Wait for ACKs from all receivers, re-send on timeout or selective ACK.

**Sender Reliable Error Detection**

- easy resource management
- wait for ACK

**Receiver Reliable Error Detection**

- does not provide 100% reliability
- select ACK
- wait for ACKs from all receivers, re-send on timeout or ACK implosion

**Implosion**

- does not provide 100% reliability
- Receiver state in sender, not scalable
- ACK implosion
- NACK implosion

**Receiver Reliable Error Detection** (Cont.)

http://merlot.usc.edu/cs551-f12

Bill Cheng

Reliable Multicast

CS551
Implosion (Cont...)
Aside - Using the Routers

- Expose (Cont...)
- Aside - Using the Routers

SRTM Request Suppression

SRM (Scalable Reliable Multicast)

No assistance from routers
Every member may multicast NACK or retransmit
NACK-based
Receiver reliable
Original designed for wb

SRM Retransmission

SRM Request Suppression

http://merlot.usc.edu/cs551-f12

Bill Cheng
[Floyd97c]
Scalable Reliable Multicast
CS551

original designed for wb

not scalable

viable end-to-end argument

the sources of exposure and expose problems but:

send retransmissions

source ACKs

other packets

Routers do transport level processing
When noticing skipped sequence number:
start a timer whose timeout is proportional to distance from Src.
When noticing skipped sequence number:
- start a timer whose timeout is proportional to distance from Src

**SRM Request Suppression (Cont...)**
SRM Summary

Better: only retransmit to those members who lost the packet.

Router support: only retransmit to those members who lost the packet.

Local recovery: discover lost final packets.

Loss at link (A,C): cause retransmission at Snoop or Snoop router.

adapttoalgorithmtoadjustconstants

Where have we seen this before?

 Repair sending

Detector package and stochastic components

Delay based on RTT estimation

used to estimate OTT from sender to receivers

NACK/Retransmission Suppression

Periodic session messages