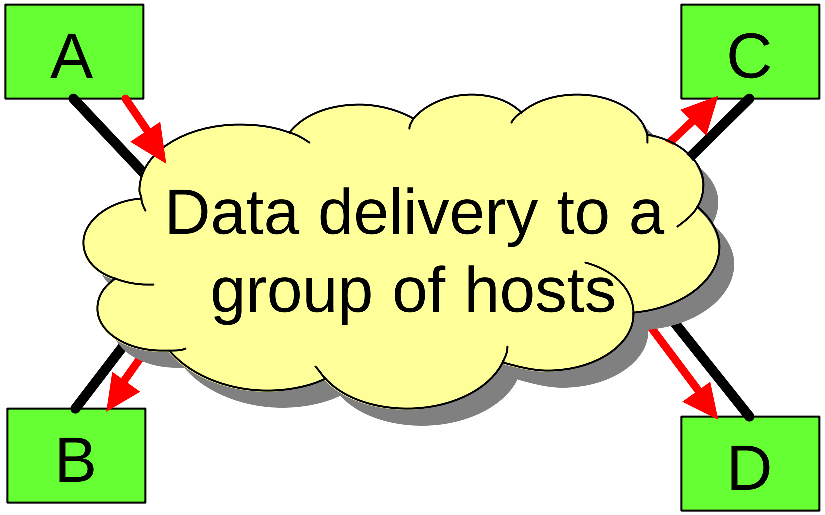
Resilient Multicast using Overlays

Aravind Srinivasan

Suman Banerjee, Ryan Braud,
Seungjoon Lee, Bobby Bhattacharjee

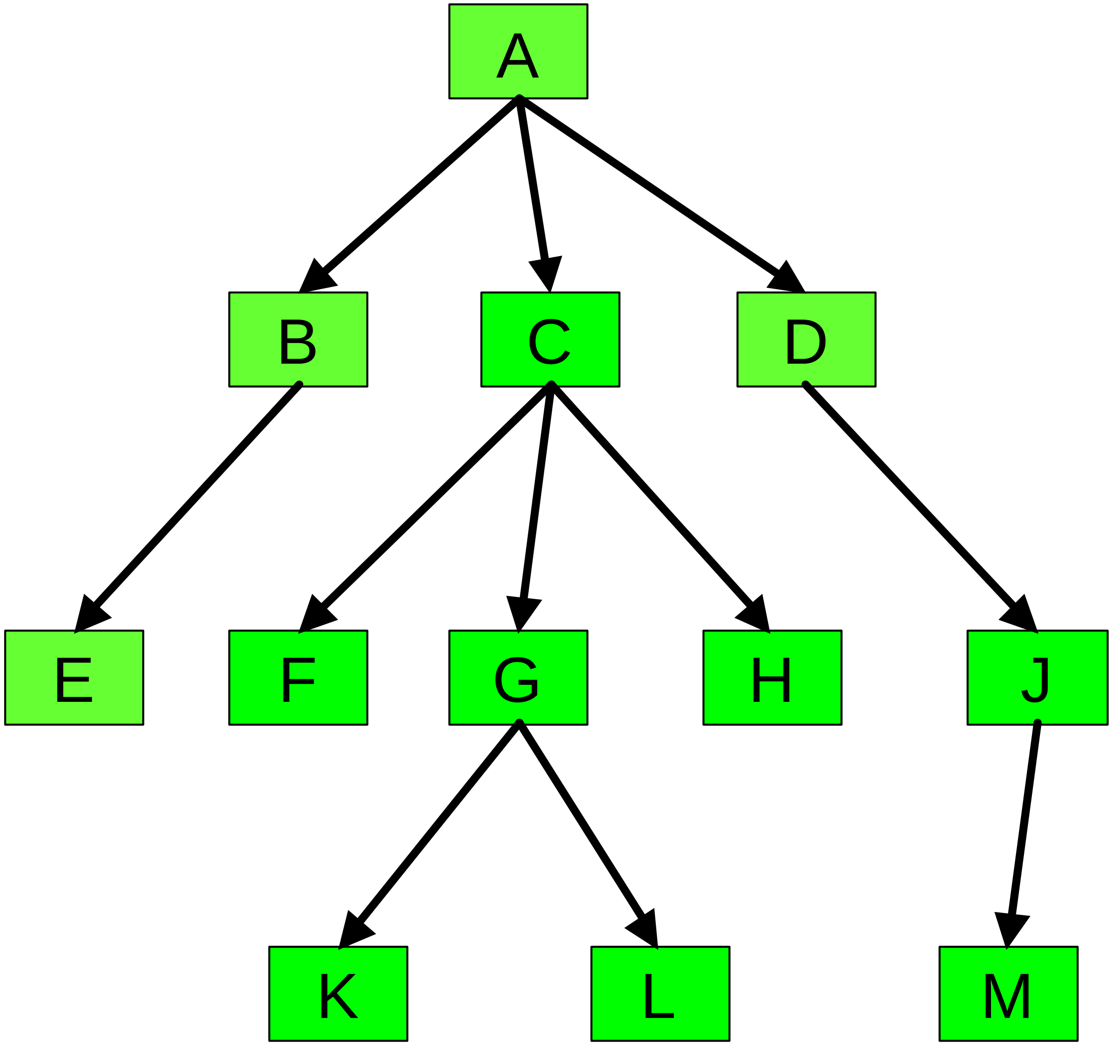
UNIVERSITY OF MARYLAND

Background

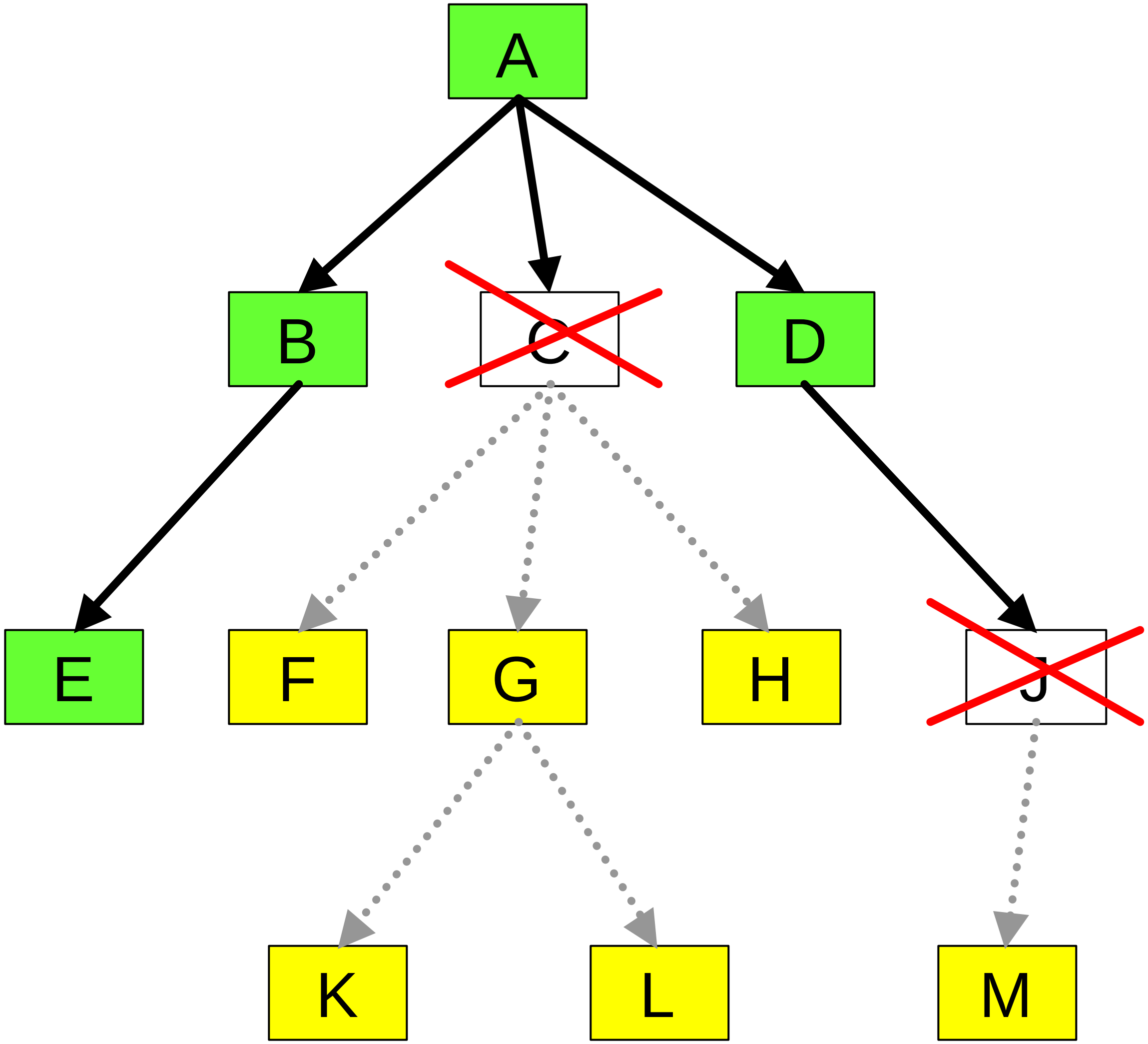


- **Network-layer Multicast**
- **Application-layer multicast**

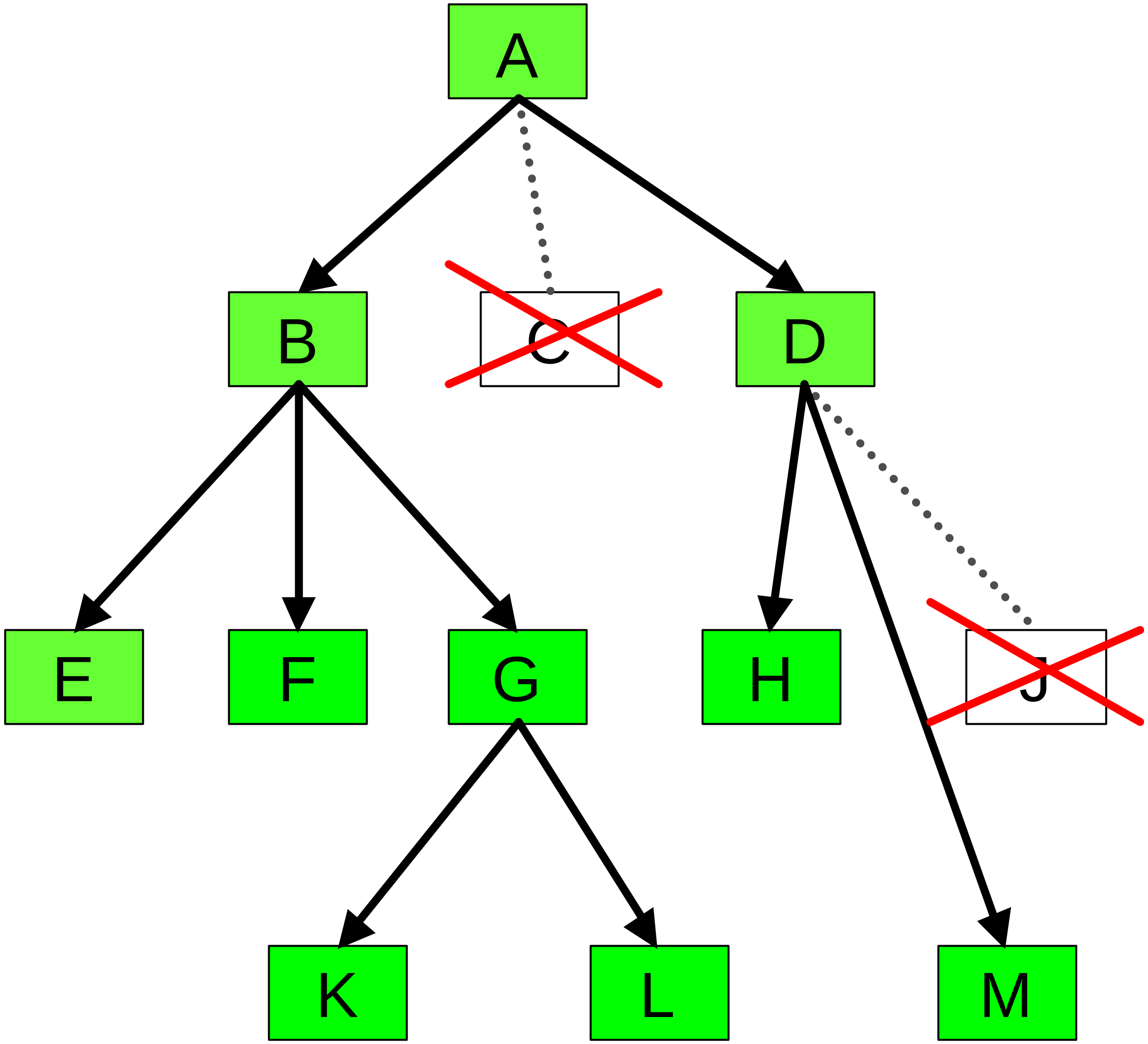
Overlay Node Failures



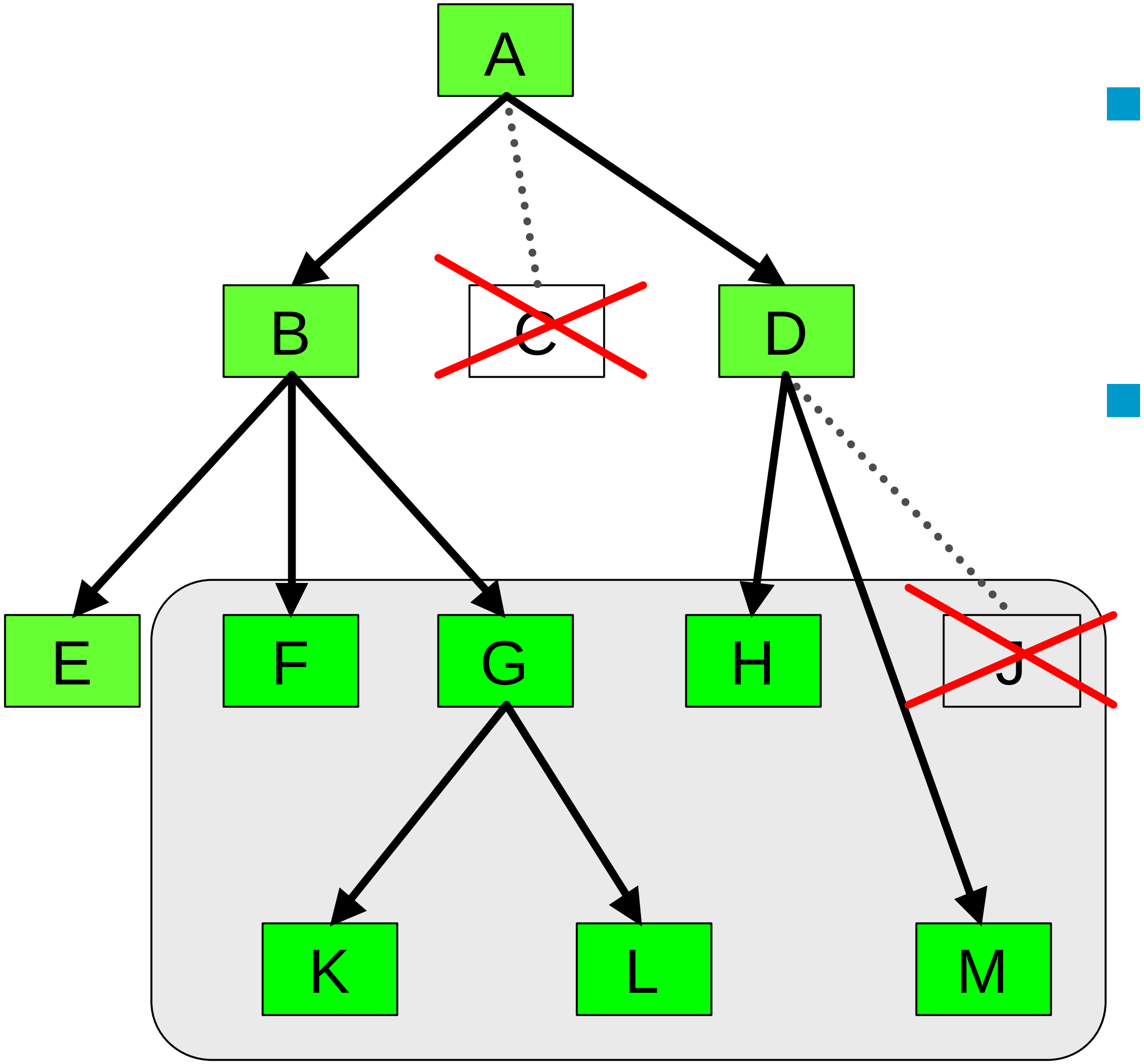
Overlay Node Failures



Overlay Node Failures



Overlay Node Failures



- Network losses are transient
- Overlay node failures are persistent until detected and repaired
  - ~ 10s of seconds

Resilience

High delivery ratio

Have latency requirements

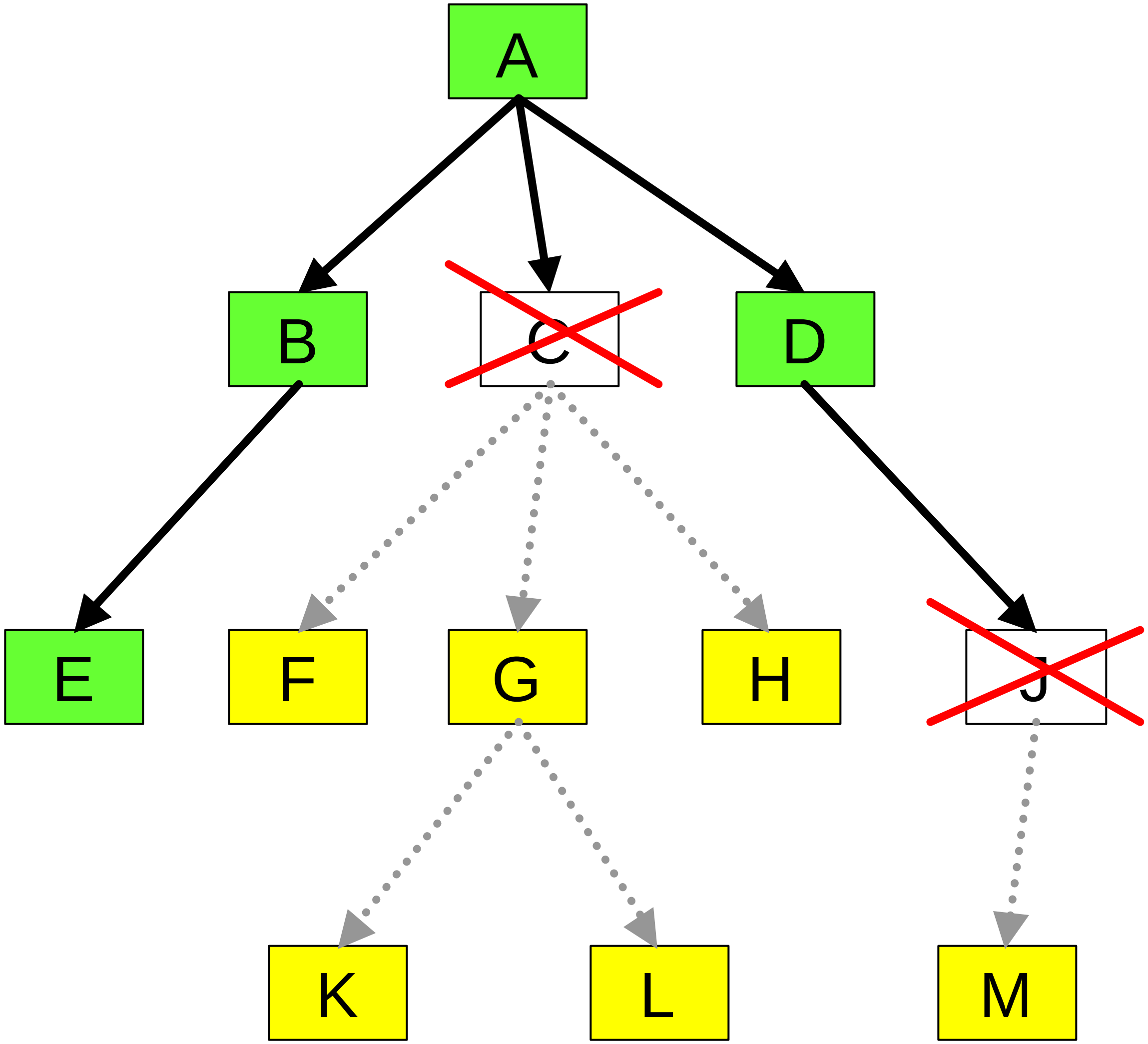
Streaming audio or video applications

Solution: Redundant, low-overhead data paths

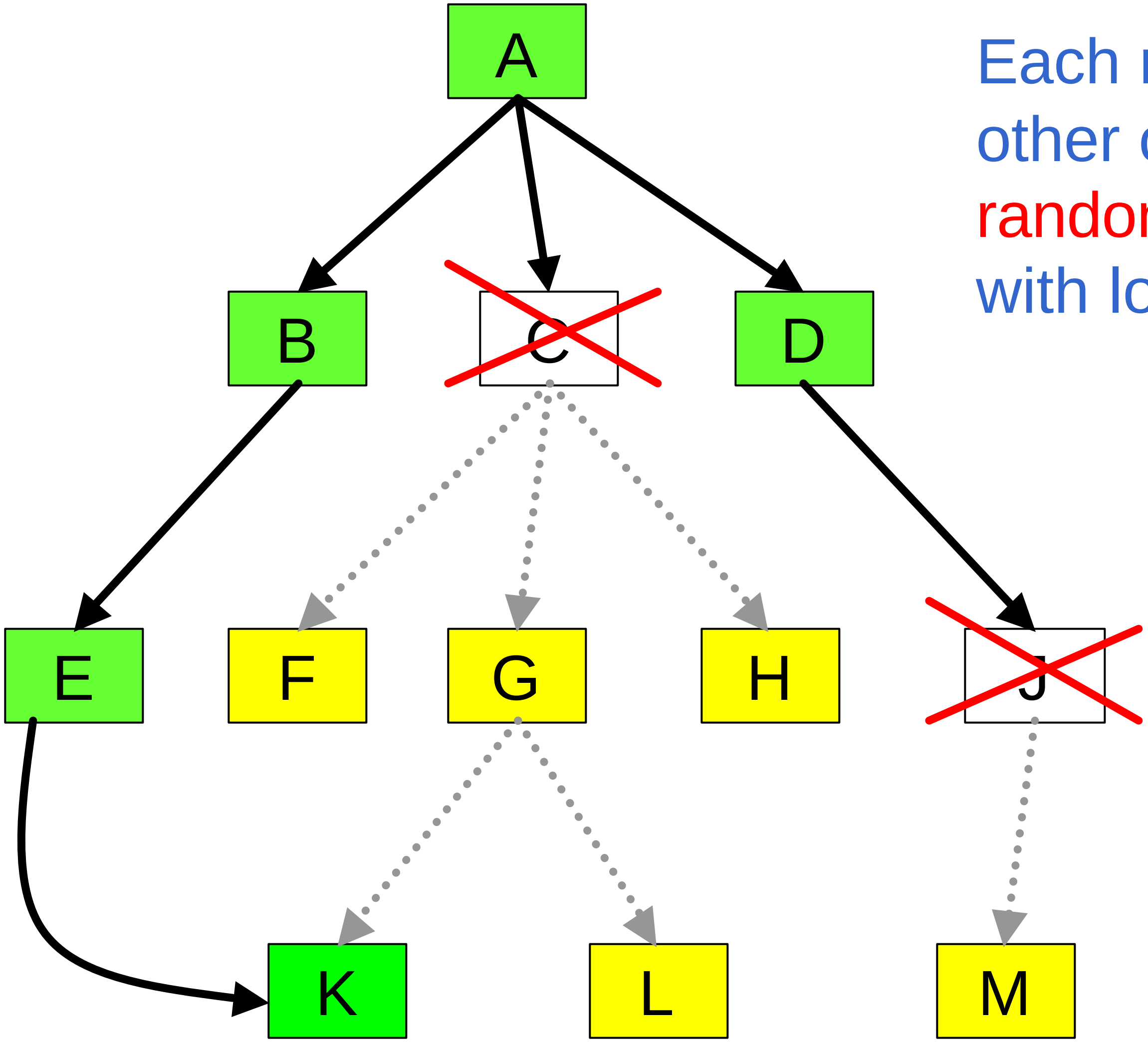
Probabilistic Resilient Multicast

- Randomized Forwarding
  - Handles overlay node failures
  - Proactive
- Triggered NAKs
  - Handles network losses
  - Reactive

Randomized Forwarding

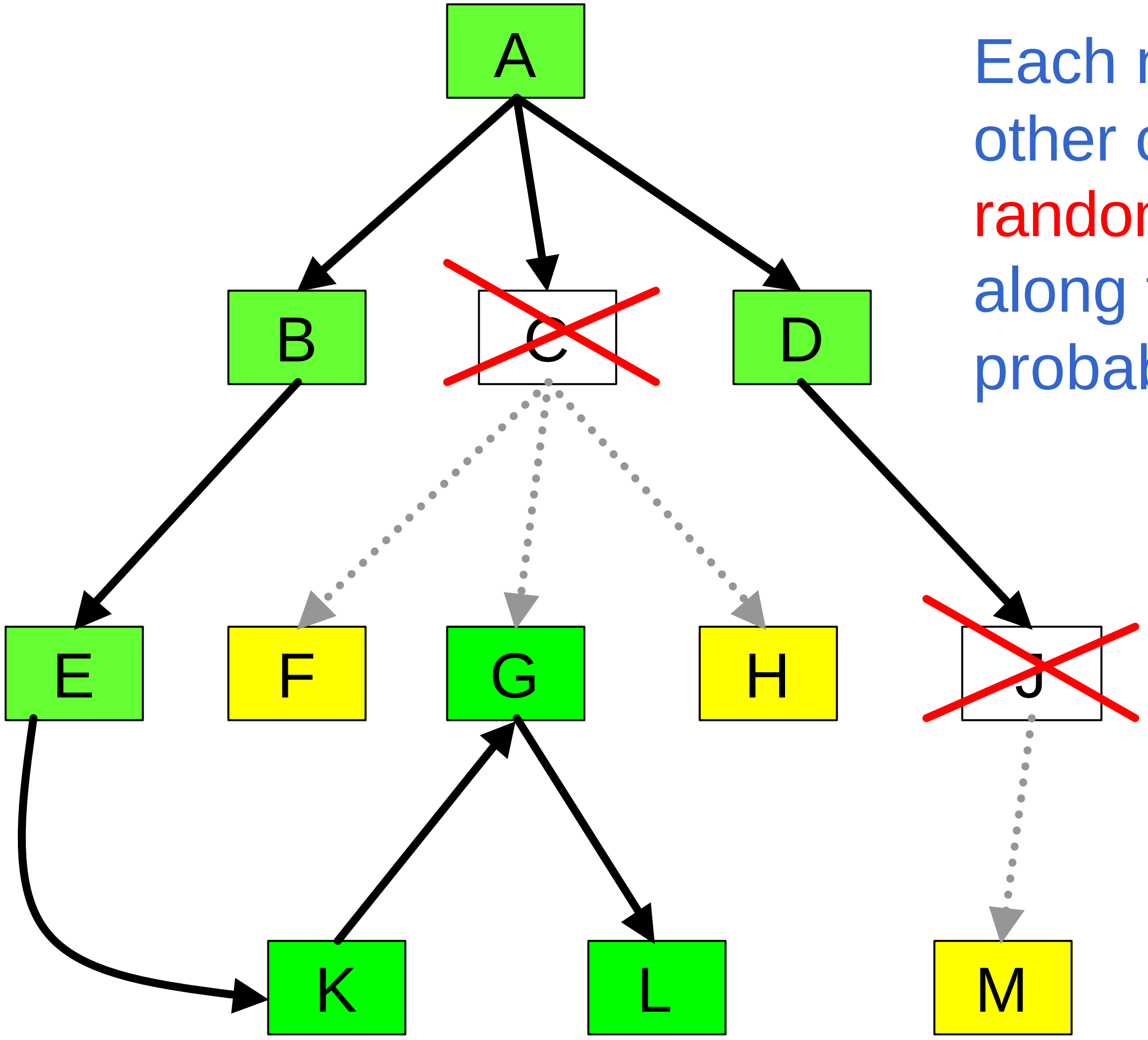


Randomized Forwarding



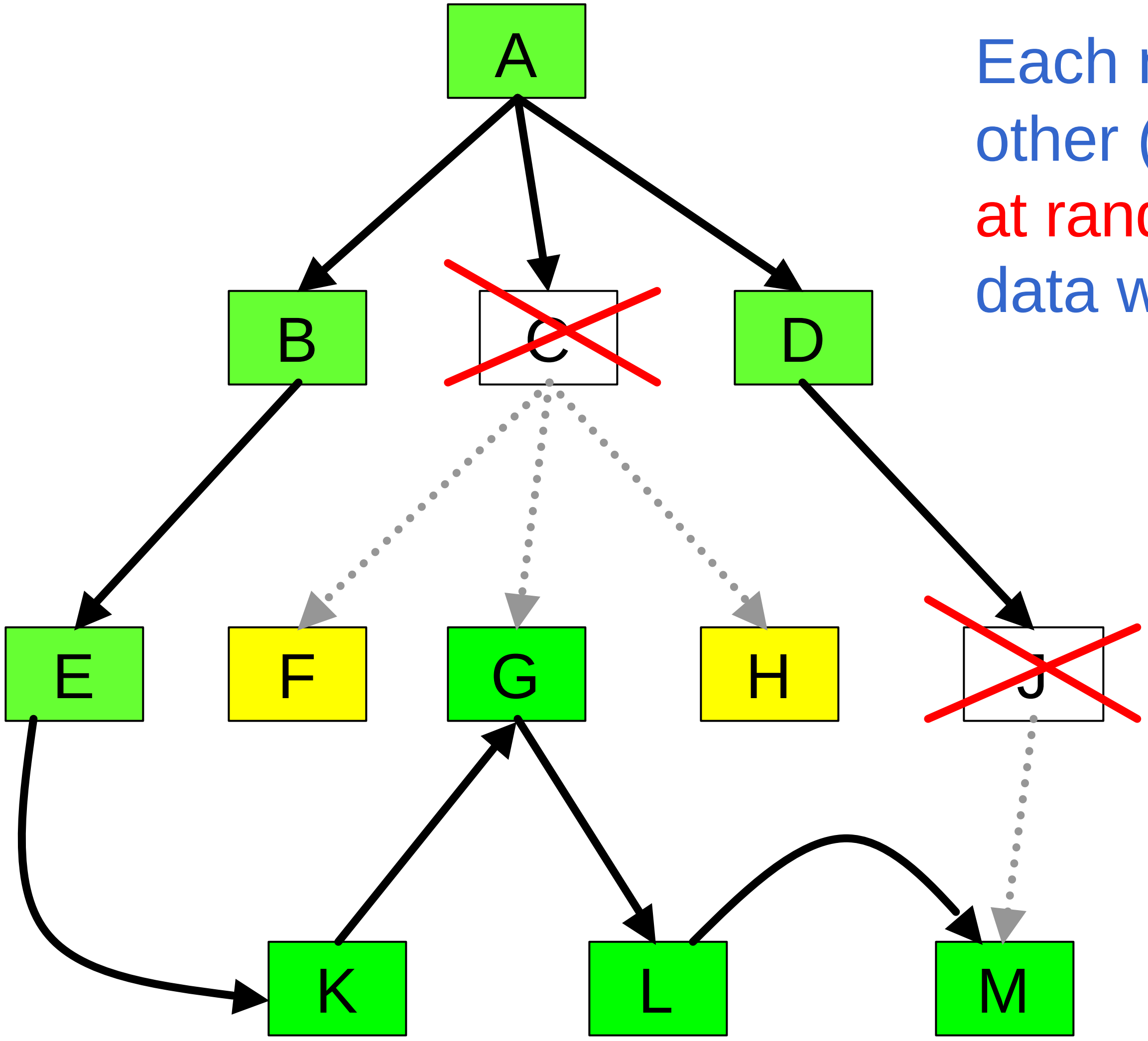
Each node chooses a few other cross tree edges at random and forwards data with low probability

Randomized Forwarding



Each node chooses a few other cross tree edges at random and forwards data along them with low probability

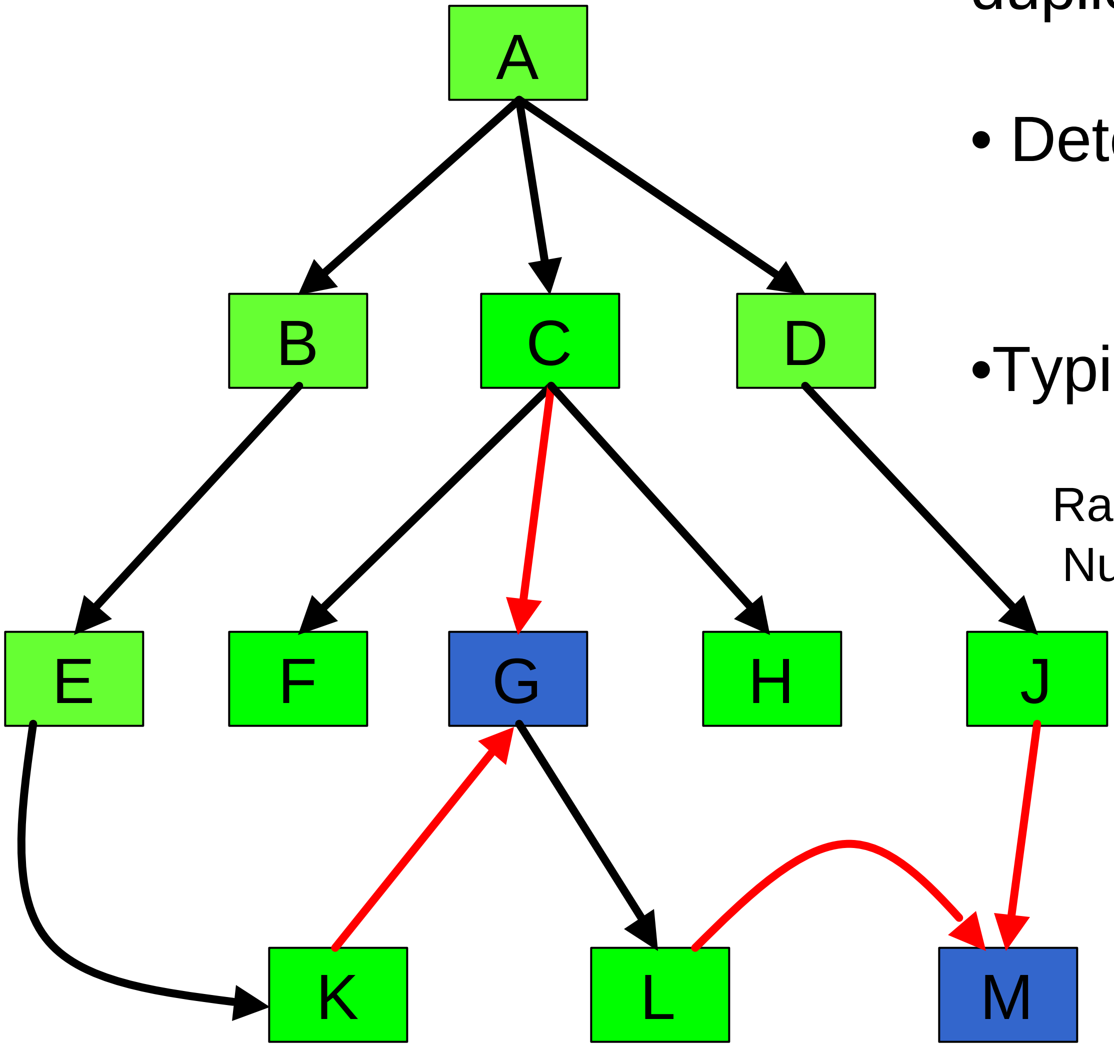
Randomized Forwarding



Each node chooses a few other (r) cross tree edges at random and forwards data with low probability (β)

Uses a low overhead random node discovery mechanism

Overheads



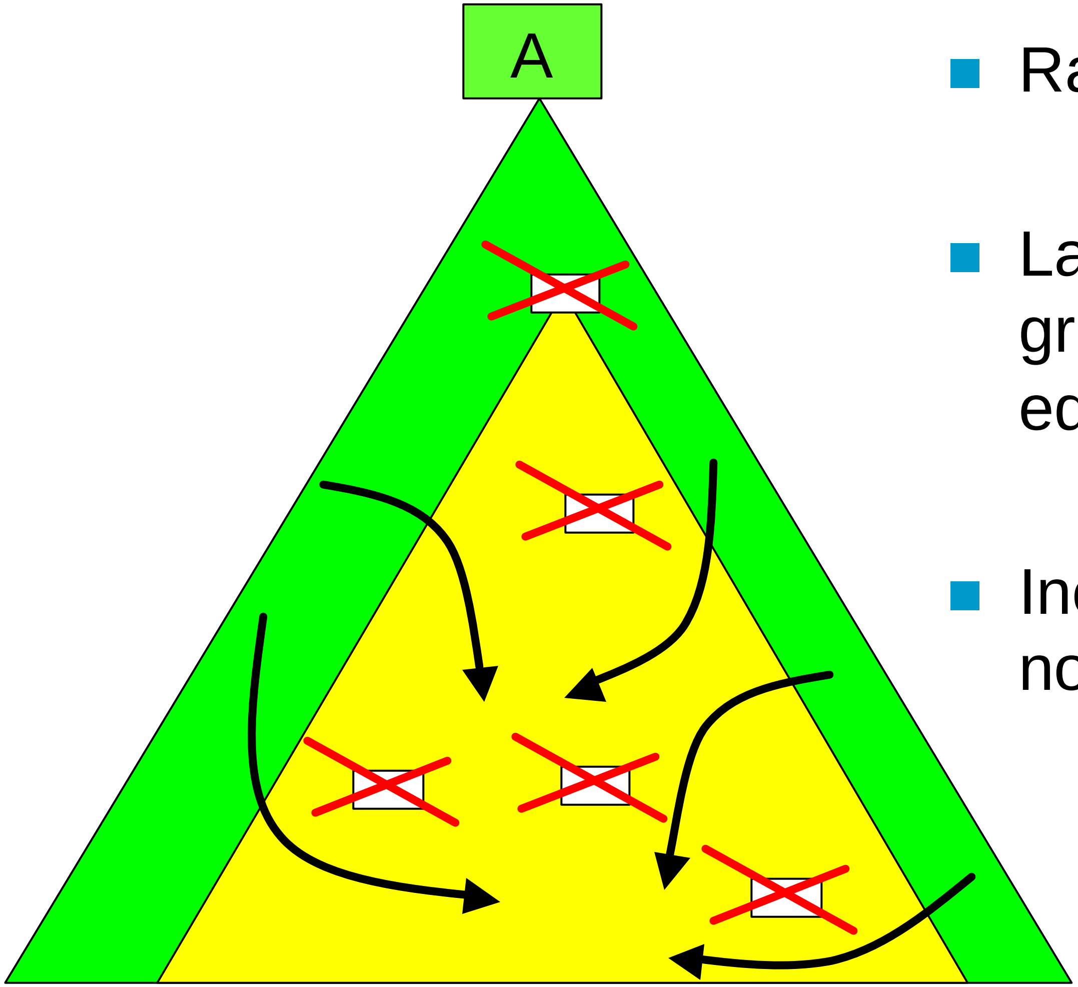
- Data overhead due to duplicates

- Detected and suppressed

•Typical overheads (r β) = 3%

Random forward probability = 0.01
Number of random neighbors = 3

It performs very well!

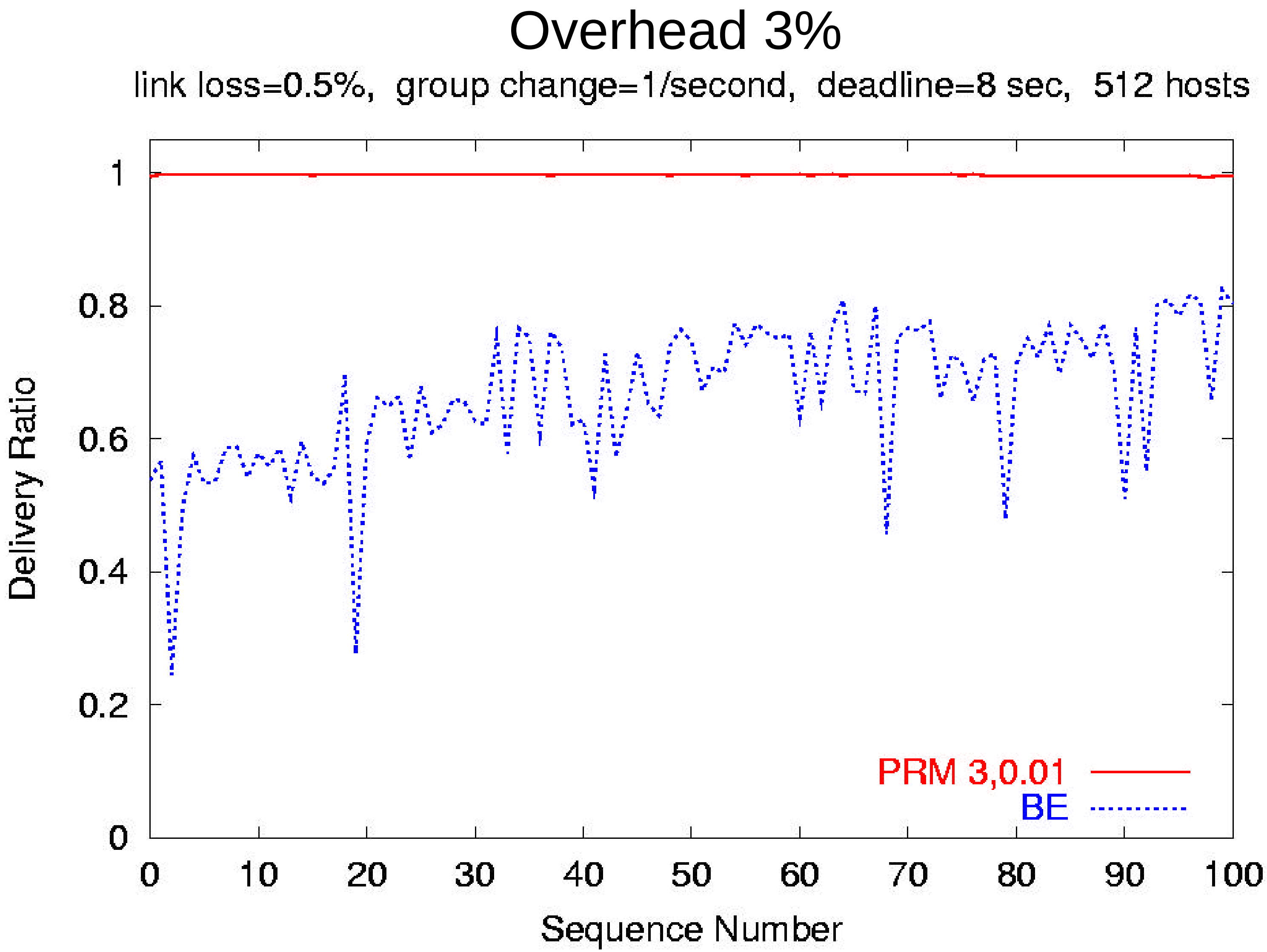


- Random choices help
- Larger the affected region, greater the number of cross edges incident on it
- Increases resilience against node failures

Analysis

- With high probability,
  - All the non-leaf nodes that did not fail successfully get data.
  - A large fraction of leaf nodes that did not fail successfully get data. (e.g. 97%)

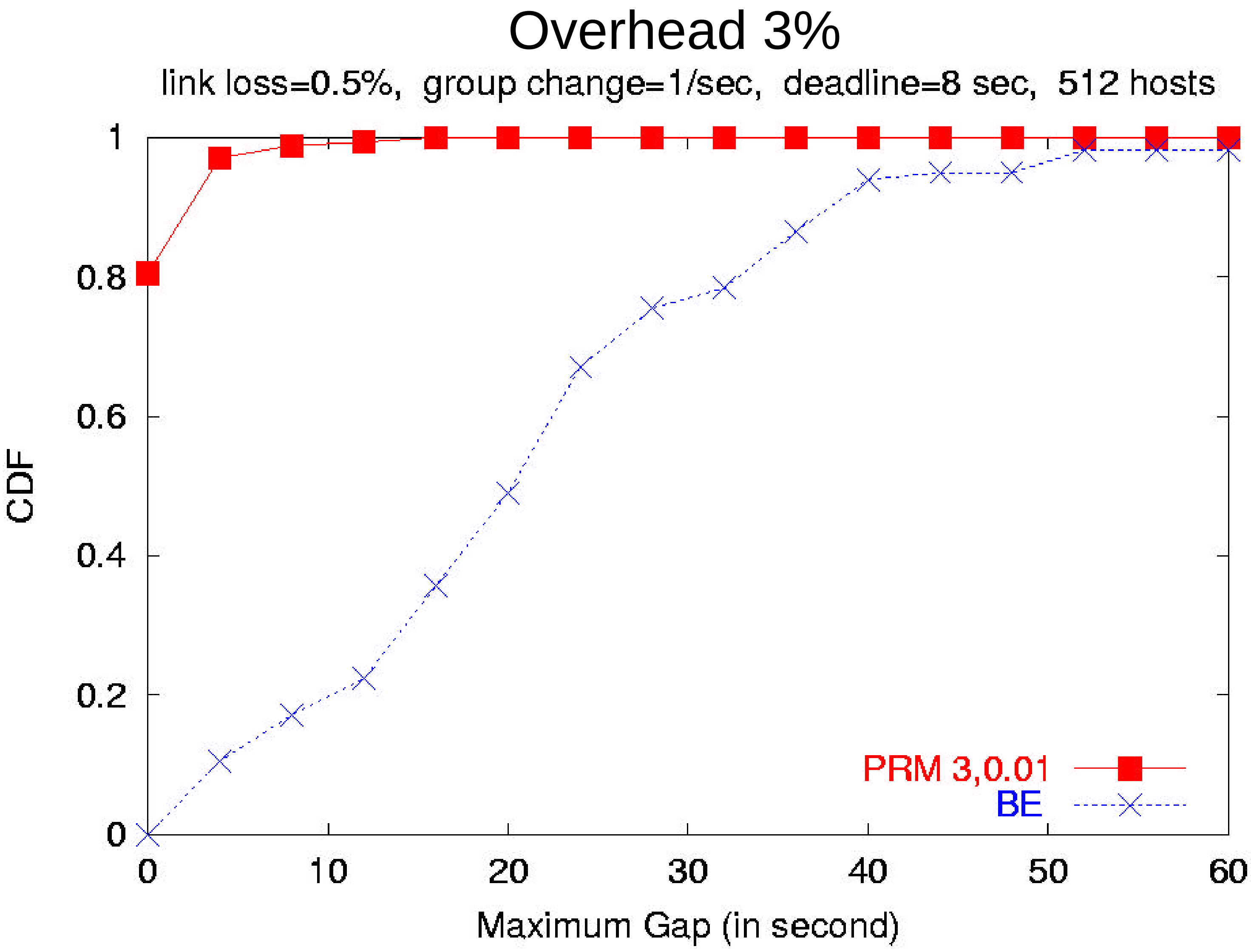
Data Delivery Ratio



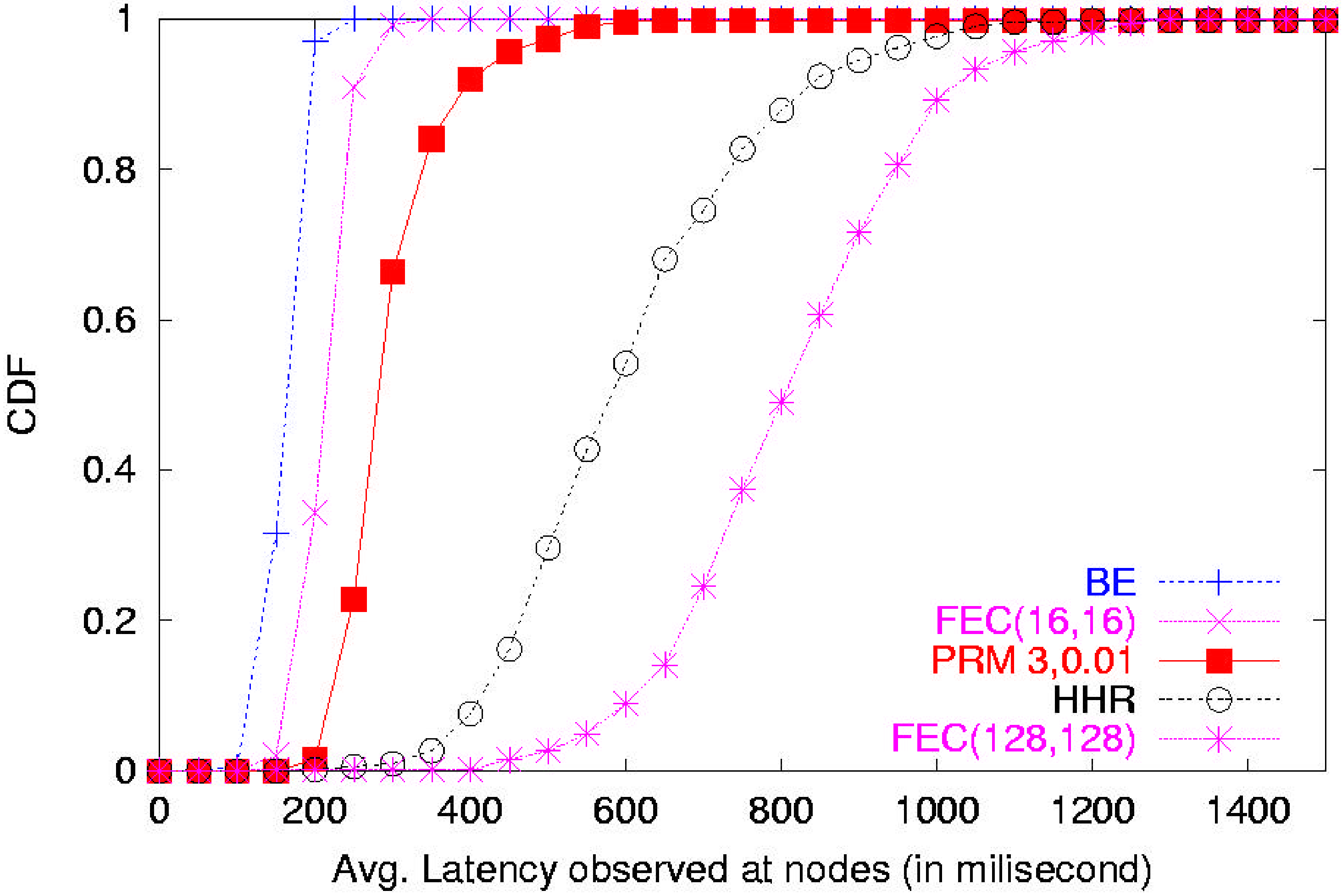
Overhead 3%

link loss=0.5%, group change=1/second, deadline=8 sec, 512 hosts

Maximum Data Outage



Latency of Data Delivery



Overhead 3%

link loss=0.5%, group change=0.1/sec, deadline=8 sec, 512 hosts

Data Overheads

Scheme, Change rate	Latency (s)	Data Delivery Ratio				
		80%	85%	90%	95%	99%
FEC 0.1/s	0.5	**87-100**	**-**	**-**	**-**	**-**
	2.0	**62-75**	**-**	**-**	**-**	**-**
	8.0	**50-62**	**75-87**	**-**	**-**	**-**
	64.0	**37-50**	**50-62**	**62-75**	**75-87**	**87-100**
PRM 1.0/s	0.2	**9-12**	**18-21**	**21-24**	**30-60**	**-**
	0.5	**0-1**	**1-3**	**3-6**	**9-15**	**30-60**
	2.0	**0-1**	**0-1**	**0-1**	**0-1**	**3-9**
	8.0	**0-1**	**0-1**	**0-1**	**0-1**	**1-3**

Scalability

Size	Control Overheads		Delivery Ratio	
	BE	PRM	BE	PRM
128	**2.9**	**4.0**	**0.68**	**0.99**
256	**3.3**	**4.4**	**0.58**	**0.99**
512	**3.4**	**4.7**	**0.60**	**0.99+**
1024	**4.1**	**5.5**	**0.51**	**0.98**
2048	**5.8**	**7.4**	**0.41**	**0.97**
4096	**10.1**	**13.5**	**0.40**	**0.97**

Experimentation on Internet

- Scalable Resilient Media Streaming System
  - MPEG4IP player
  - MPEG-4 movie clip streamed cyclically from Darwin Media Streaming Server
  - PRM-enhanced NICE application-layer multicast
- Tested on Emulab and RON testbed
  - RON testbed: 32 hosts in USA, Canada, Europe, Asia
- Dynamic joins and leaves of clients

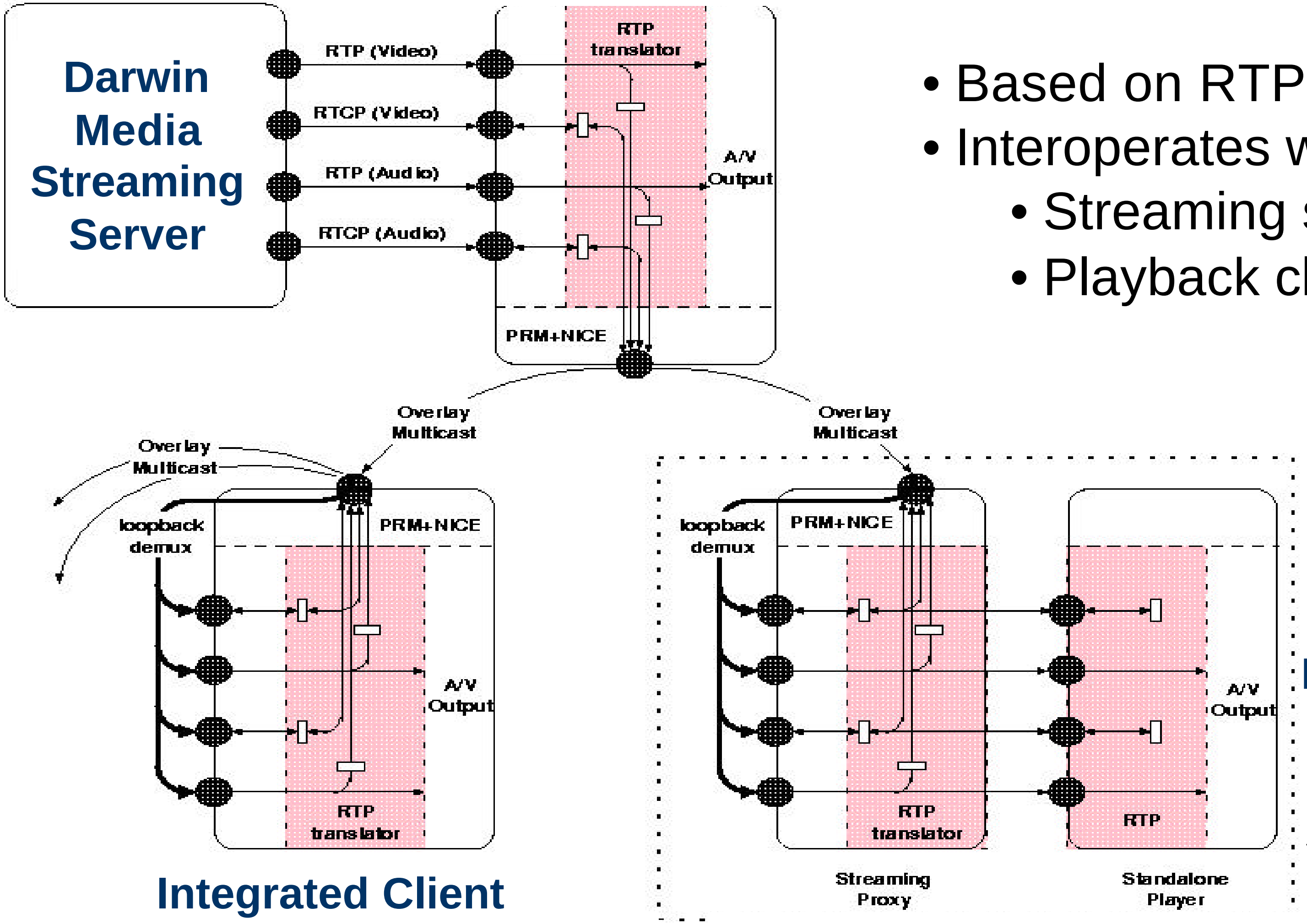
Implementation

PRM-based media streaming system implemented and tested on the Internet

- Based on RTP
- Interoperates with any:
  - Streaming server
  - Playback client

From apple.com

Designated Source



Darwin Media Streaming Server

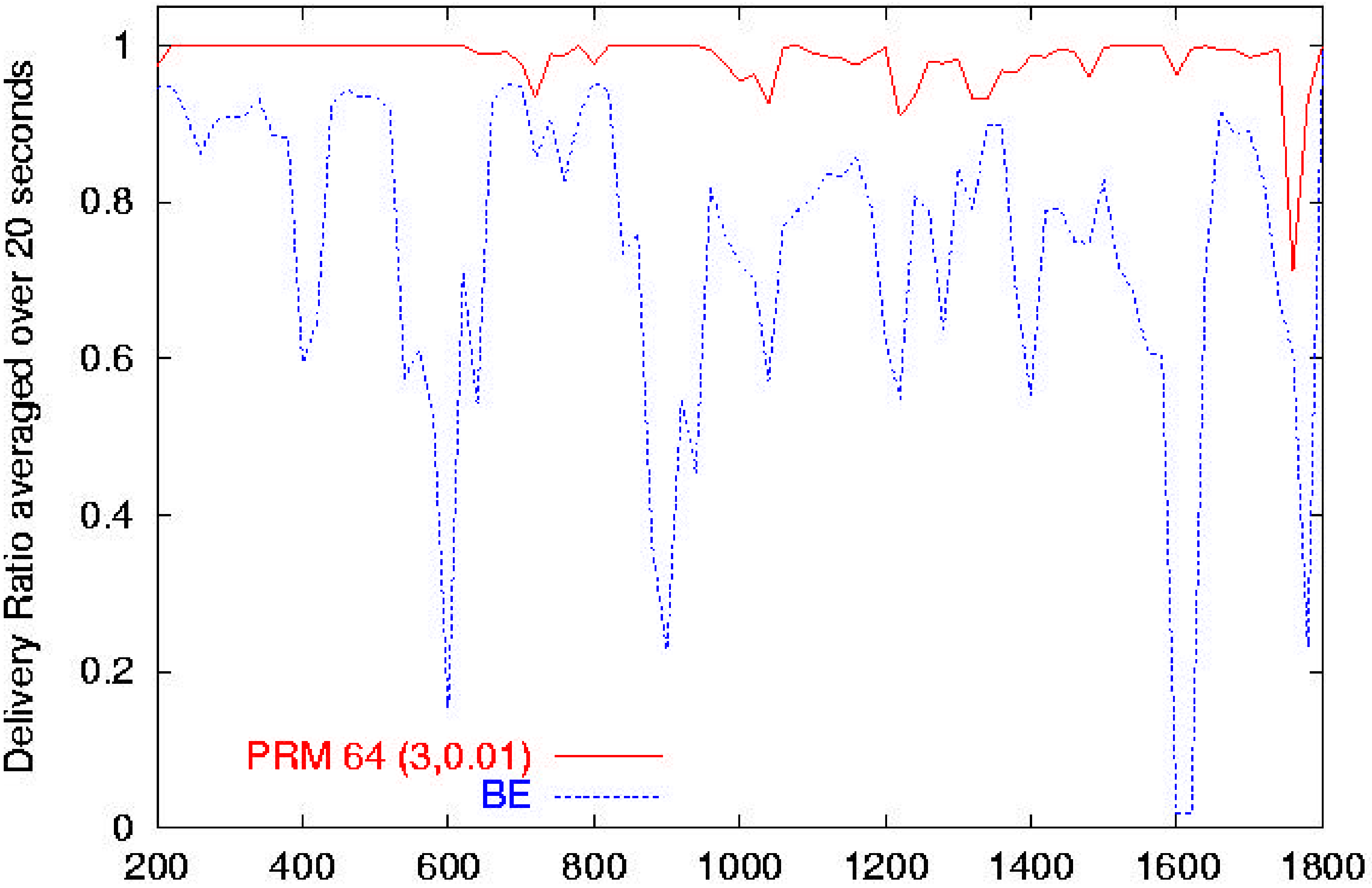
Integrated Client

MPEG4IP player

From sourceforge.net

Proxy-based Client

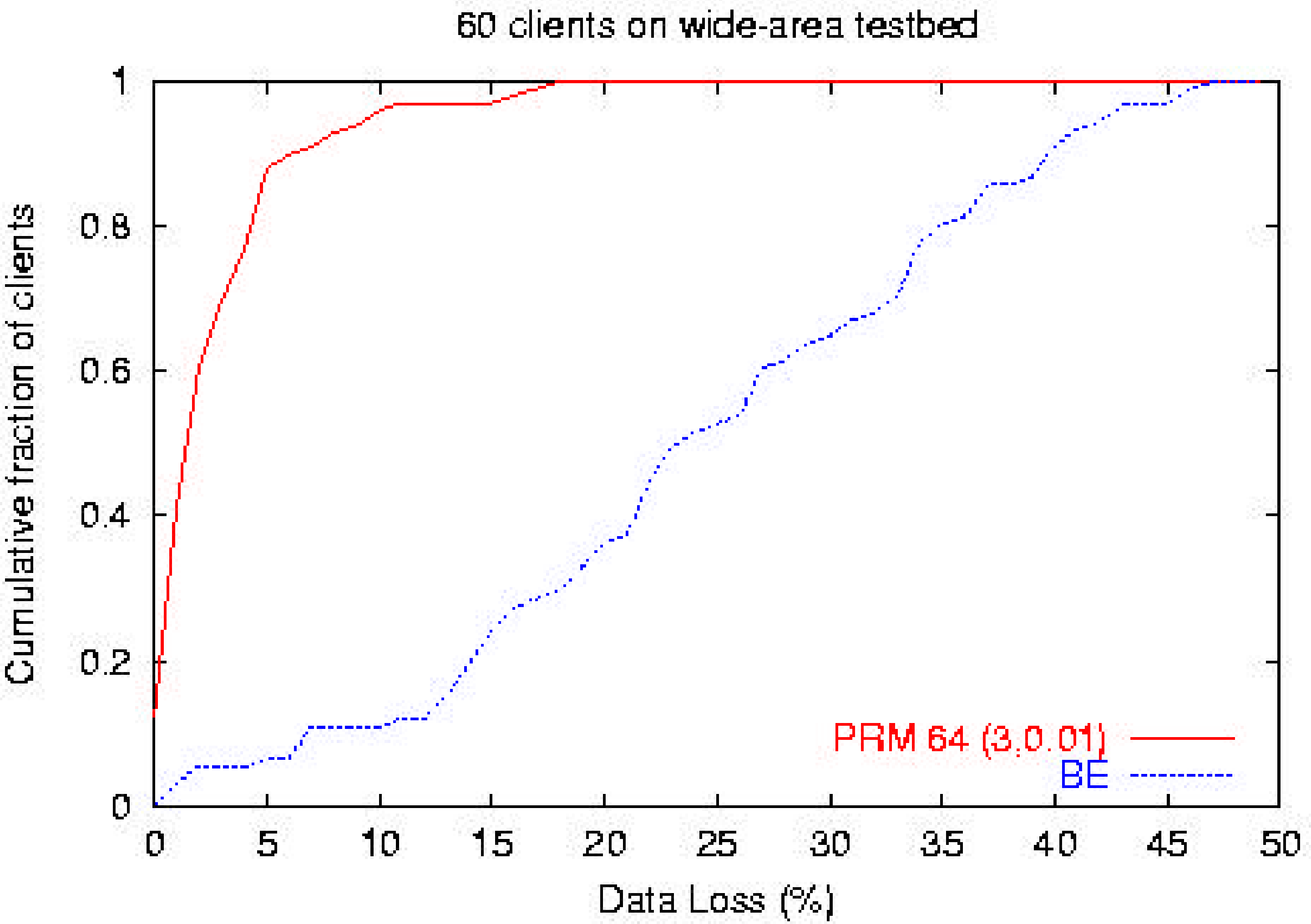
Data Delivery Ratio



PRM incurs 3% overhead

Group change: 4.8/min

Aggregate Delivery



PRM incurs 3% overhead
Group change: 4.8/min

Summary

- NICE-resilient multicast
  - Low overhead
  - High data delivery
  - Efficiently scales to large groups
- Implementation and Experimentation on the Internet
  - Implementation of a media streaming service
  - Experiments on Emulab and RON testbed

www.cs.umd.edu/projects/nice/papers/cs-tr-4482.pdf