Search Party:
An Approach to Reliable Multicast
with Local Recovery

(work in progress)

Adam M. Costello

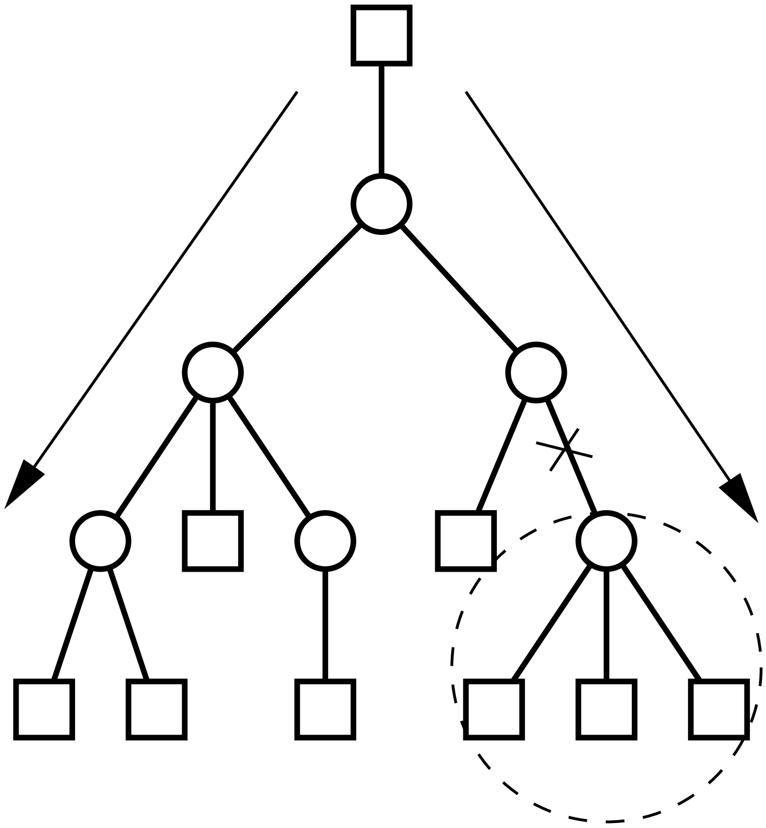
MASH Retreat
Wed 25 Jun 1997

http://www.cs.berkeley.edu/~amc/research/search-party/

Outline

- Background
  - IP multicast
  - reliability challenges
  - existing approaches
- Search Party
  - randomcast
  - reliable multicast
  - virtual randomcast
- Future Work
- Conclusion

IP Multicast

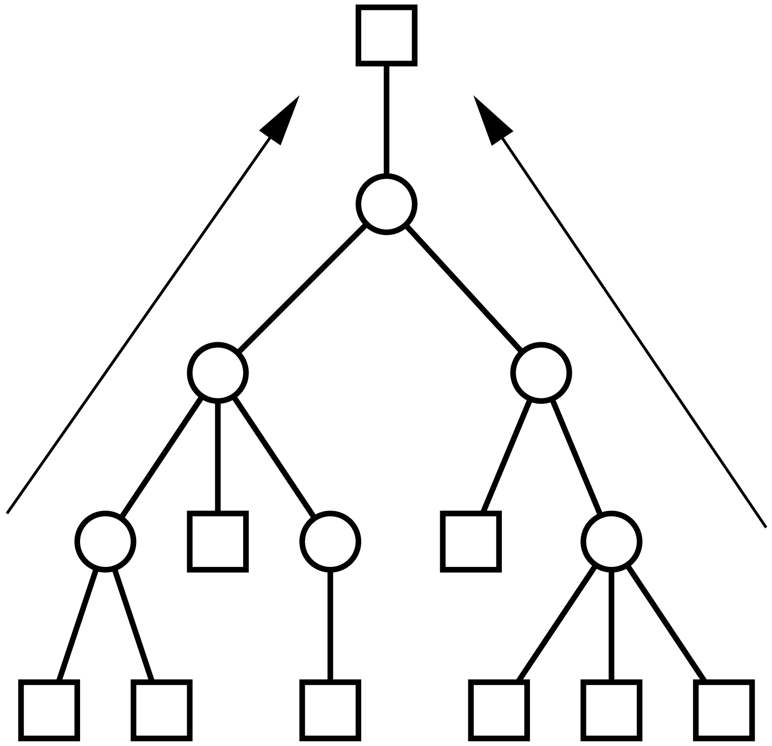


Best-effort is okay for real-time audio/video, but reliability is needed for

- software updates & news articles
- whiteboard & multiplayer games

Reliability Challenges

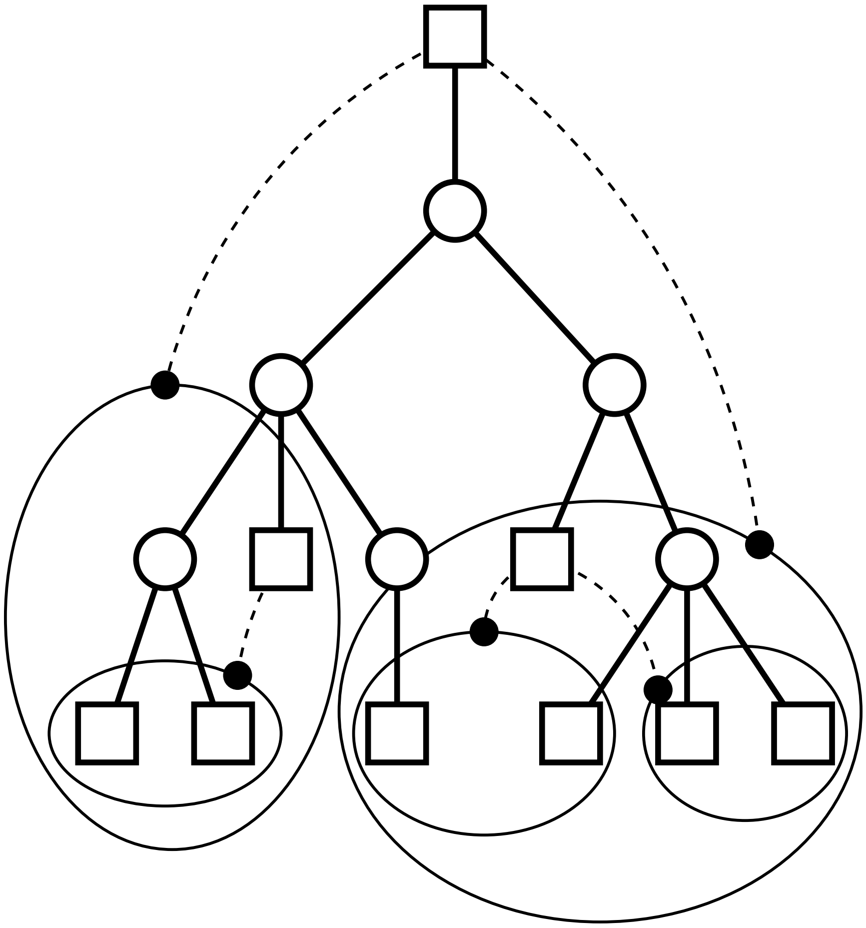
- implosion



- scalable retransmission

Existing Approach: RMTP, TMTP

- *logical* hierarchy of receivers
- merge acks
- local retransmissions
- **drawback:** hard to maintain good hierarchy



Existing Approach: SRM

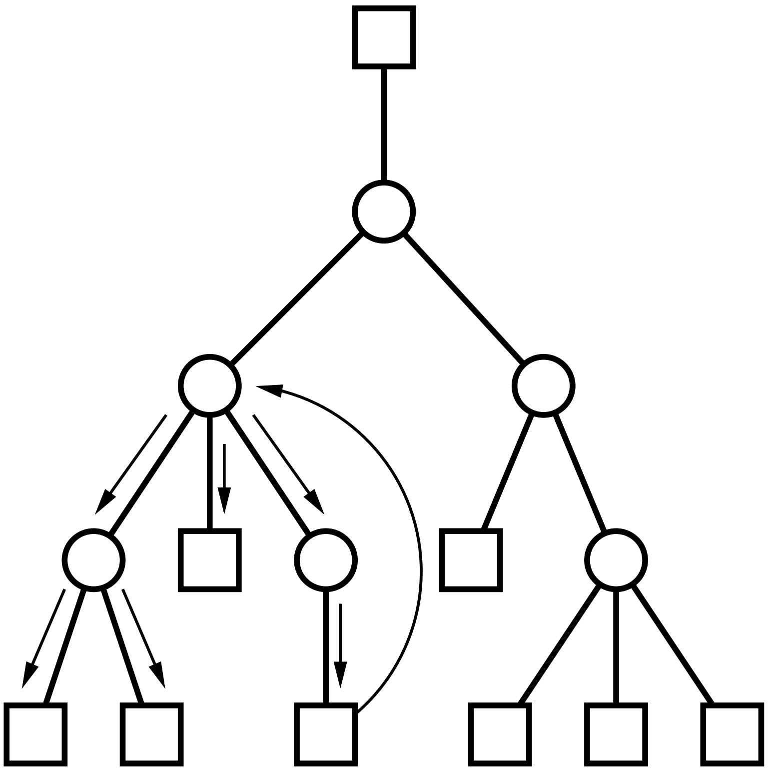
- no hierarchy

- randomly delayed nacks

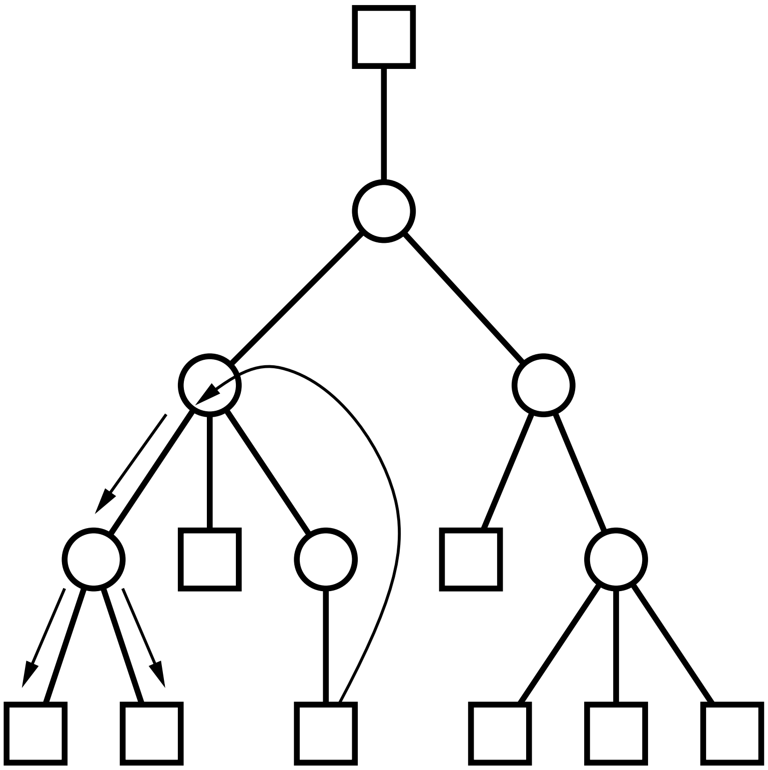
- **drawback:**
  ideas for local retransmission are sketchy

Interjection: New Forwarding Services

Subcast:

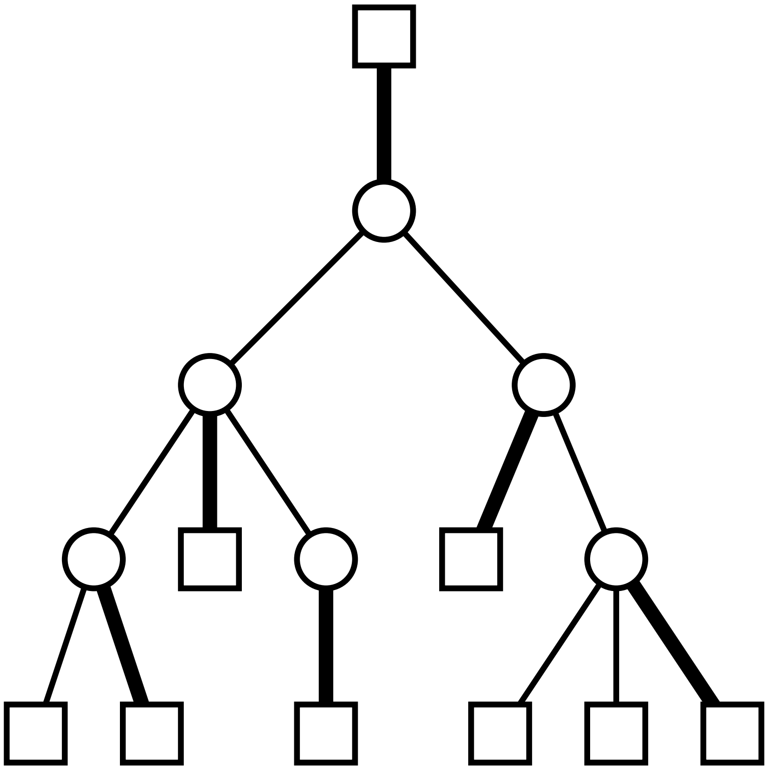


Directed Multicast:



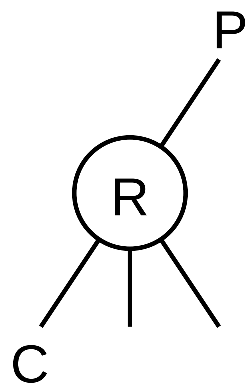
Existing Approach: Replier (Christos)

- uses multicast tree as hierarchy
- new forwarding service for requests
- directed multicast responses
- **drawback:** fragile



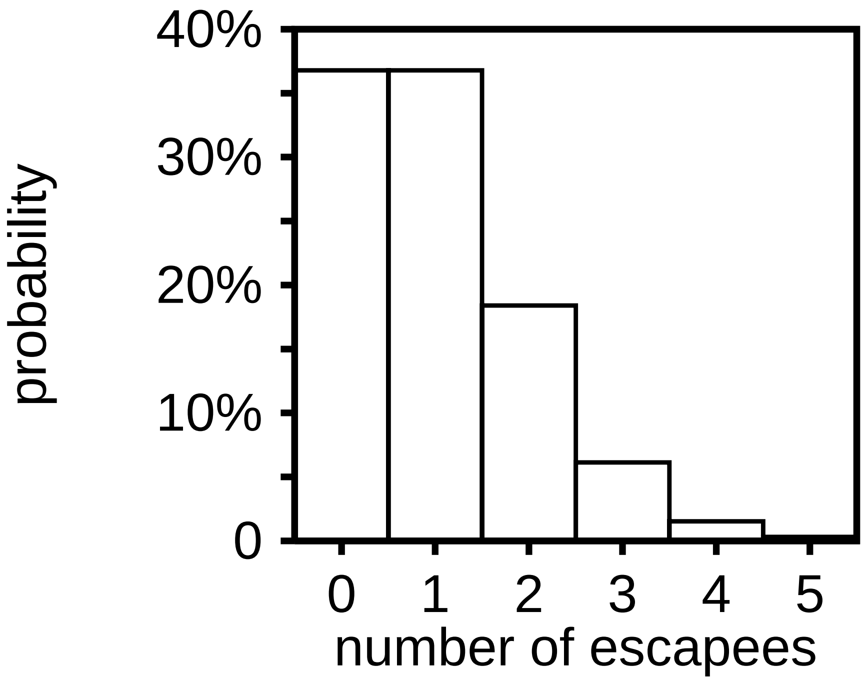
Randomcast

New forwarding service: random tree-walk

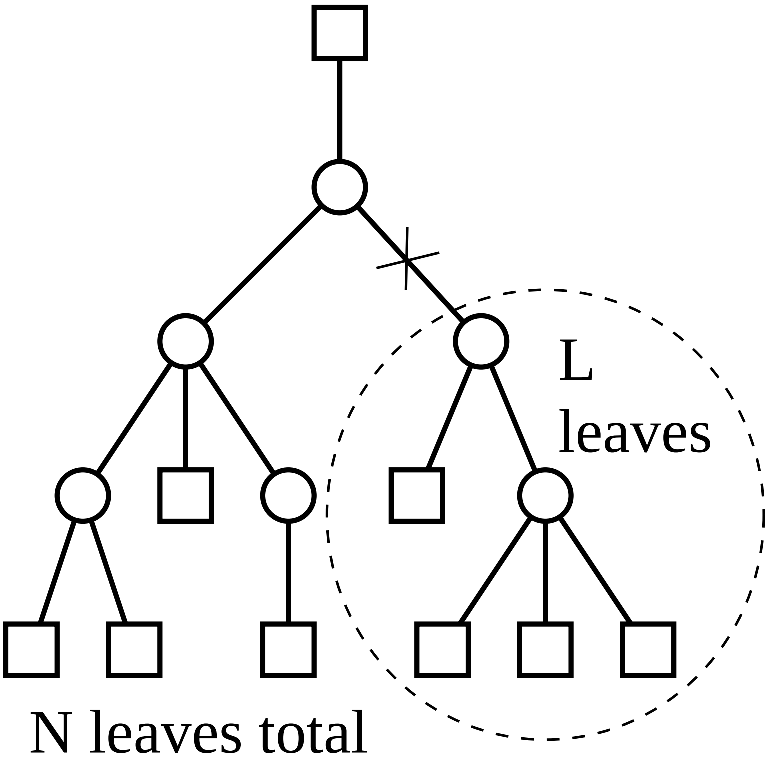


- from C $\Rightarrow$ forward to P w/prob size(C)/size(R) else forward to any other child (uniformly)

- from P $\Rightarrow$ to any child (uniformly)



Search Party Reliable Multicast



- randomcast requests at average rate μ (Poisson)

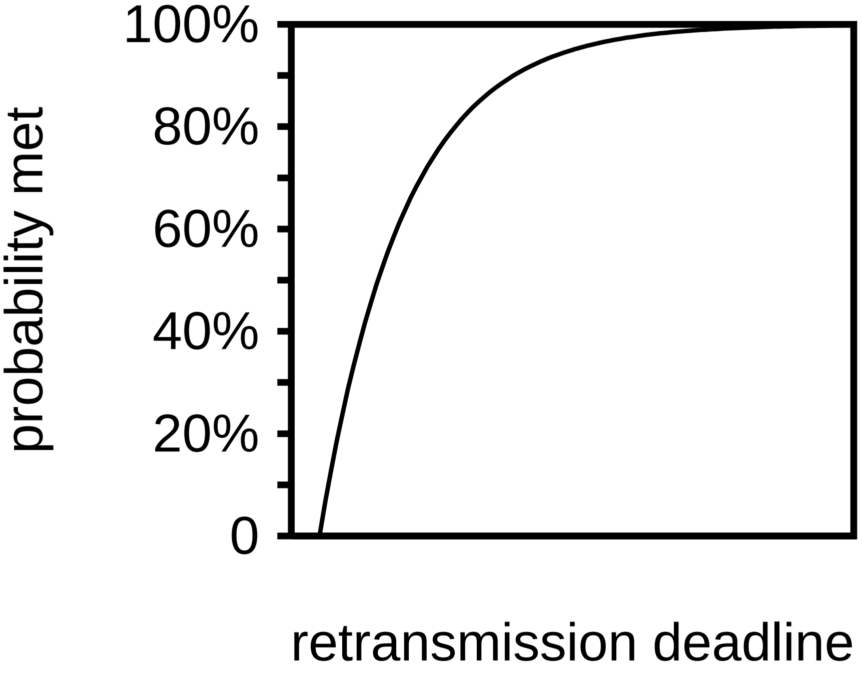
- directed multicast responses, like Replier

- request traffic:
  outside loss subtree: μ total
  inside, upward: μ on each link
  inside, downward: $< 2\mu$ on each link

- local retransmissions: exposure $= O(1 + \ln \frac{N}{L})$

Search Party Reliable Multicast (cont.)

- retransmission delay: independent of L, N

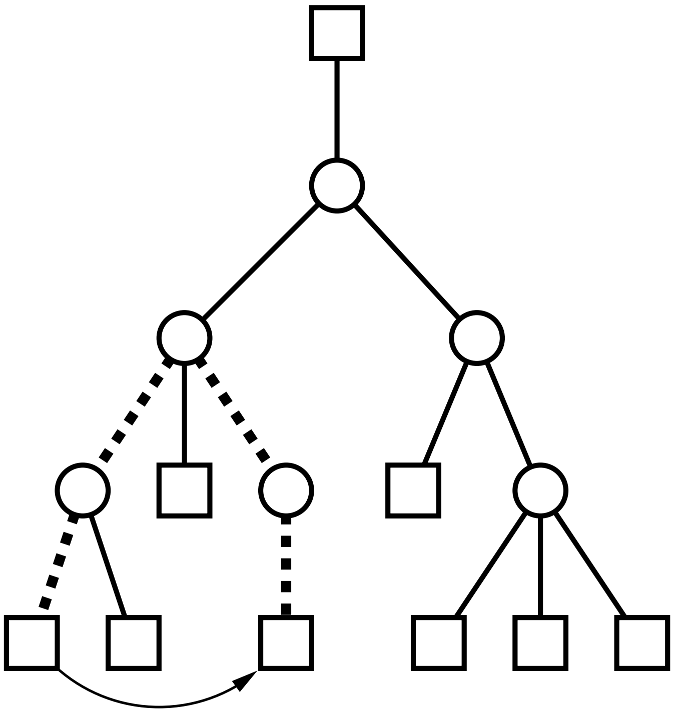


- fine-tunable delay/bandwidth tradeoffs

- flakiness is contained

Virtual Randomcast

- receivers use subcast to estimate subtree sizes, and perform randomcast "routing" *themselves*



- benefit: more easily deployable
  costs:
  - $O(1 + \ln N)$ overhead traffic per receiver
  - unknown per-link request traffic (off-tree)
  - one of:
    - router addresses exposed to receivers
    - no bound on exposure
    - double retransmission delay

Future Work

- Simulate (in progress)
- Compare to SRM & RMTP
- Unify with SRM?

Conclusion

Search Party is a promising scheme that may prove to be:

- efficient
- scalable
- tunable
- robust
- deployable