensured by mobile reliable multicast protocols. Thus, once a mobile node has joined a multicast group, it will not experience packet losses or duplications due to roaming. This allows the node to enjoy the service quality of reliable multicast as if it were a stationary host, because it can receive data streams continuously and reliably even while roaming. The challenge to extend reliable multicast protocols for ad hoc networks is coping with nodes movements even while forwarding packets. Therefore, when a node has departed, some nodes may be disconnected from the multicast tree and may miss some multicast packets even though they will eventually be glued back to the tree.

There have been many solutions proposed to support reliable multicast for stationary hosts [1-2]. However the problem of providing reliable multicast service for ad hoc networks has attracted little attention. [3] investigated the problem of reliable multicast in ad hoc networks. The authors suggested an adaptive scheme to support reliable multicast to a set of predefined group members. This protocol, however, needs each recipient to send feedback (to acknowledge on reception of packets or to request for retransmissions) directly back to the source. Thus, it suffers from the problem of feedback implosion.

In this paper, we propose a tree-based protocol, called Family ACK Tree (FAT), for reliable multicast in ad hoc networks. Unlike [3] which requires each node to acknowledge a reception directly back to the source and thus suffers from ACK implosion, FAT is a tree-based recovery mechanism which localizes retransmissions and thus performs better as compared to the existing solution. Note that FAT is not concerned with the data link layer and multicast routing issues, but with the transport layer issue (i.e., reliable multicasting). It runs on top of a constructed multicast tree using arbitrary multicast routing protocols, such as [4-6].

* This work is supported by the National Science Council, Taiwan, under Grant Number NSC90-2213-E-002-119.

¹ This author is now with Benq Corp. in Taiwan.

² This author is also with the Graduate Institute of Communications Engineering, Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan. Email: wjliao@cc.ee.ntu.edu.tw

Thus, FAT works with any existing ad hoc multicast routing protocols.

The rest of the paper is organized as follows. Section 2 describes the proposed Family ACK Tree (FAT) protocol. Section 3 presents the simulation study conducted to evaluate the performance of FAT. Finally the concluding remarks are included in Section 4.

II. FAMILY ACK TREE (FAT) PROTOCOL

The Family ACK Tree (FAT) is the proposed acknowledgement (ACK) tree for reliable multicast in ad hoc networks. A Family ACK tree is overlaid on the underlying multicast tree, in which a parent node (immediate upstream node) serves as the reliability agent for its children nodes. Each node cumulatively acknowledges the reception of packets or requests for retransmissions to its parent node.

A. Basic mechanism

1) Definition

In FAT, each node maintains an ACK table which contains reachability information to three generations of family in the ACK tree. It includes the node's grandparent, the parent, and the children, identified with the Grandparent ID (GID), Parent ID (PID), and Children ID (CID), respectively. A GID, PID, or CID indicates a multicast group of nodes rooted at the grandparent, the parent, and the node itself, respectively.

1. GID (i.e., the grandparent's family) identifies the group of nodes containing the node's parent, the node's grandparent (the parent's parent), and the parent's siblings of the node. In other words, GID describes the family of the node's grandparent.
2. PID (i.e., the parent's family) is the collection of nodes including the node itself, the node's parent, and the node's siblings. In other words, PID describes the family of the node's parent.
3. CID (the node's family) is the group of nodes with the node itself and the node's children. In other words, CID describes the family of the node itself.

Fig. 1 shows the set of nodes corresponding to the GID, the PID, and the CID maintained by node D. D's GID includes nodes A, B, and C, corresponding to the grandparent, the parent, and the uncle (i.e., parent's sibling), respectively. D's PID includes nodes B and D, which are its parent and itself, respectively. D's CID includes itself, D, and its children, nodes E and F.

Figure 1. An example of a Family ACK Tree

2) Message types

There are two types of control messages to be exchanged among the three generations of a node in Family ACK Trees.

1. Sub-group ID advertisement (denoted as Ad)
It is used to configure the ACK table of each node in a Family ACK Tree. Each mobile node periodically broadcasts beacons to its neighbors. Those nodes on the multicast tree will associate Ad messages with the beacons to be broadcast. Upon receipt of an Ad message from the parent of the multicast tree, a node copies the values of PID and CID in the Ad to be its GID and PID, respectively. The node generates its CID, using whatever algorithm chosen by itself so long as the generated CID is different from its GID or PID, or from its siblings' CID's. Once the CID is determined, the node updates the GID and PID in the received Ad with its PID and CID, respectively, and then rebroadcasts the Ad to its neighbors. Note that for those nodes not involved in the multicast tree just ignore the received Ads.
2. Retransmission request (i.e., RTQ)
It is used to report a loss gap to its reliability agent and to request for a retransmission. When a node has departed, the tree is fragmented into a forest. Each subtree rooted at a child of the departed node will be glued back using a multicast routing protocol. Once a node has reconnected to the tree, and has detected the missing packets which cannot be repaired by the new parent, the node multicasts an RTQ message with a time-to-live (TTL) value to its previous GID (i.e., using the previous GID as the destination address of the request). The TTL value is set to limit the range of message forwarding according to the characteristics of network topology.

B. FAT protocol operation: an overview

1) The construction of a Family ACK Tree

1. Once a multicast delivery tree has been constructed by an ad hoc multicast routing protocol, the sub-group ID advertisement process starts. The tree root (i.e., the source) does not have a GID and a PID. Thus, the source generates its CID and duplicates the value of CID to the PID in an Ad. Upon receiving an Ad, a node records the PID and CID as its GID and PID, respectively, generates its CID to be the PID of its children, and then broadcasts the updated Ad message. All the tree nodes store their GIDs, PIDs, and CIDs in the respective acknowledgement tables, except the root which stores a CID only.
2. All nodes on the multicast tree are involved in the sub-group ID advertisement process. Through this process, all nodes are able to acquire their GIDs, PIDs, and CIDs to configure their ACK tables.

2) The maintenance of a Family ACK Tree

1. Upon receipt of a multicast packet, if the packet is received reliably, a node caches it with a predefined timer T_c ; otherwise, the node negatively³ acknowledges the packet for a retransmission upstream to its parent on the ACK tree and starts a predefined timer T_r .
 - (a) If the node has not received a retransmission request for the packet from any of its children on expiry of timer T_c , the node removes the packet from the buffer; otherwise, the node retransmits the packet to those children from whom a NAK is received.
 - (b) If the node has not received a retransmission from its parent on expiry of timer T_r , an RTQ is resent.

This approach allows reliability agents in the upper hierarchical levels in a Family ACK tree to have smaller buffers, as compared to the approach which acknowledges on a reception only after feedbacks from all the children have been received. In the latter approach, there is increasing buffer size in the reliability agents as we move upstream toward the root.

³ To further improve reliability, FAT can also be modified to positively acknowledge correct reception, so that a packet in the buffer is removed only when a node has received feedbacks from all its children before the timer expires. Here, we just demonstrate how FAT works with NAKs.

3. When a node departs, the nodes residing within the transmission range of the departing node will be informed of the node's departure by the data link layer protocol. The nodes notified may include the parent node, the children nodes, and possibly other on-tree sibling nodes. The departing node forwards the packets in its cache upstream to its parent. This rollbacks the parent's buffer and allows the grandparent (i.e., the parent node of the departed node) to serve as the foster parent for its grandchildren (i.e., the children of the departed node).
4. Once a node has departed, all the children of the departed node become orphans. Each orphan node along with its descendants (i.e., a sub-tree rooted at an orphan node) attempts to glue back to the tree independently using the underlying multicast routing protocol. During the "glue back" process, an orphan node sends a request message downstream to prevent its descendant nodes from leaving the sub-tree (e.g., trying to reconnect to the multicast tree by themselves or leaving the group reluctantly). Once an orphan node has reconnected to the tree, it reconfigures its ACK table and identifies the missing packets in which the new parent node is unable to repair. The node then negatively acknowledges the loss gaps to its previous GID and requests for retransmissions.
5. Each node may be in one of two states: normal state and repairing state.
 - (a) Usually, a node is in the normal state.
 - (b) Upon detecting a departure of the parent, a node enters the repairing state.
 - (c) When a node is in the repairing state, it may maintain two ACK tables: the previous ACK table and the current ACK table. Once the node has reliably received all the retransmissions to repair the missing packets, it enters the normal state, when only the current ACK table is maintained.
 - (d) When a node in the normal state receives an RTQ whose GID matches its PID, the request is discarded.
 - (e) When a node in the repairing state receives an RTQ whose GID matches its PID, the node encapsulates the RTQ and forwards the encapsulated RTQ to its previous GID.
 - (f) When a node, say F, receives an RTQ whose GID matches its CID, node F must be the grandparent of the requesting node. If the received request is encapsulated, node F decapsulates the request to identify the original requesting node, say node O; otherwise, the request sender must be the

original requesting node. Node F then retransmits the requested packets directly to node O.

III. PERFORMANCE EVALUATION

This section describes the simulations we have conducted to evaluate the performance of FAT. We investigated the behavior of FAT and compared FAT with a protocol in which feedbacks are directly sent back to the source (denoted as source_ACK in the rest of the paper) as in [3].

A. Simulation setup

This section describes the simulation setup for our experiments. To evaluate the performance for the protocols providing reliable multicast service in ad hoc networks, we assumed that packet transmission was error-free and all the losses were caused by node movements. We considered a 100-by-100 mesh in which nodes were roaming across the mesh during the simulations. The distance between two adjacent intersection points (i.e., coordinates) on the mesh is 20 meters. We randomly determined a set of N mobile nodes, initially located at N randomly selected coordinates in the mesh. We assumed that there was only one sender, randomly selected from N mobiles, and the rest of the $N-1$ mobiles constituted the group members. Each node stayed in a location with an exponentially distributed sojourn time with a mean of M . On expiry of a sojourn time, a node moved to another randomly selected coordinate in the mesh in 2 seconds. Each node had a transmission range of 20 meters, within which mobile nodes could directly communicate with one another. A multicast delivery tree was rooted at a node (i.e., the source sender) randomly selected from N mobile nodes, and spanned over all other nodes. The sender generated data packets at a constant rate of one packet per second.

B. Simulation results

1) Source_ACK vs. FAT

This experiment was conducted to compare FAT with the source_ACK approach. We defined a reliability index as the ratio of the number of the granted requests to the total number of the received requests. A request is granted if the requested packet losses can be retransmitted successfully. The higher the reliability index, the larger the number of lost packets recovered, and the better the performance. We also defined “mobility” as the inverse of the mean time that a node stays at a location. The shorter a node stays at a location, the more frequently a node

moves, and the higher the mobility.

Figure 2. Reliability index vs. mobility

In the source_ACK approach, each recipient negatively acknowledges packet losses directly back to the source. Any lost packet can be recovered by the source with retransmissions. Thus, as shown in Fig. 2, the source_ACK approach has a reliability index of 1, irrespective of host mobility. However, this approach does not scale well, because it suffers from the problem of ACK implosion. Fig. 3 shows that the total number of feedbacks increases dramatically in the source_ACK approach as the number of mobile nodes becomes large (see the left scale in the y axis). This is because with Source_ACK, all these requests go directly to the source, causing the source to become a bottleneck and introducing a large overhead to the system. FAT, on the other hand, is a tree-based local recovery mechanism. Each parent node serves as the reliability agent for its children nodes, thus distributing the overhead of retransmissions to each parent node. This renders lower overhead as shown in Fig. 3 (see the right scale in the y axis). In addition, FAT employs the Family ACK Tree mechanism to cope with node movements, providing a reliable multicast service in ad hoc networks almost as good as in source_ACK (as shown in Fig. 2).

Figure 3. Overheads of both protocols

2) The Family ACK Tree mechanism

1. Bandwidth consumption (scalability)

This simulation measured the bandwidth consumption by FAT. We defined a bandwidth consumption index as the total number of retransmission requests multiplied by the average path length (the number of hops) traversed by each retransmitted packet, normalized by the total number of mobile nodes. Fig. 4 shows that as the number of nodes increases, the bandwidth consumption index decreases, thanks to retransmissions being localized by each parent node. Thus, FAT is scalable, providing reliable multicast services even in a large ad hoc network.

Figure 4. Bandwidth consumption

2. Normalized reliability index vs. buffer size

Fig. 5 shows the three curves of the normalized reliability index, with buffer size varying from 5 s to 10 s. Normalized reliability index is defined as the ratio of the reliability index with a certain buffer size to the reliability index with an infinite buffer.

Figure 5. Normalized reliability index vs. buffer size

The three curves in the figure correspond to the sojourn times with means of 10 s, 20 s, and 80 s, respectively. The longer a node stays at a location, the better reliability FAT achieves. It can be observed that even with a small buffer size, say 5 s, with a mean sojourn time of 20 s, the normalized reliability index reaches above 0.9. As buffer size increases, the normalized reliability index is further

improved. For the all three mean sojourn times, FAT with a buffer size of 8 s is sufficient to reach a normalized reliability index of 0.999, achieving the performance of an infinite buffer.

IV. CONCLUDING REMARKS

In this paper, we have proposed “FAT” to support reliable multicast service in ad hoc networks. FAT employs a tree-based recovery mechanism that localizes ACKs and retransmissions. To cope with node movements, FAT constructs an ACK tree in which each node maintains reachability information to three generations of family in the tree. When a tree is fragmented due to a departed node, the fragments will be glued back to the tree using the underlying multicast routing protocol. FAT then adopts an adaptive mechanism to recover missed packets that have been multicast to the group during fragmentation. We have conducted simulations to evaluate the performance of FAT. The results show that FAT achieves the best performance in providing reliable service in ad hoc networks, in terms of reliability, scalability, and delivery efficiency.

REFERENCES

- [1] S. Paul et al, “Reliable Multicast Transport Protocol (RMTP),” *IEEE Journal on Selected Areas in Communications*, Vol. 15, No. 3, pp. 407-421, April 1997.
- [2] Luigi Rizzo and Lorenzo Vicisano, “A Reliable Multicast data Distribution Protocol Based on Software FEC techniques (RMDP),” *Proc. IEEE HPCS '97*, pp. 115-124, 1997.
- [3] Sandeep K. S. Gupta and Pradip K. Srimani. “An Adaptive Protocol for Reliable Multicast in Mobile Multi-hop Radio Networks,” *IEEE Workshop on Mobile Computing Systems and Applications*, 1999.
- [4] Elizabeth M. Royer and Charles E. Perkins. “Multicast Operation of the Ad-hoc On-Demand Distance Vector Routing Protocol,” *Proc. ACM/IEEE MOBICom'99*, 1999.
- [5] S.-J. Lee, M. Gerla, and C.-C. Chiang, “On-Demand Multicast Routing Protocol,” *Proc. IEEE WCNC '99*, 1999.
- [6] J. J. Garcia-Luna-Aceves and E. L. Madruga, “A Multicast Routing Protocol for Ad-Hoc Networks,” *Proc. IEEE INFOCOM '99*, New York, 1999.