Reliable Multicast Using Shared Ack Trees

J.J. Garcia-Luna-Aceves **Brian Neil Levine**

Computer Engineering Department

University of California, Santa Cruz

Overview

- Definition of reliable multicast service.
- Taxonomy of reliable multicast protocols.
- Maximum attainable throughput analysis.
- Description of Lorax, a shared ack tree protocol.
- Conclusions.

A Reliable Multicast Service Definition

- ***Every packet is delivered*** **to each receiver within a finite time and free of errors.**
- **Additionally, only a single copy of each packet is delivered, in the order sent by the source.**
- **Efficient garbage collection. Packets can be *safely deleted* within a finite amount of time.**

Taxonomy

Ways to solve the ACK implosion problem.

- **Whether ACKs or NAKs are used is not important, but rather:**
  - **Who controls memory allocation and pacing?**
  - **How are receivers organized?**
- ***Structures imposed on the members of the session.***
  - **None, rings, or trees.**

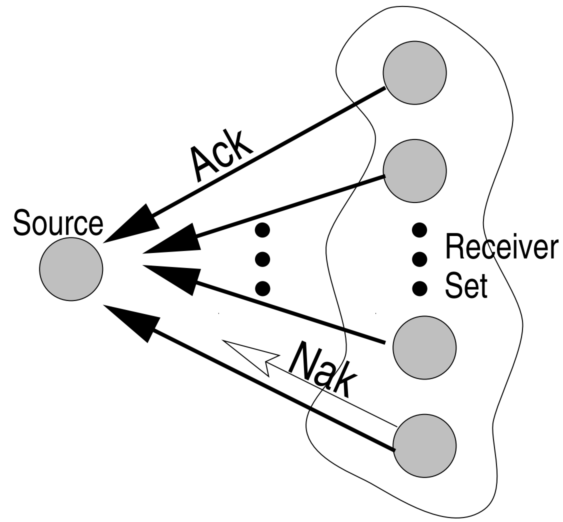
Taxonomy (cont'd)

- ***Congestion Window.***
  - **Packets possibly requiring retransmission.**
- ***Memory Window.***
  - **Packets requiring acknowledgment from all receivers.**

Reliable Protocol Classes

1. **Sender-Initiated**

- **No structure,**
- **deletion with *ACKs*, repairs with *ACKs* or *NAKs*.**



2. **Receiver-Initiated**

- **No structure,**
- ***no deletion*,**
- **repairs with *NAKs*.**

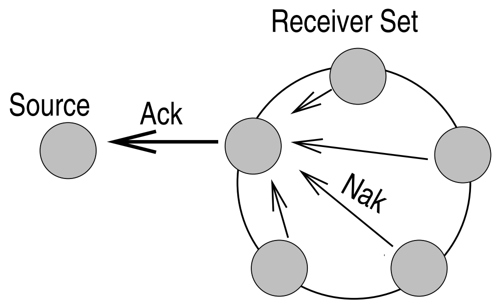
Reliable Protocol Classes (cont'd)

2 Subclass.

Receiver-initiated with NAK-avoidance (RINA)

- **Nodes wait and listen before sending NAKs.**

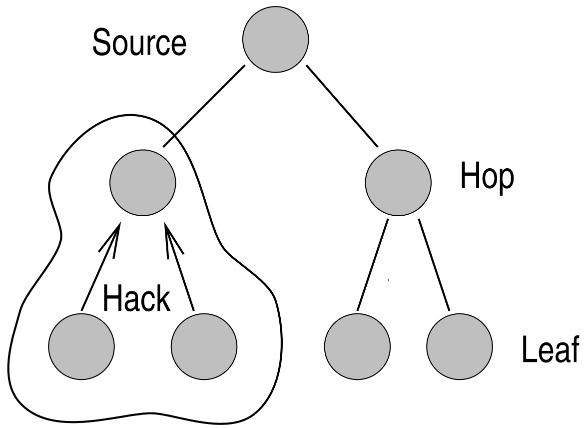
3. **Ring-based**



- ***logical ring structure,***
- **rotating *token* allows deletion,**
- ***NAKs* sent to token site to request repairs.**

Reliable Protocol Classes (cont'd)

4. Tree-based



- ***ack tree structure***,
- deletion with ***Aggregate ACKs***,
- repairs with local ***ACKs***.

4 Subclass.

Tree-based with local NAK-avoidance and Period Polling (Tree-NAPP)

- NAK-avoidance within local group.

Method of Analysis

- **Based on work by Pingali, Towsley, and Kurose.**
- **We are interested in the *maximum throughput* of each class.**
- **All loss events at any node are mutually independent.**
- **No acknowledgments are lost.**
  - **Protocols that rely on NAK-avoidance are helped the most. Tree-NAPP is not helped as much as RINA.**

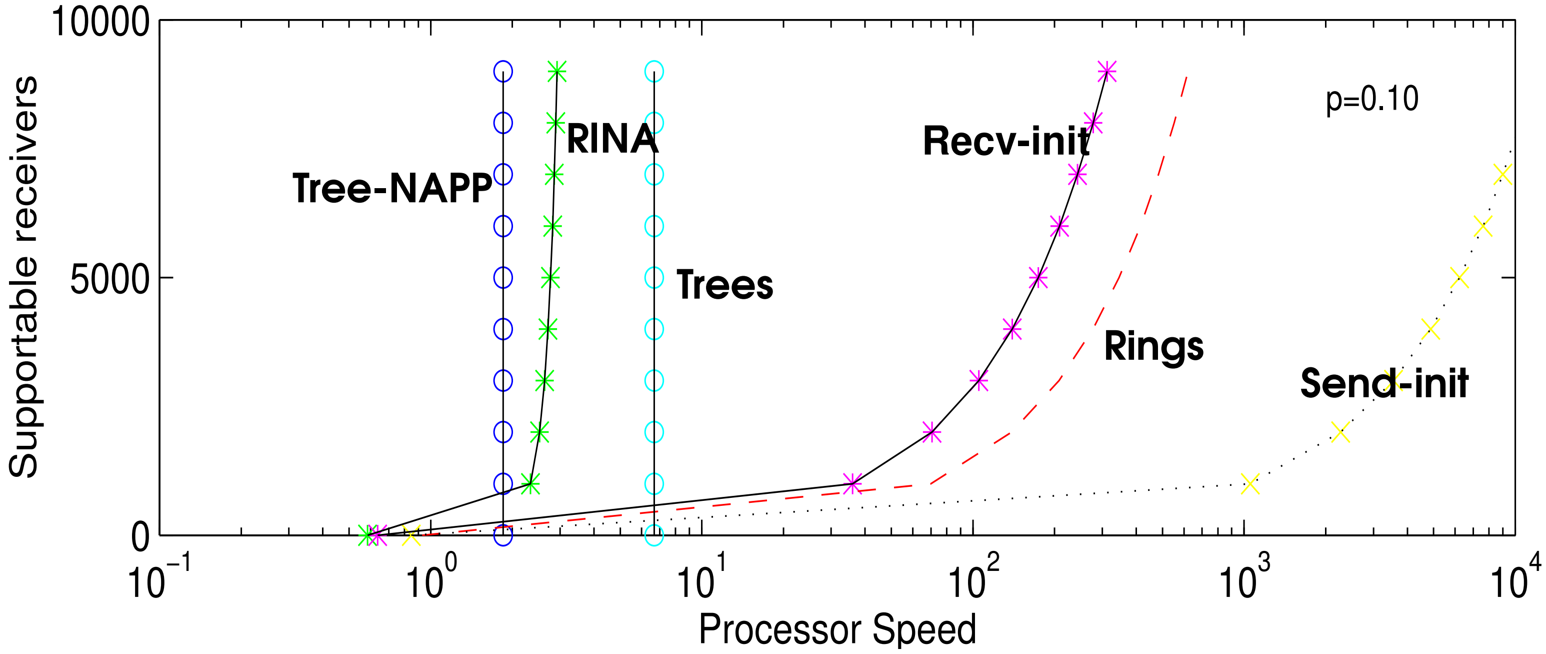
Analytical Bounds

Protocol	Processing, func. of recv. set
Sender-Initiated*	$O(R \ln R)$
Receiver-Initiated*	$O(R)$
RINA*	$O(\ln R)$
Ring-based‡	$O(R)$
Tree-based‡	$O(1)$
Tree-NAPP‡	$O(1)$

❋ *Results from Pingali, Towsley, and Kurose.*

‡ *Results from Levine and Garcia-Luna.*

Processor Speed vs. Number of Supportable Receivers



RINA Protocols

e.g., Log Based Receiver-Reliable Multicast or SRM

- **A server logs and stores the entire session indefinitely.**
- **All receivers store the entire session indefinitely.**
- **Soft-state NAK-avoidance timers is an $O(n^2)$ operation that does not scale.**
- **The limitations of sender-initiated is the need to process ACKs and to know *all* the receivers.**
- **Using the TTL field is a poor approximation to local groups.**
- **NAK-avoidance not as useful when NAKs are lost.**

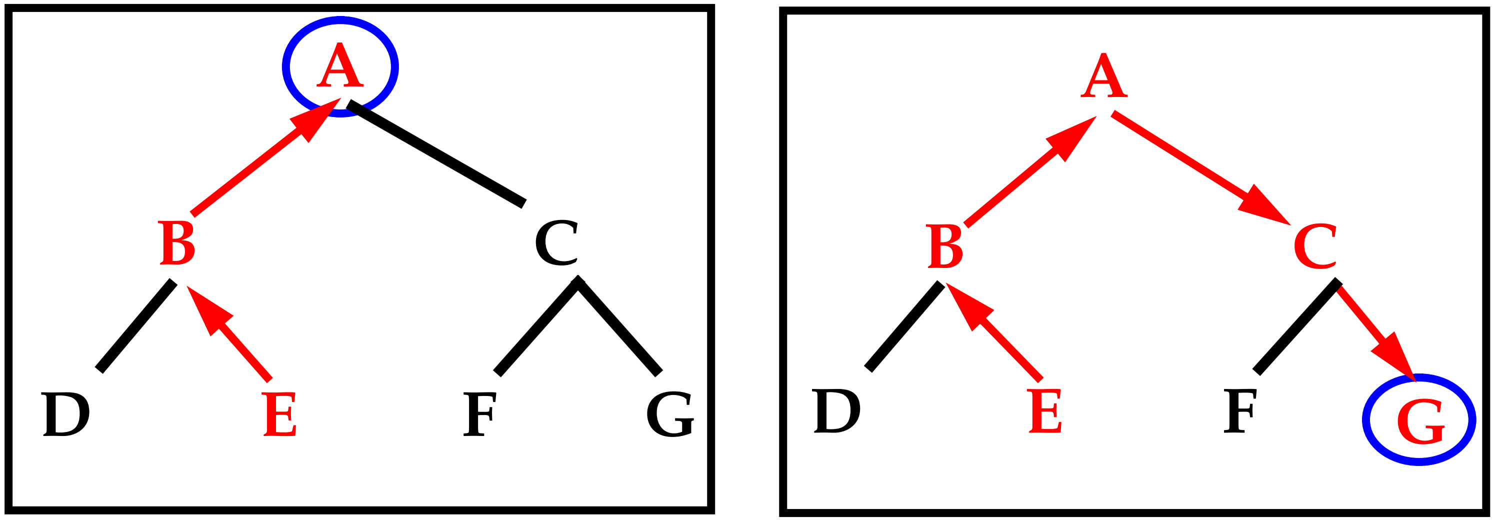
The Lorax Protocol

- The first Reliable Concurrent Multicast protocol that constructs and maintains a ***shared ack tree*** for all sources in a session.
- Lorax ***self-routes*** acknowledgments towards sources. There are ***no routing tables***.
- Lorax is the first RINA or Tree-based protocol proposed to date that ***has safe deletion***.
- The deletion mechanisms used in Lorax can be combined with any retransmission strategy.

Shared Ack Trees

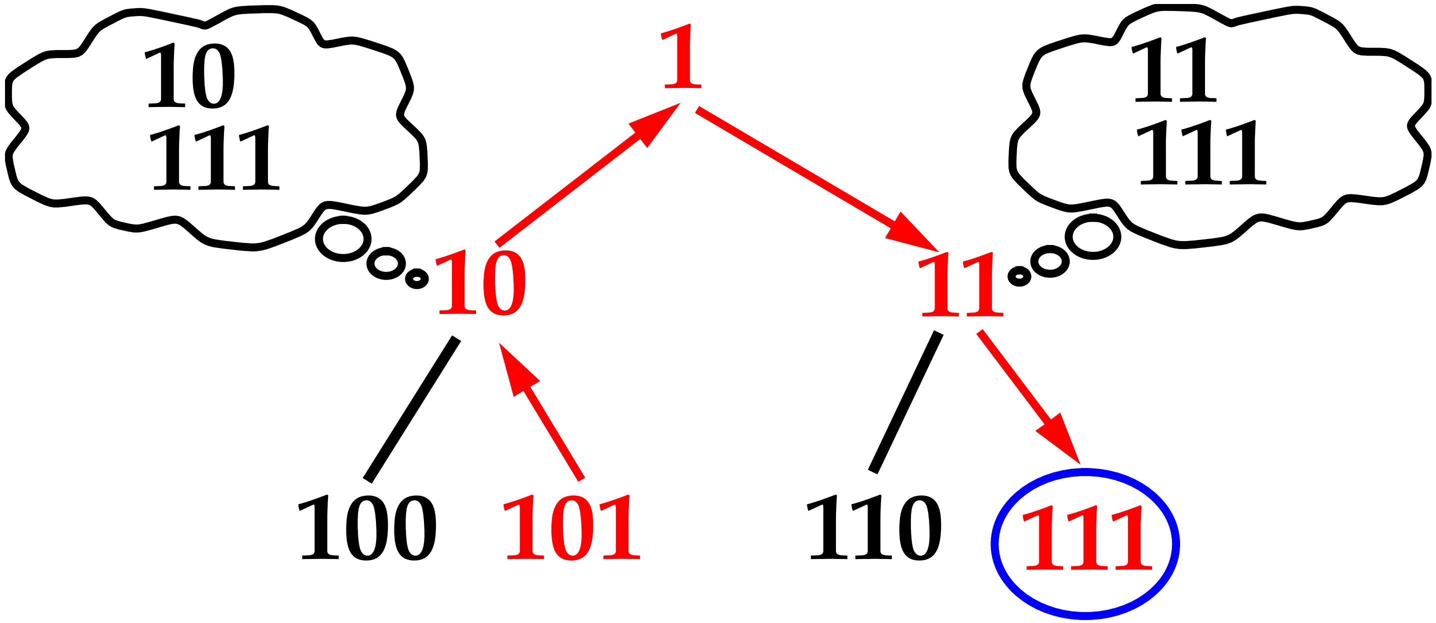
- Current tree-based protocols do not scale, because they need one ack tree per source per group.

Shared Ack Trees must be able to route ACKs.



The path lies through parent or through a child.

Implicit Labels and Routing



No routing tables are needed.

Lorax Is Two Protocols in One

The tree built by Lorax can be used to handle Retransmissions, or Deletion, or both.

Retransmissions	***Deletion***
Sender-Initiated	**No Structure**
Receiver-Initiated	**No Deletion**
Rings	**Rings**
Trees	**Trees**

Summary

- Our maximum throughput analysis suggests that trees are the best method.
- Lorax is the first protocol to construct and maintain a shared ack tree for multiple sources.
- Routing of acknowledgments is performed without routing tables at each host.
- Lorax can be setup as an extension to protocols that cannot safely delete packets.

Postscript of papers available at:

http://www.cse.ucsc.edu/research/ccrg/publications.html