A Reliable Multicast Framework for Light-weight Sessions and Application Level Framing

Sally Floyd, Van Jacobson, Steve McCanne
Lawrence Berkeley National Laboratory

Ching-Gung Liu
University of Southern California

Lixia Zhang
Xerox PARC

SIGCOMM ’95
Cambridge, MA
1 September 1995

Errata:

- Figure 3 in the proceedings contains the wrong figures.
- Graphs all show Delay/RTT that is 0.5 too big (e.g., 3.0 should be 2.5).

Corrected paper and tech report (longer version) available at:

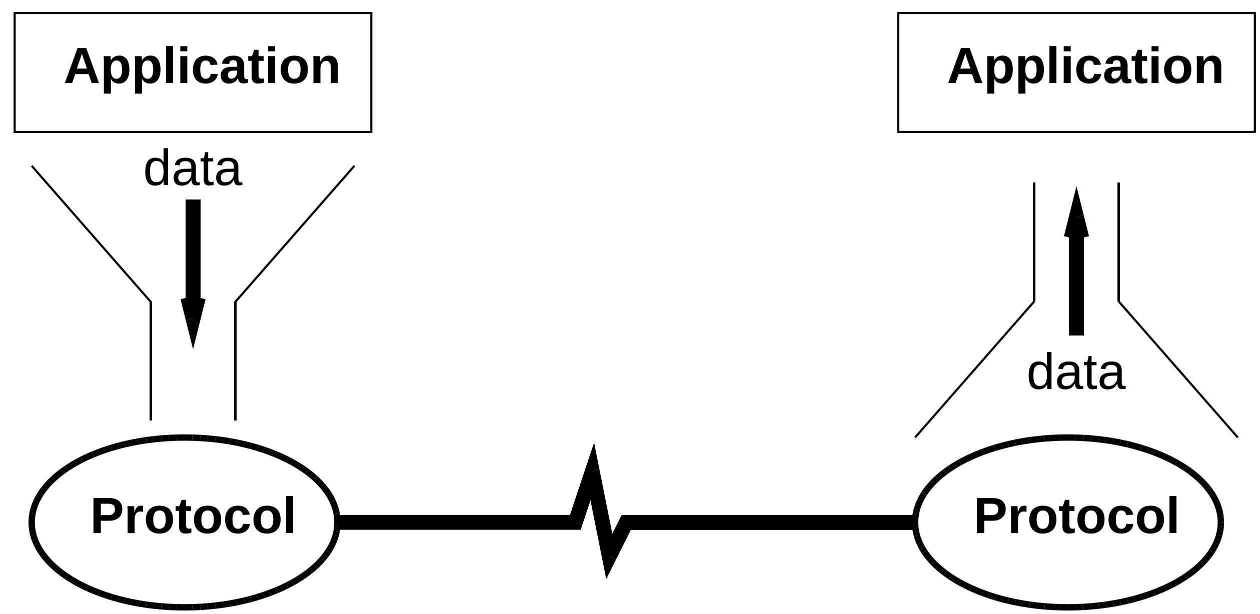
`ftp://ftp.ee.lbl.gov/papers/srm.ps.Z`

`ftp://ftp.ee.lbl.gov/papers/wb.tech.ps.Z`

Why Multicast?

- Efficiency (only one copy of data per link, independent of number of receivers).
- Group queries (can request data without knowing who has it).

The World used to be so simple . . .



. . . **but multicast changes the rules**

- Sender can’t keep ‘state’ for unknown number of receivers.
- Algorithms based on estimating path properties (RTT, congestion window) don’t generalize to trees.
- Model of communication as ‘conversation’ breaks down.

Most work on reliable multicast attempts to condition environment so unicast transport models will work. E.g., Chang & Maxemchuk (and derivatives like RMP) form members into token ring; MTP elects a central controller.

These approaches have serious scaling problems. (Forming ring or electing leader require group-wide agreement which is expensive and problematic when membership changes frequently.)

At SIGCOMM 90, Clark and Tennenhouse proposed a new communication model, Application Level Framing (ALF), that easily generalizes to multicast.

Some key parts are to let applications manage the communication, speak in “application data units” (e.g., video frames, disk blocks) and use an application-specific namespace for data (e.g., filename & sector offset).

Since 1991, we have been trying to elaborate the ALF model.

One piece we’ve developed is a scalable, reliable multicast framework, SRM. It is fully decentralized (no ring or central controller) and handles arbitrarily large groups.

A complete protocol using the framework has been implemented in the LBL whiteboard tool, *wb,* and tested on the MBone. Wb has been in widespread use since 1993 for conferences with anywhere from two to several thousand participants.

SRM Reliability Machinery

- *All* traffic is multicast.

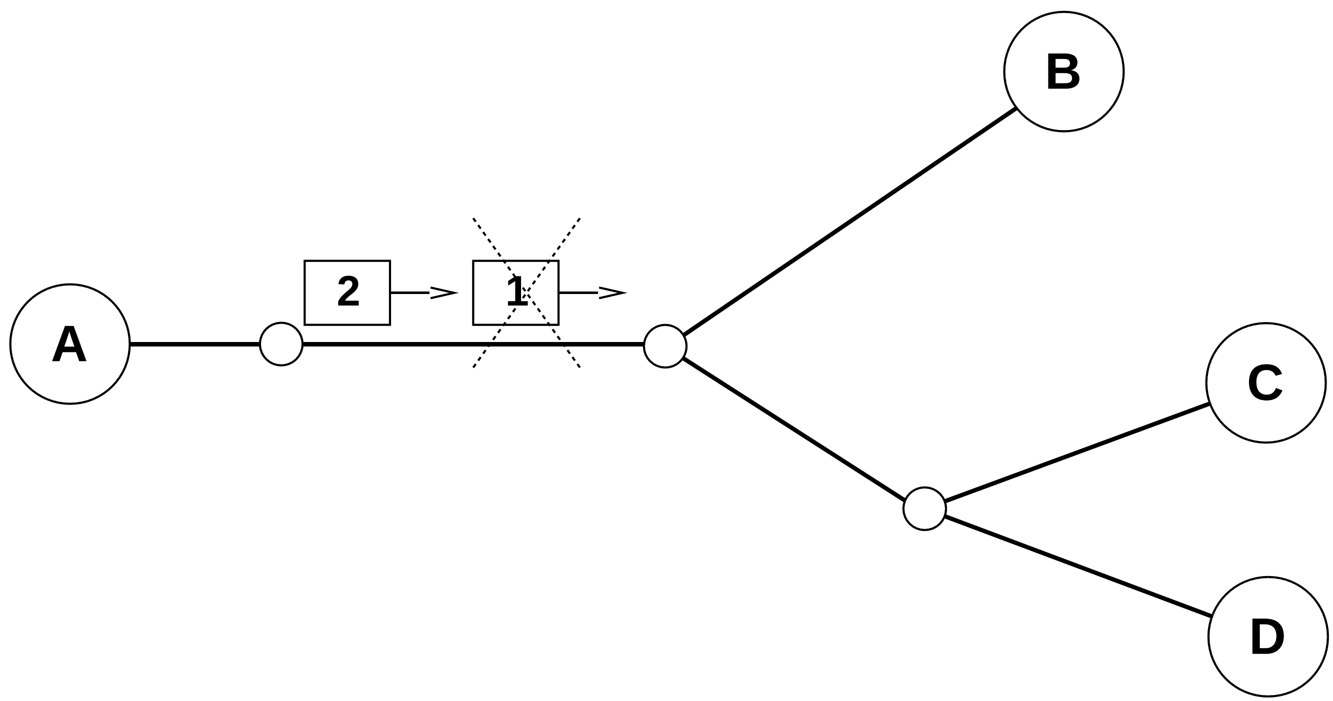
- Each session has a bandwidth limit. Anyone can send if have data and aggregate traffic is under limit.

- All members send low-rate ‘reports’ that contain their current state. Report sends randomized and rate limited to 3% of session bandwidth

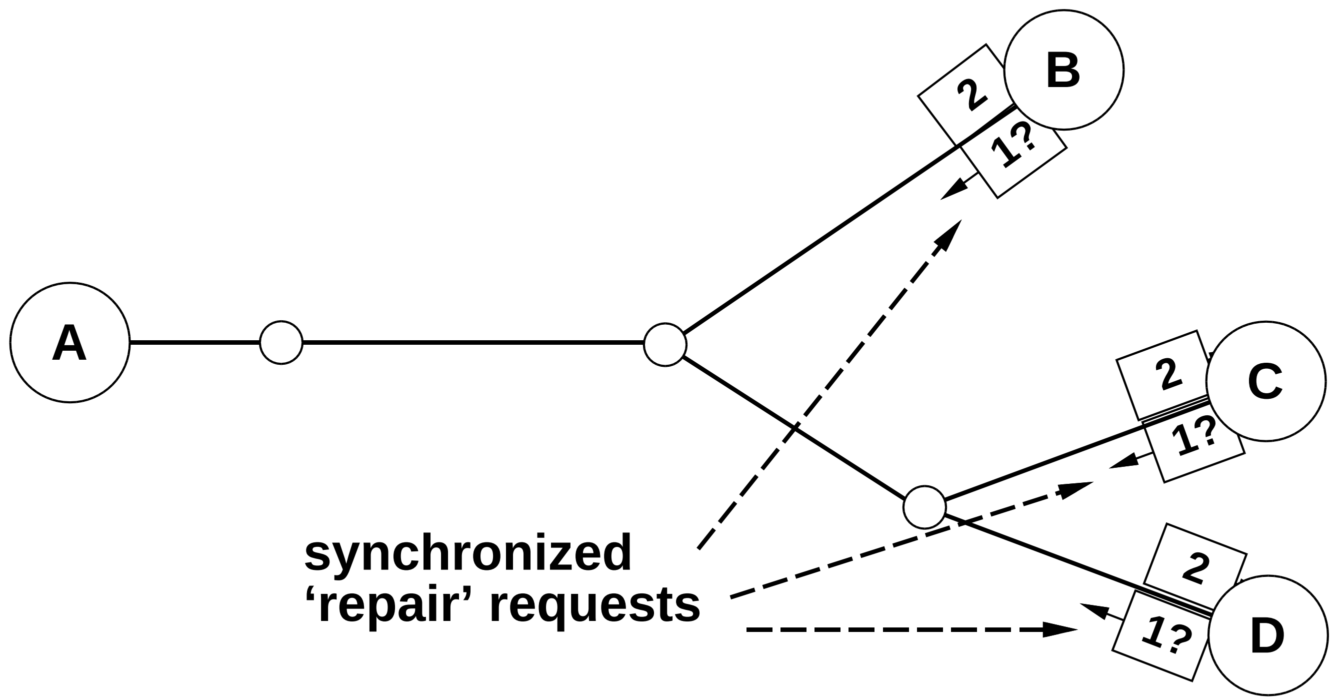
SRM Reliability Machinery (cont.)

- Receivers learn they’re missing data either from hole in sequence space or from someone’s report.
- Receivers multicast a ‘repair request’ to ask for missing data.
- Anyone that has data can reply, not just original source of data.

‘Ack Implosions’



‘Ack Implosions’ (cont.)



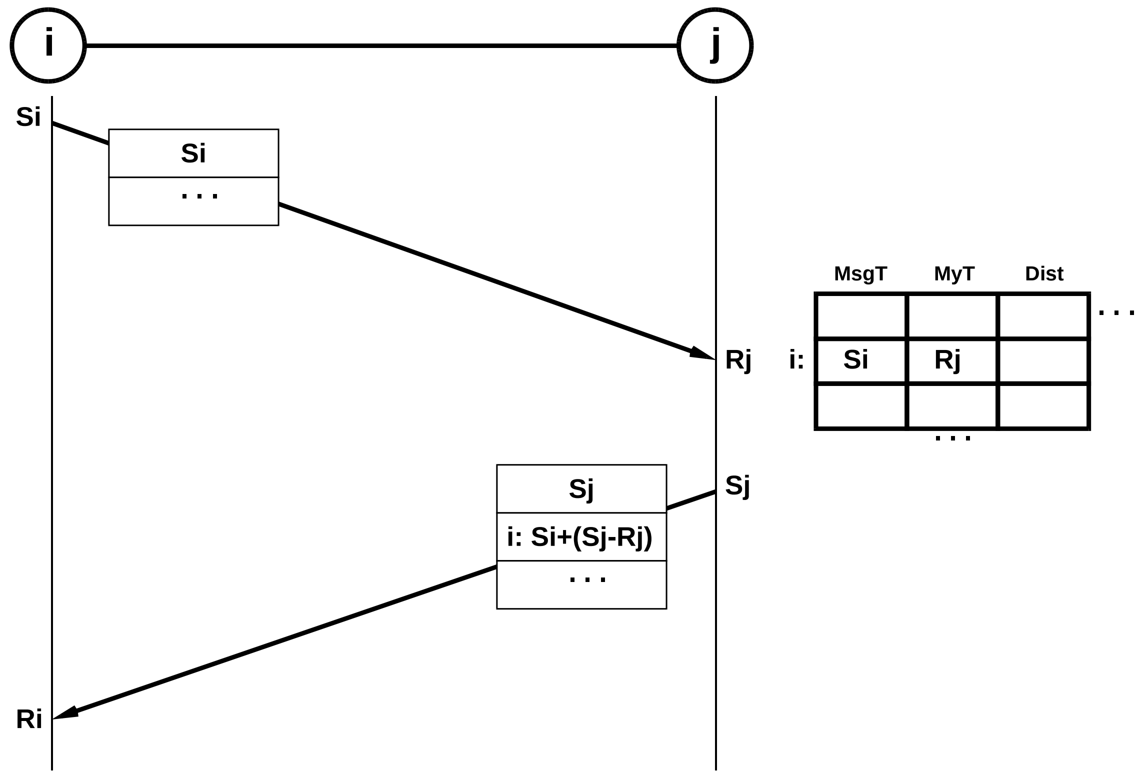
Avoiding ack implosions

- Every node estimates distance (in time) from every other node. (Info for this carried in session reports.)

- Nodes use randomized function of distance to decide when they should request repair (or reply to a repair request).

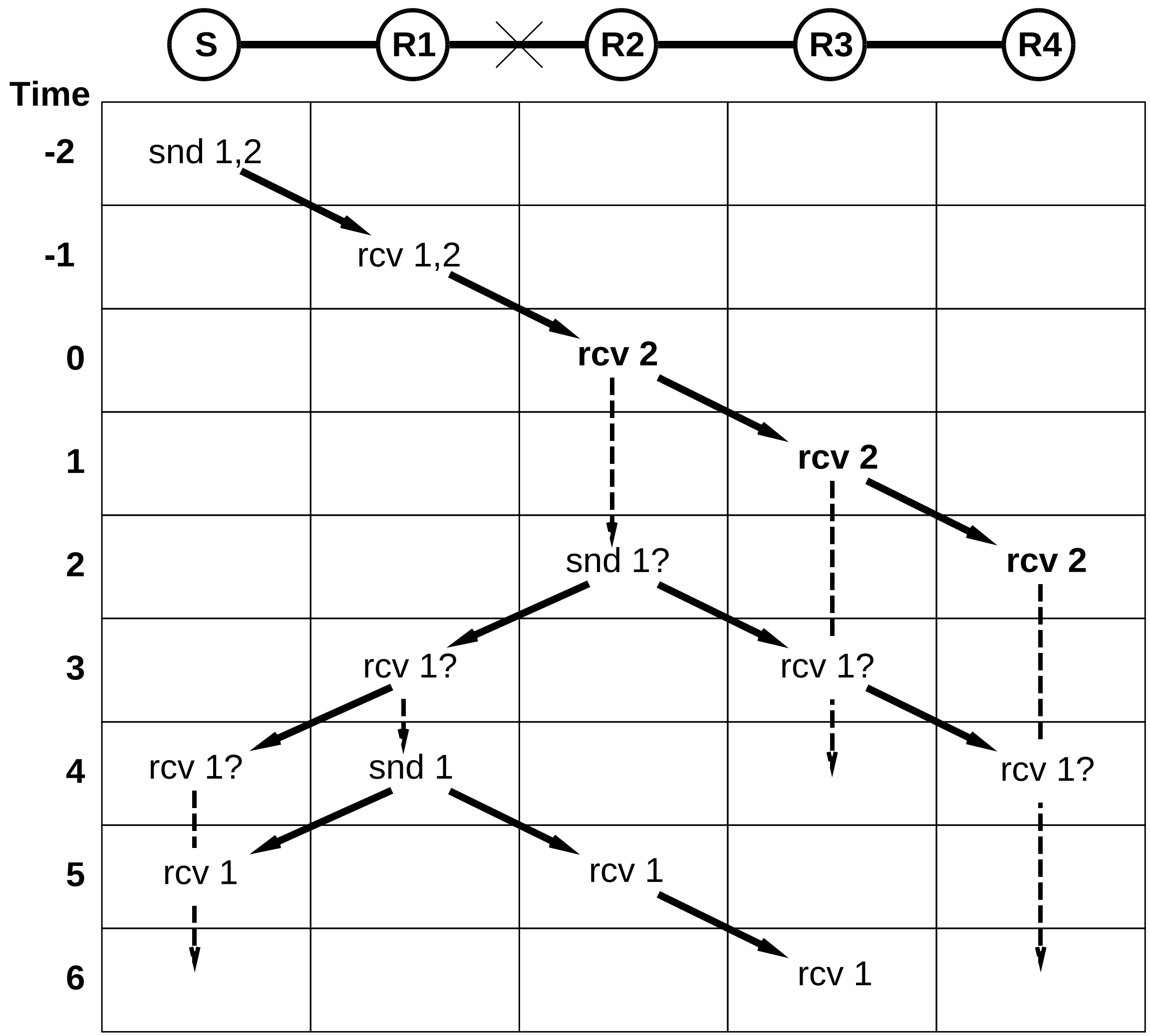
- Receipt of request or reply causes node to suppress its own attempt.

Distance Estimates

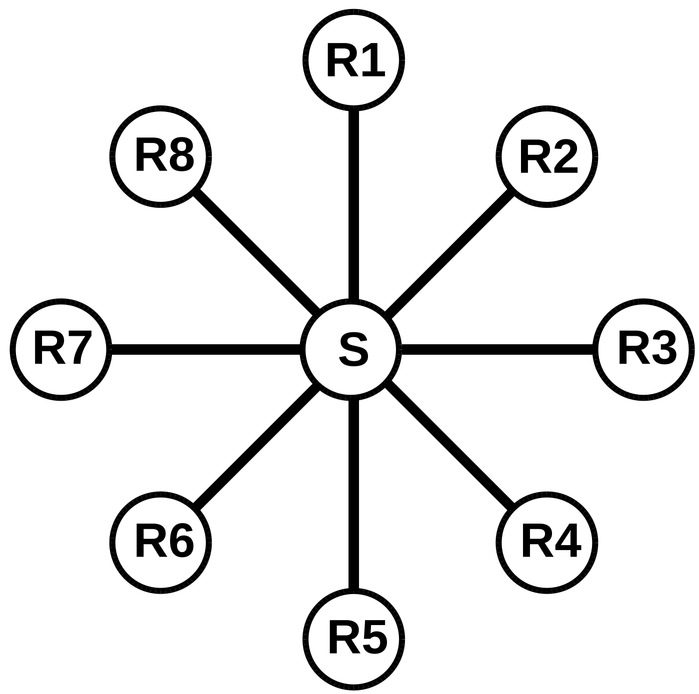


When *j*'s report arrives at *i*, distance from *j* is calculated as: $(R_i - S_i^*)/2$.

Linear Topology Repair Chronology



Worst case topology (star) and randomization



Request and repair timers set to random number in intervals:

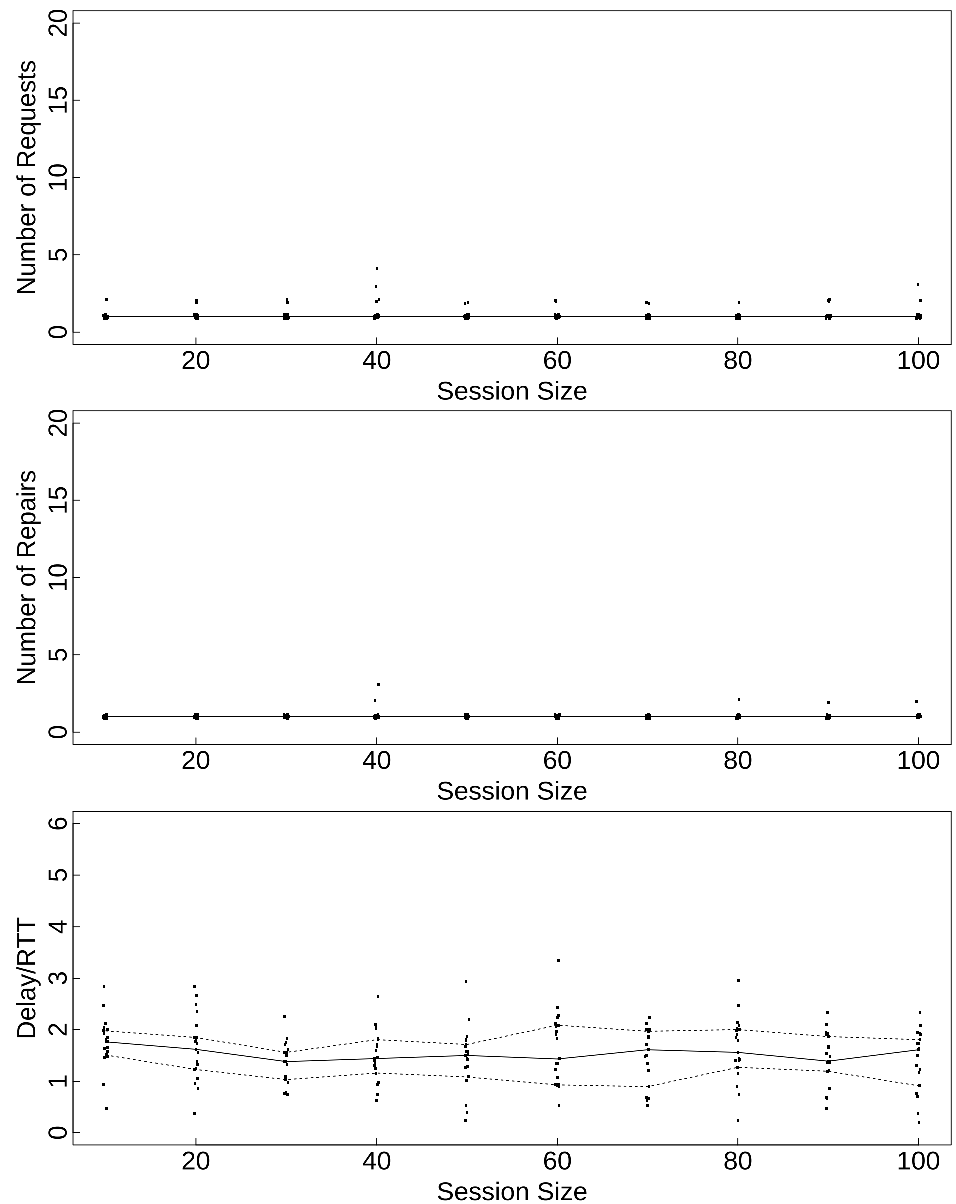
$$[c_1, c_1 + c_2)D_s$$

$$[d_1, d_1 + d_2)D_s$$

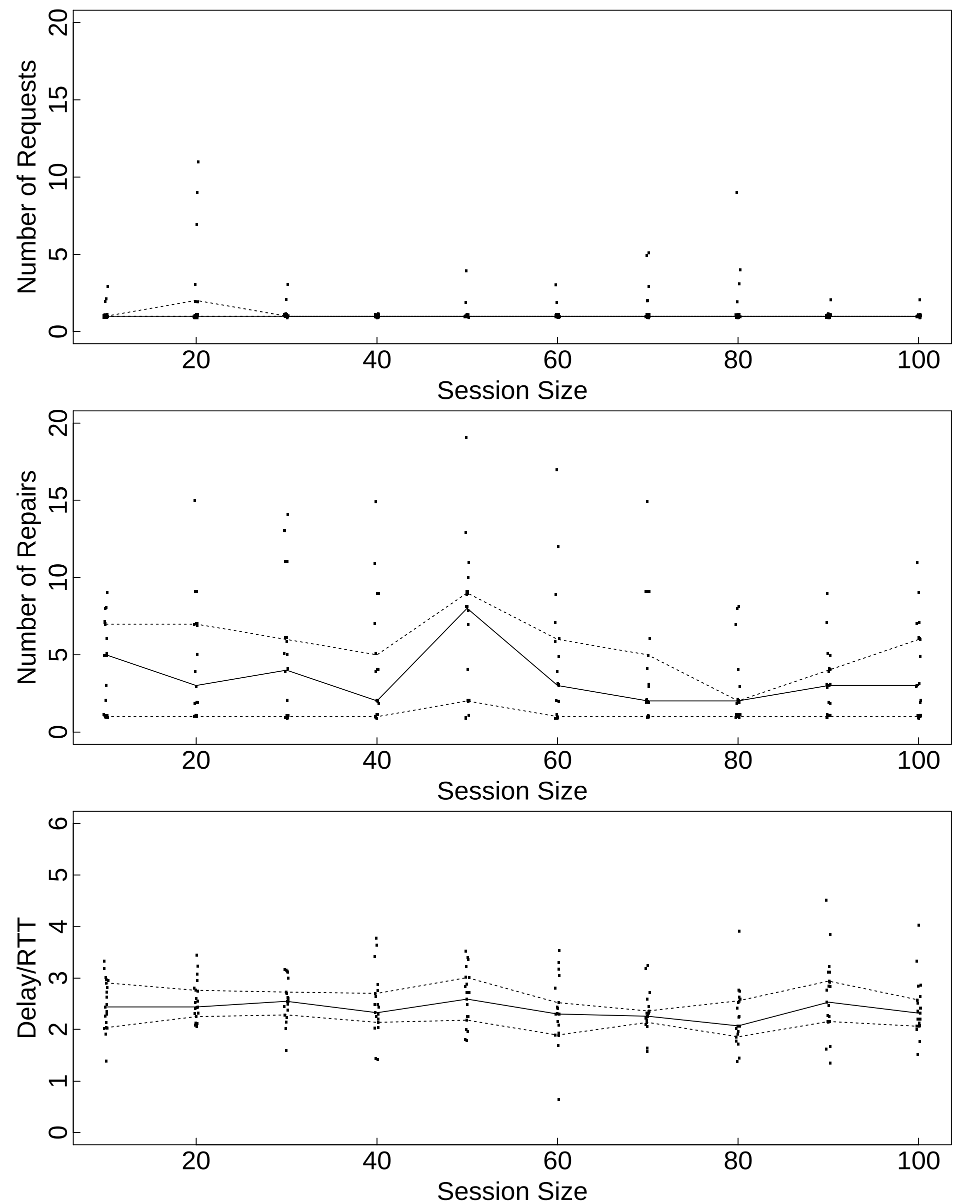
Simplest SRM uses fixed values for constants:

$$c_1 = c_2 = 2$$

$$d_1 = d_2 = \text{log}_{10}(\text{members})$$

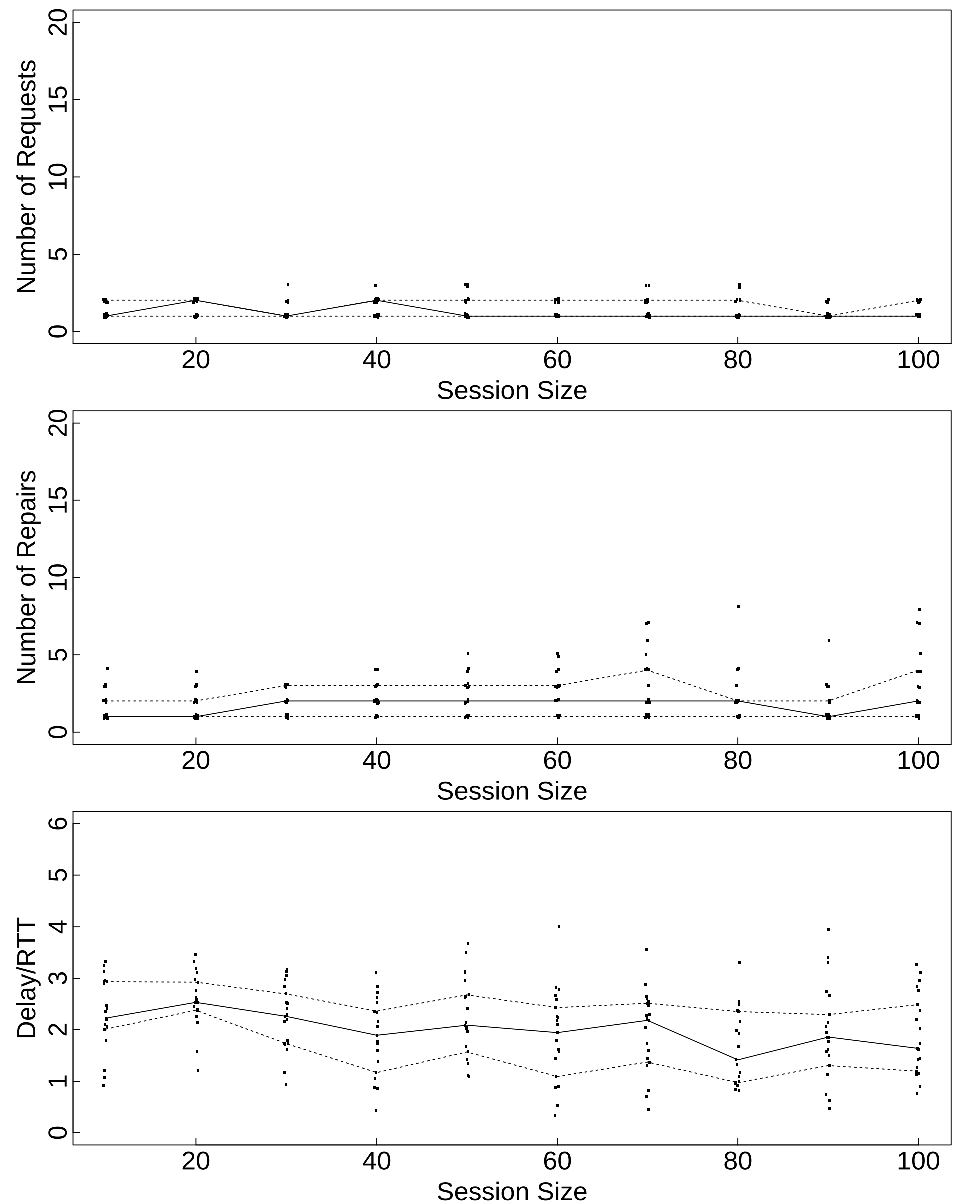


(random trees; all nodes members)



(1000 node, bounded degree trees)

Random interval constants (weakly) sensitive to both topology and location of loss. Can get better repair response, fewer duplicates, or both, if c and d dynamically adjusted:



(1000 node, bounded degree trees, adaptive algorithm)

Other SRM Applications

- Almost any large-scale data distribution — BGP routes, DNS zone xfers, Usenet news, stock quotes, etc.)
- Self-configuring cache hierarchies for, e.g., Web or FTP data.

Some Open Questions

- ‘Local’ repair to avoid ‘crying baby’ problem.
- Other forms of bandwidth adaptation / congestion control.