Scalable Application Layer Multicast

Suman Banerjee

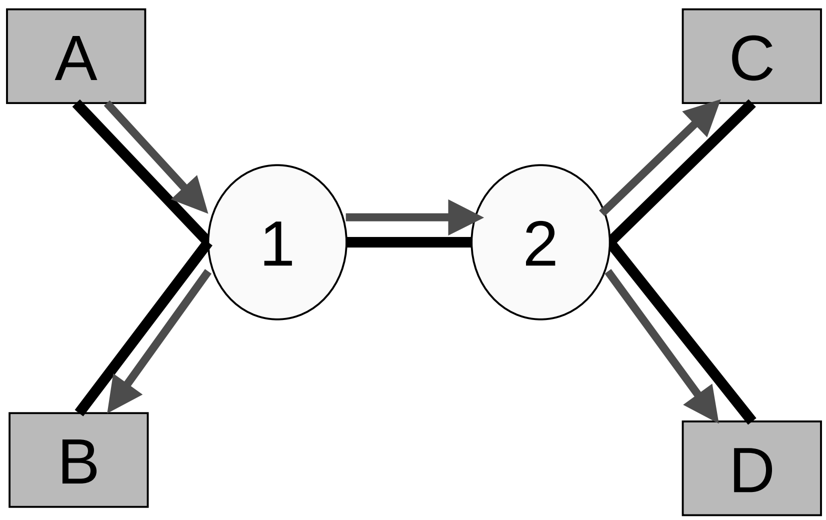
Bobby Bhattacharjee

Christopher Kommareddy

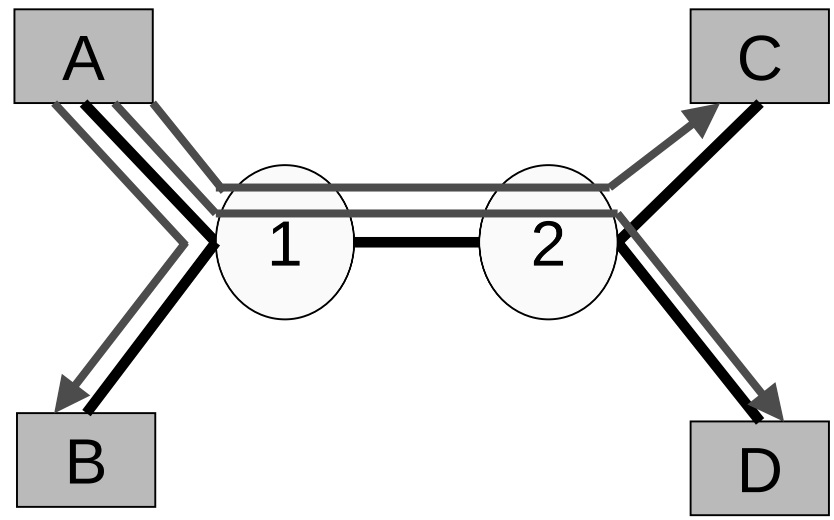
UNIVERSITY OF MARYLAND

http://www.cs.umd.edu/projects/nice

Group Communication

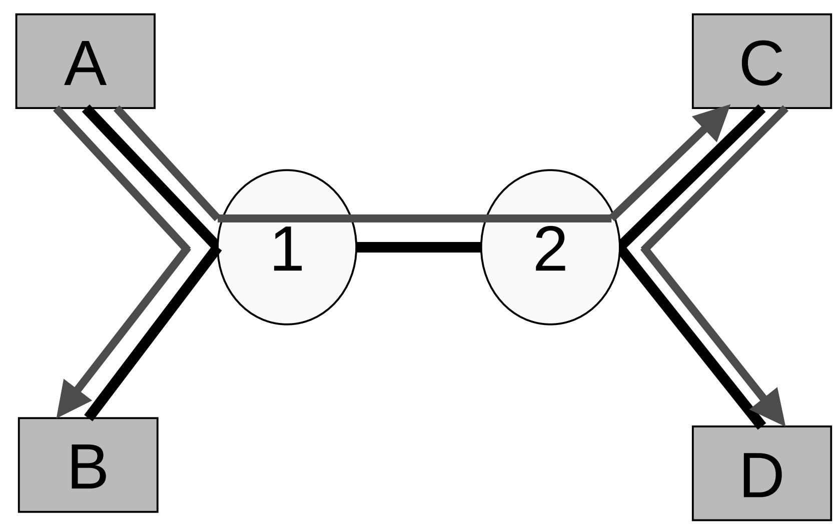


Network-layer Multicast
Replication at routers



Sequence of Direct Unicasts
Replication only at source

Application-layer Multicast

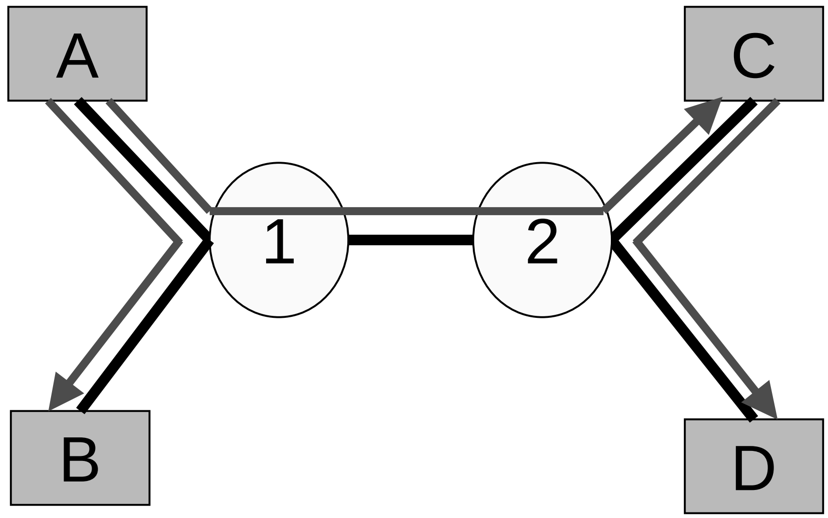


Replication at end-hosts

Examples:

- Narada, Yoid, Gossamer, HMTP, Scribe, Bayeux, CAN-multicast, DT, …
- **NICE**

Application-layer Multicast



Replication at end-hosts

Metrics

Tree Quality

State / Control Overheads

Robustness

Talk Outline

- Introduction
- NICE Application-layer Multicast Protocol
- Results
- Conclusions

NICE Application-layer Multicast

- Scales to large group sizes
  - Low average and worst case control overheads
  - Does not compromise tree quality or robustness
- Even low-bandwidth applications are efficient
  - Web tickers

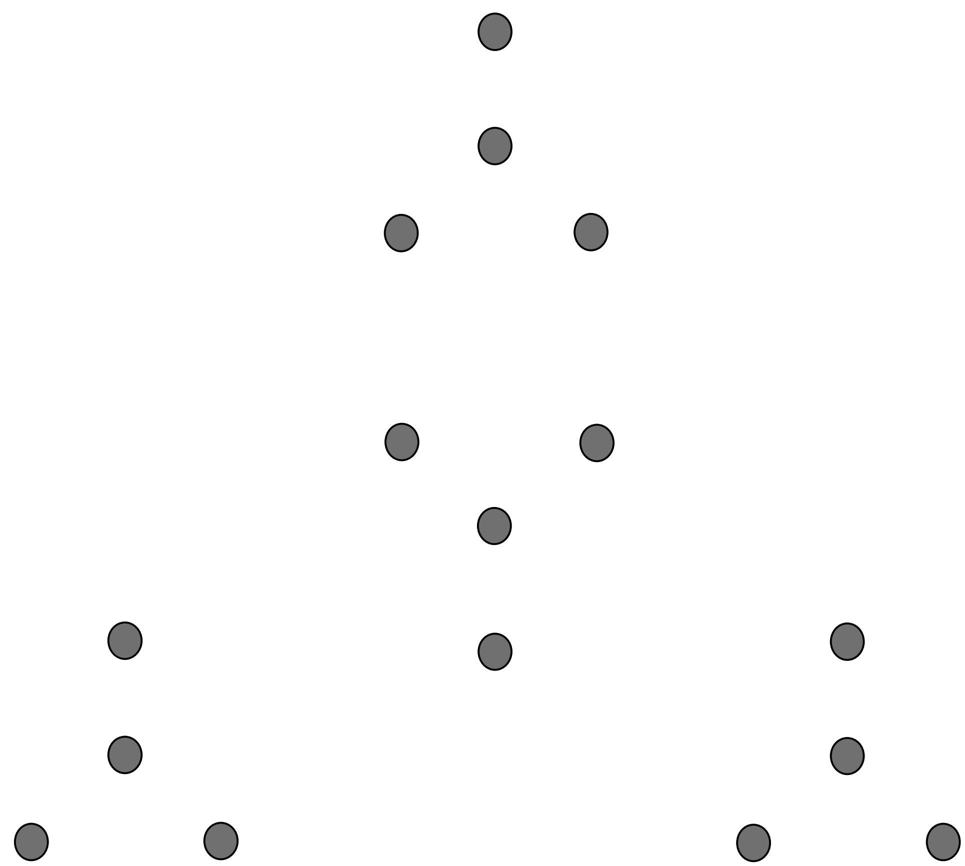
Uses a hierarchy

UNIVERSITY OF MARYLAND TERRAPINS

NICE Topologies

- Control topology
  - Detects host failures and re-structure the overlay
- Data delivery topology
  - Basic path: Implicitly defined by the hierarchy
  - Can be independent of the control path

NICE Hierarchy

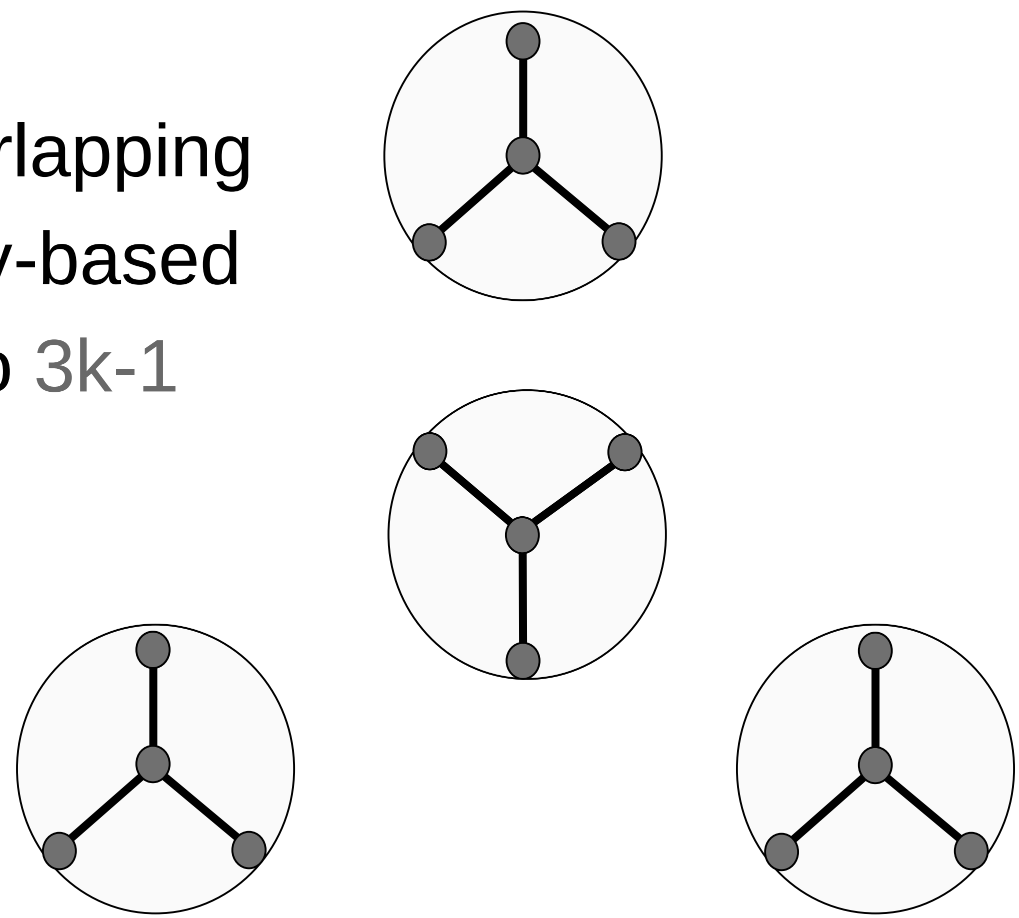


A Set of Members

NICE Hierarchy

Clusters

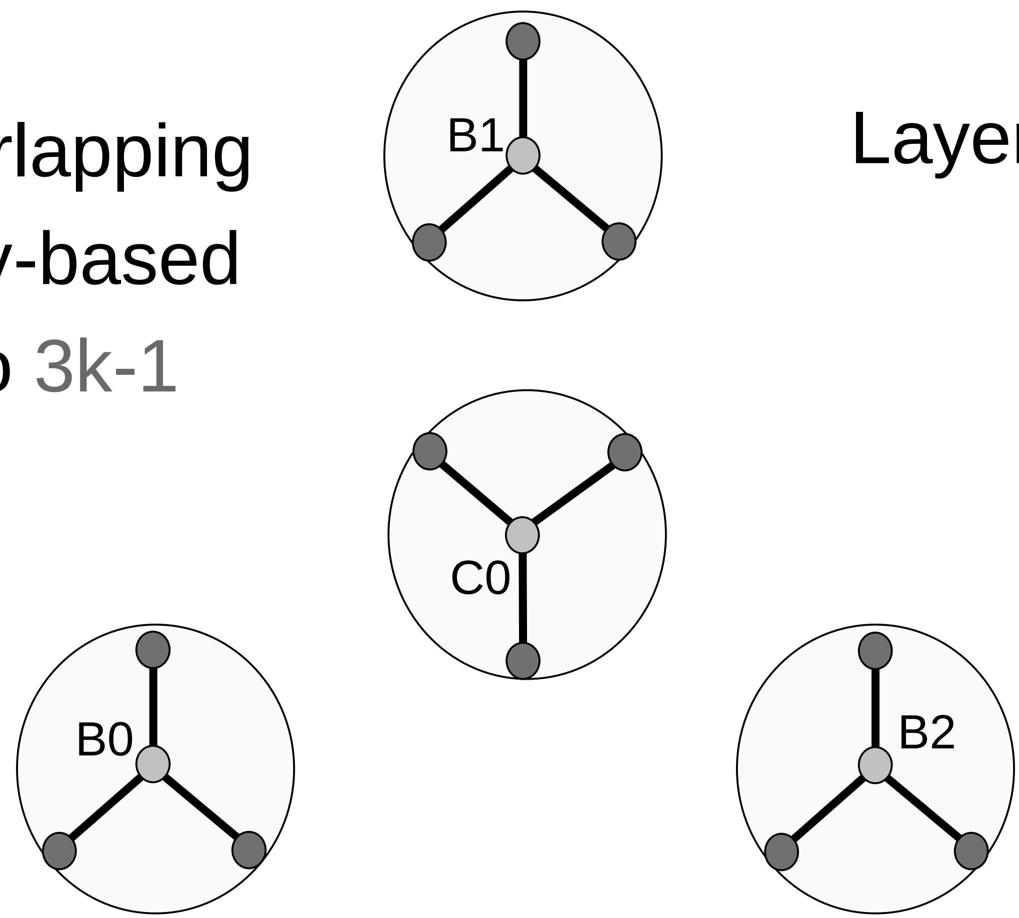
- Non-overlapping
- Proximity-based
- Size: k to 3k-1



NICE Hierarchy

Clusters

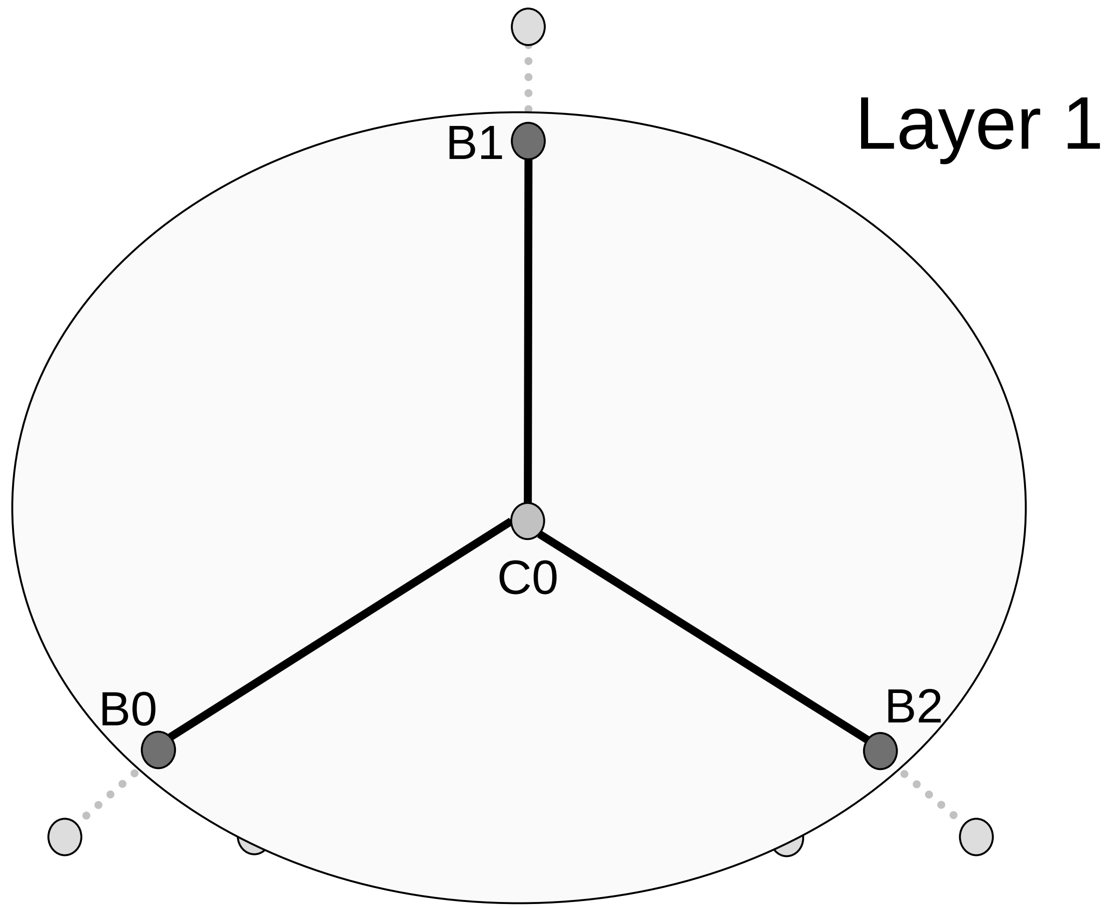
- Non-overlapping
- Proximity-based
- Size: k to 3k-1



Layer 0

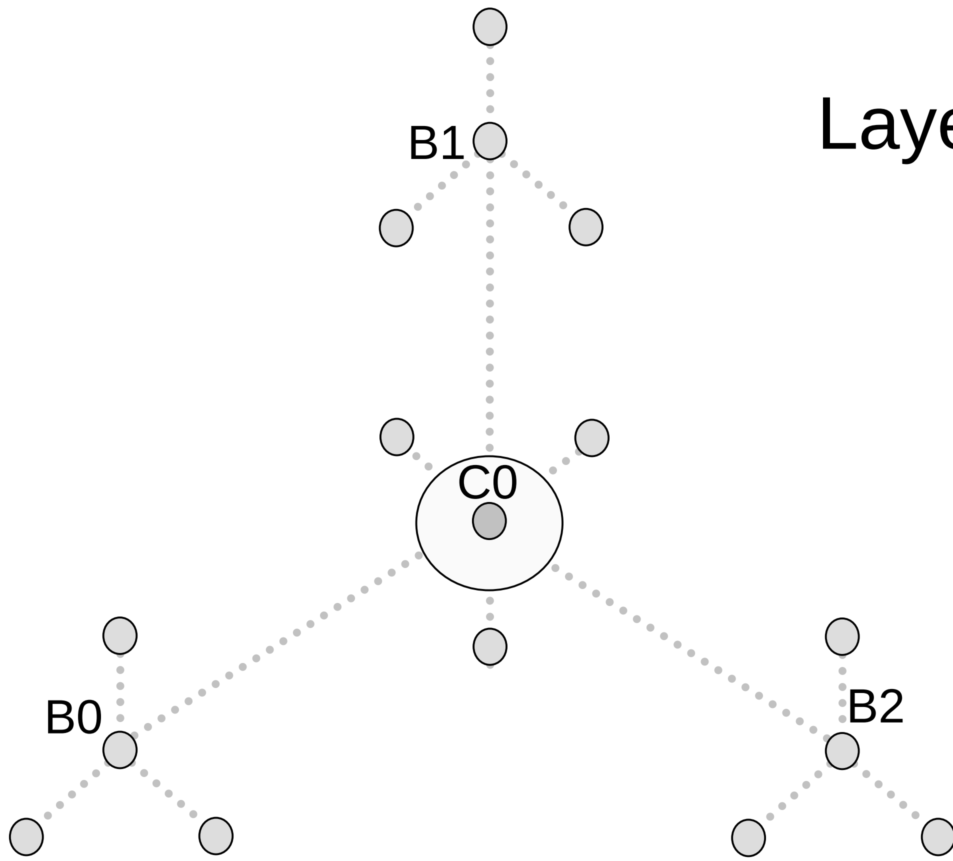
Graph-theoretic center is the cluster leader

NICE Hierarchy



Leaders form the higher layer and repeats

NICE Hierarchy

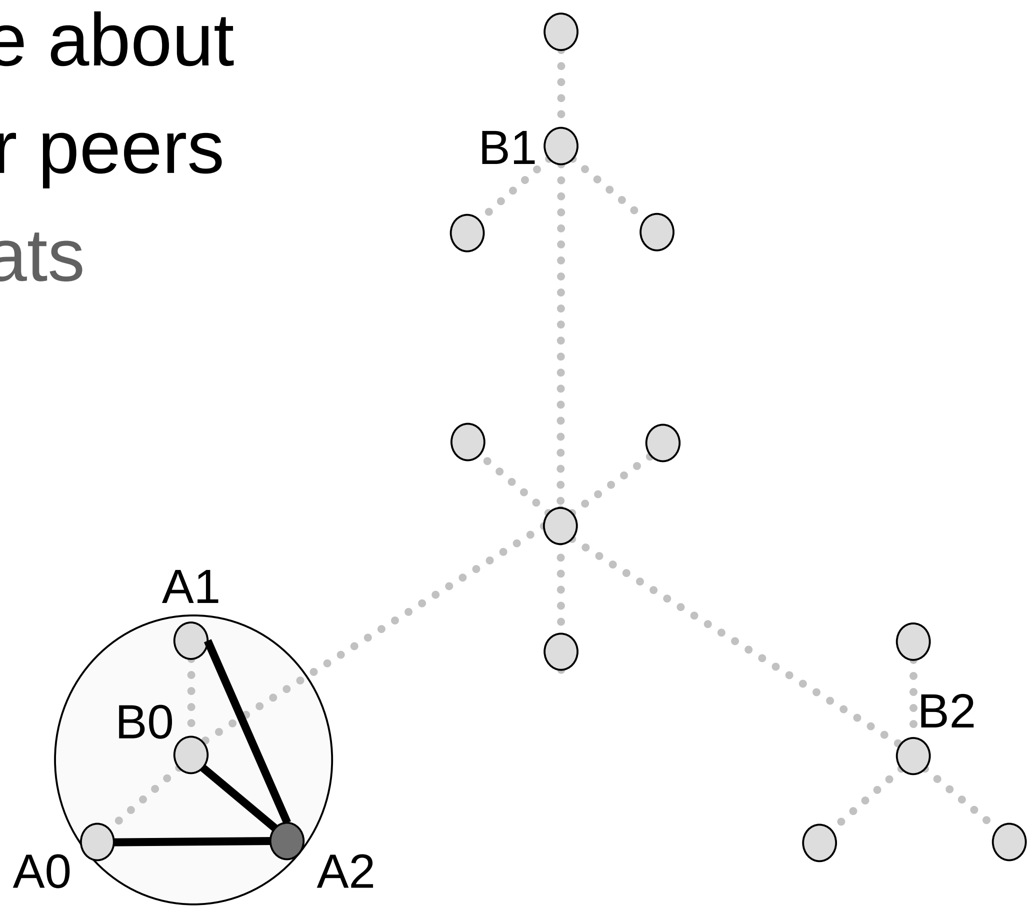


Layer 2

log N layers

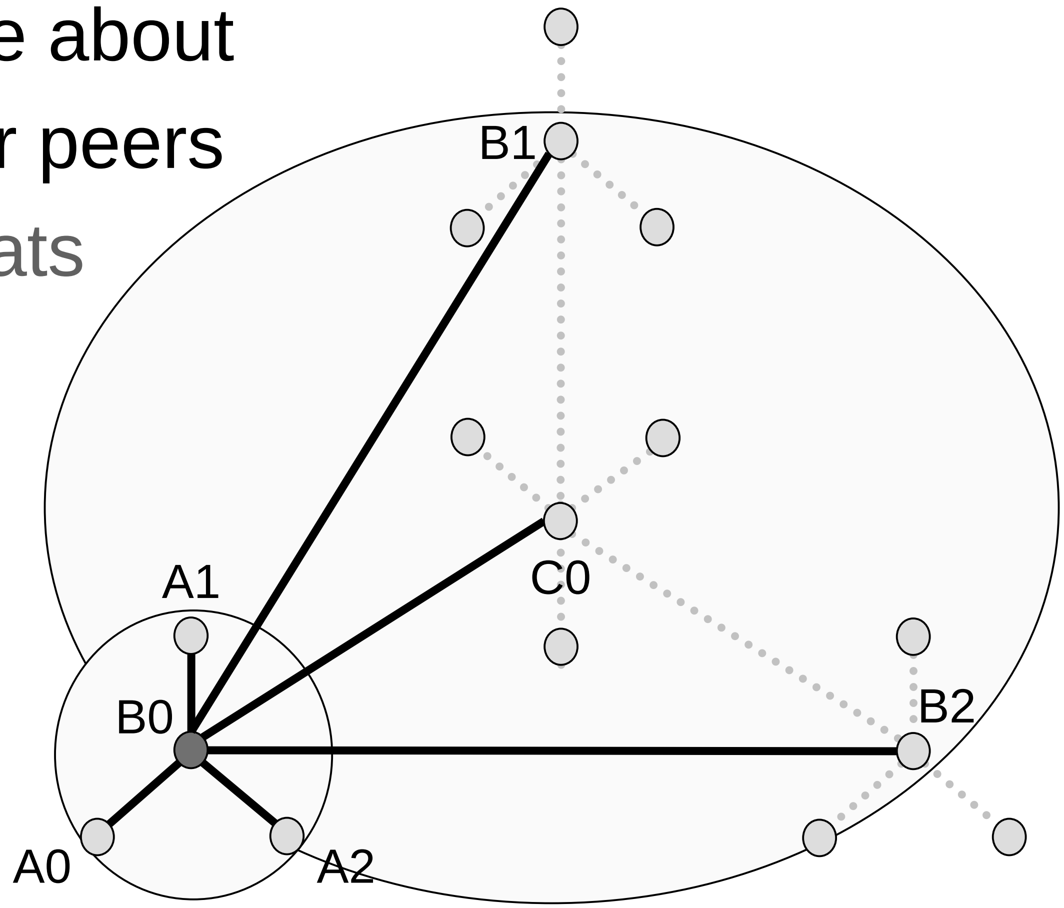
Control Topology

- Soft state about all cluster peers
- HeartBeats



Control Topology

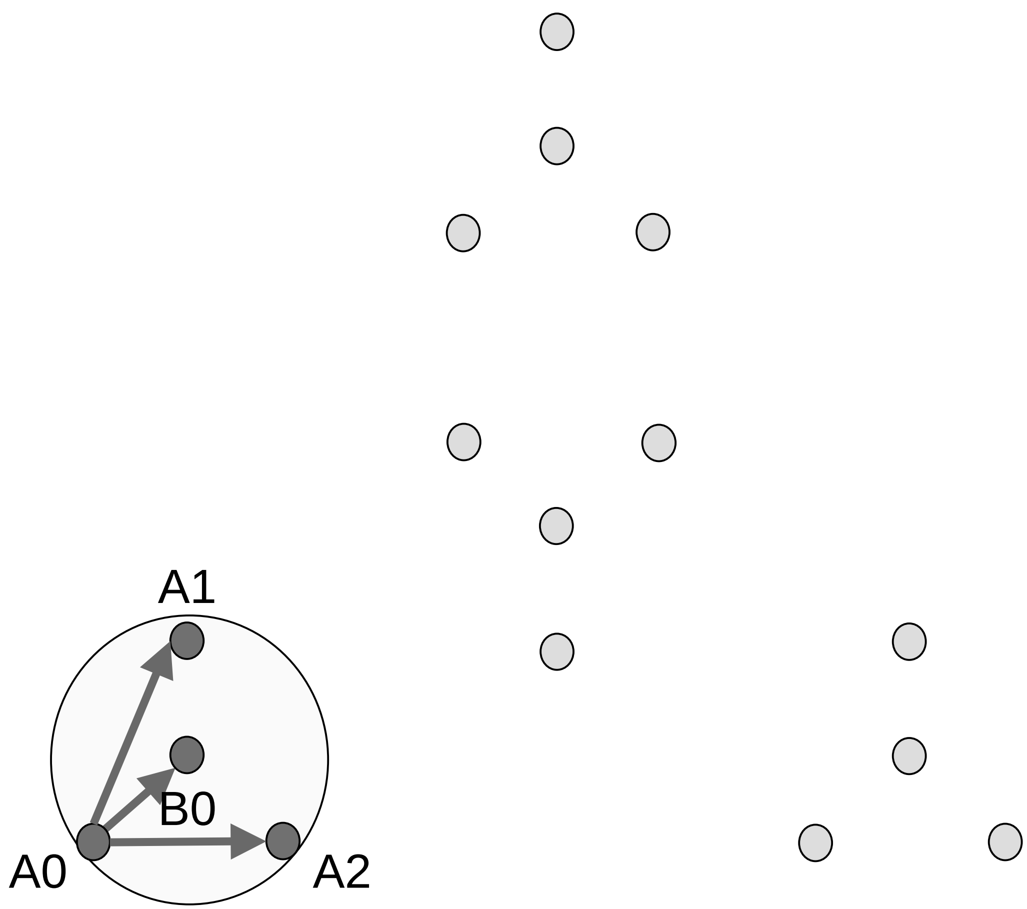
- Soft state about all cluster peers
- HeartBeats



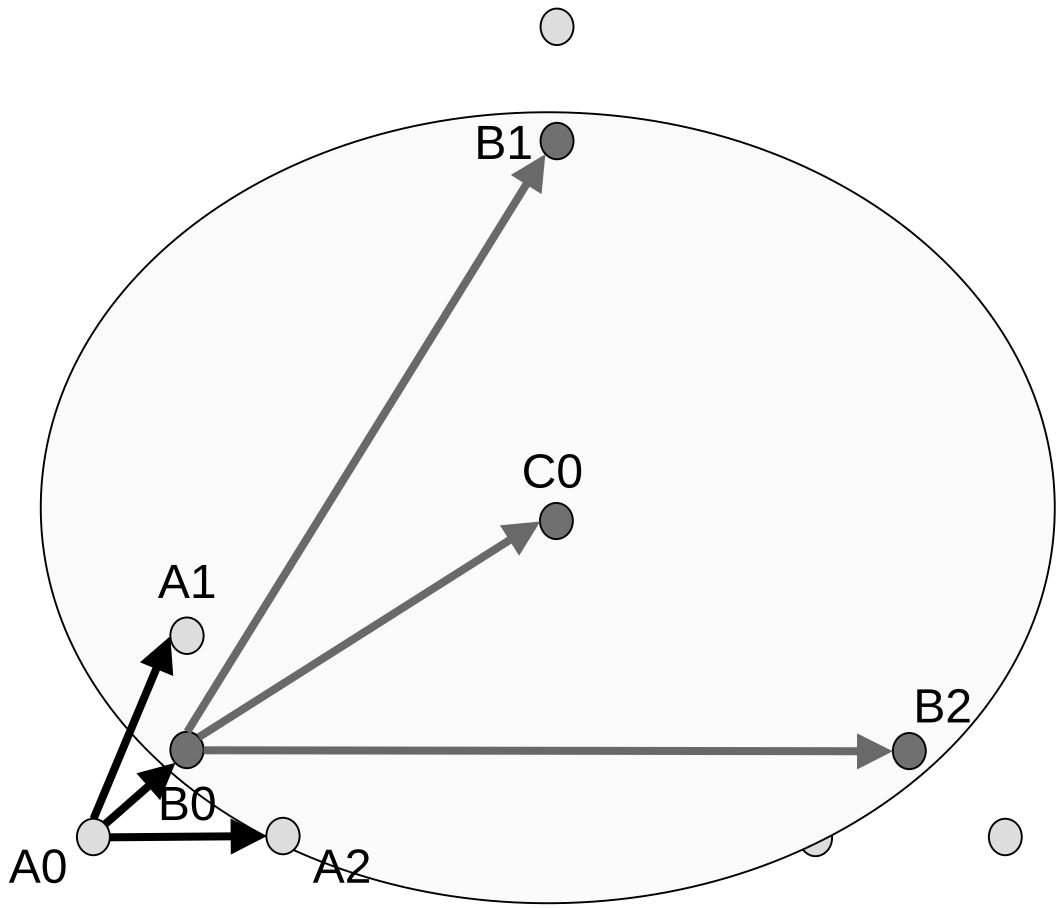
State and Control message overheads:

Average: Constant   Worst case: O(k log N)

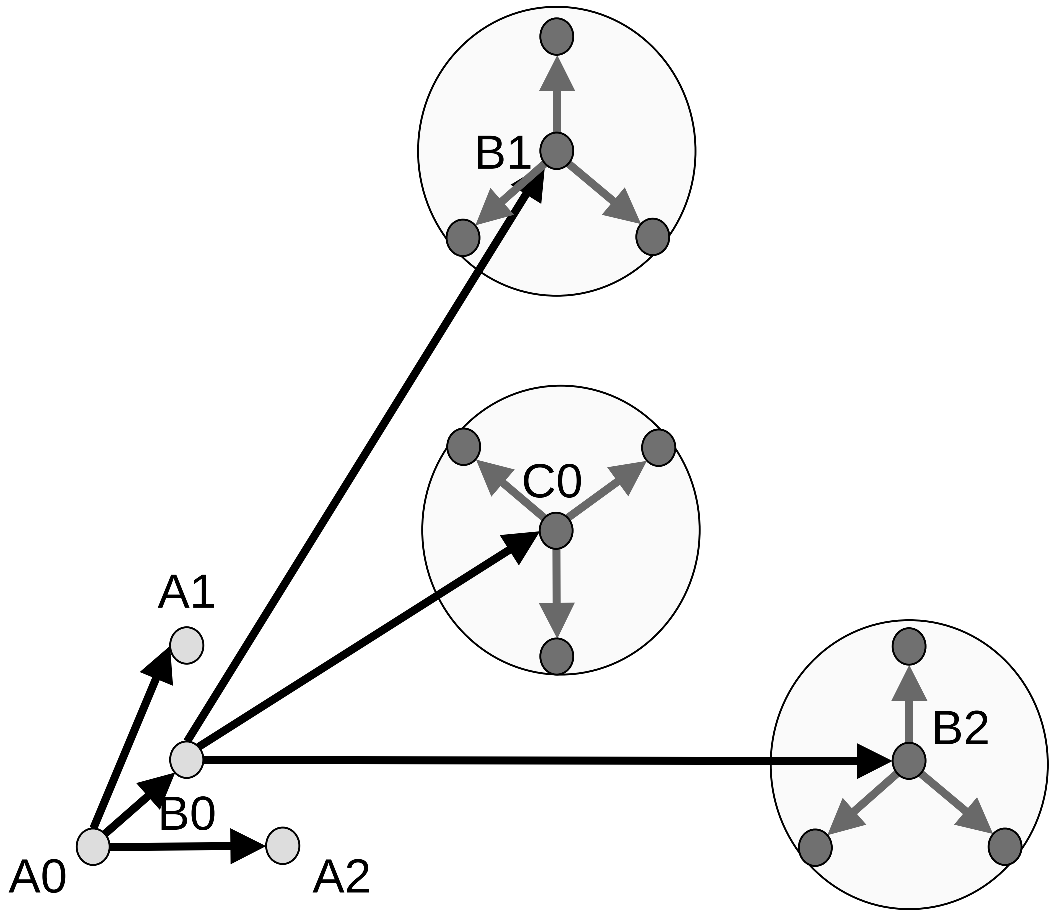
Basic Data Path



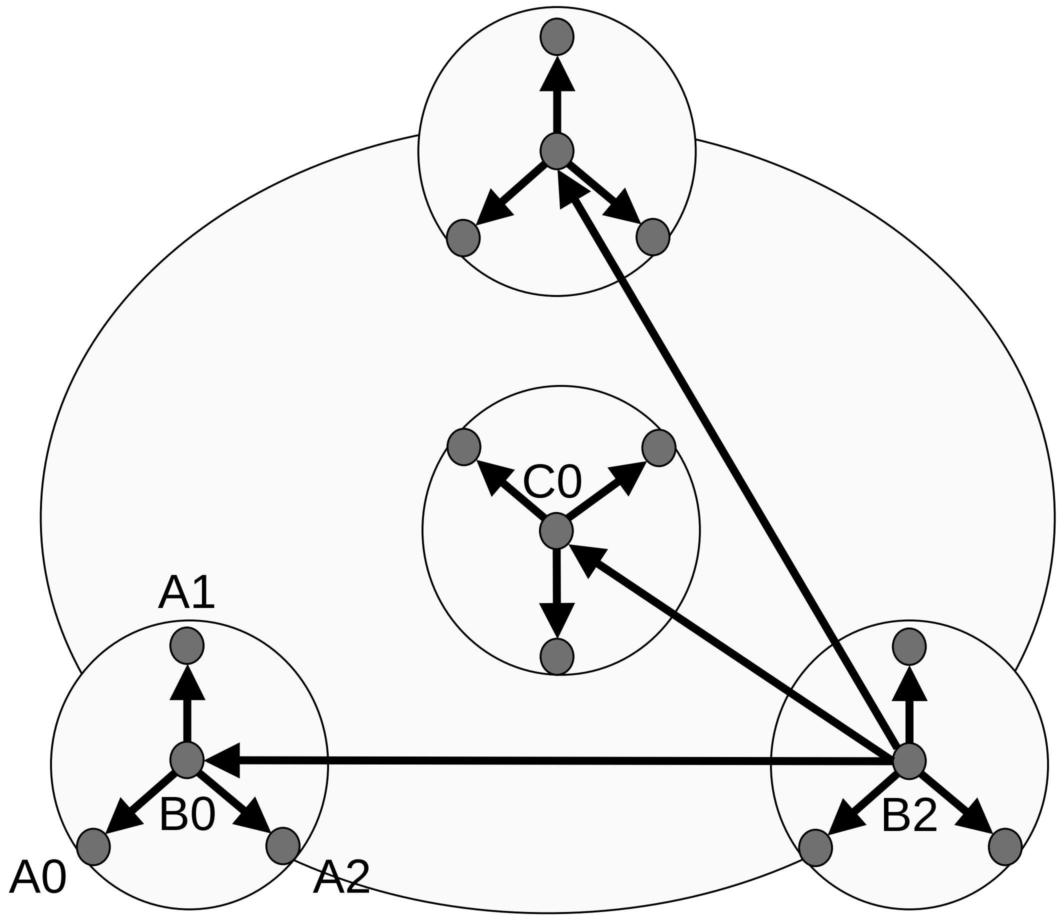
Basic Data Path



Basic Data Path



Basic Data Path



NICE Invariants

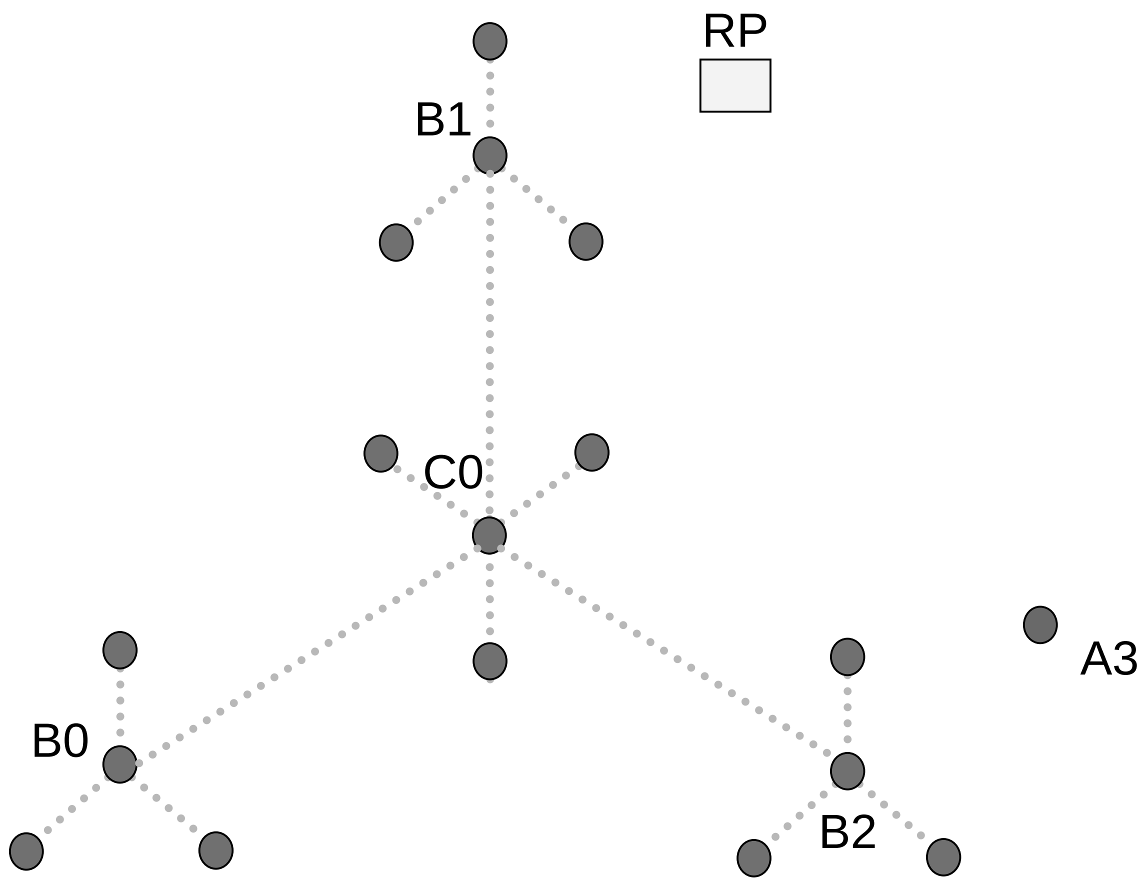
- Cluster sizes between k and 3k-1
- Cluster leader is the central member
  - Leaders for next higher layer

NICE protocol maintains these invariants

NICE Protocol Operations

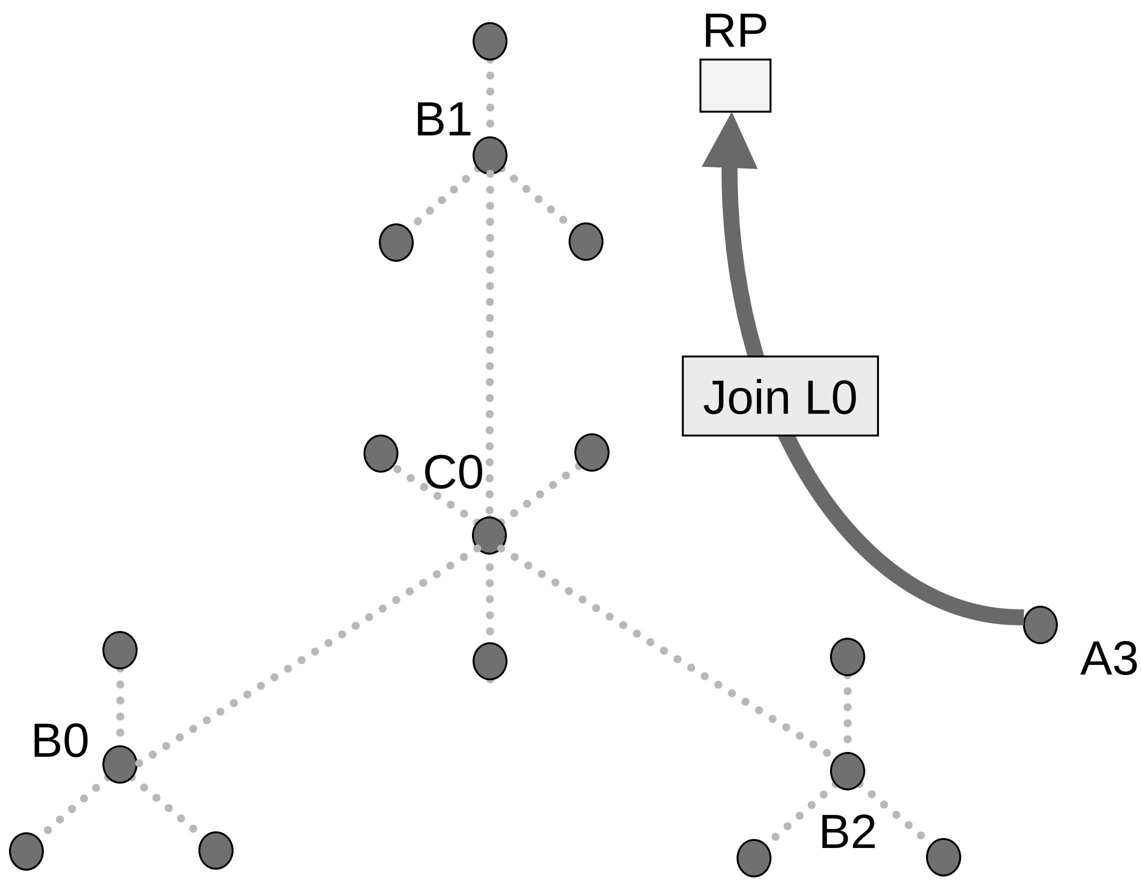
- Member Join
- Member Depart
- Cluster Split
- Cluster Merge
- Cluster Refine

Join Procedure

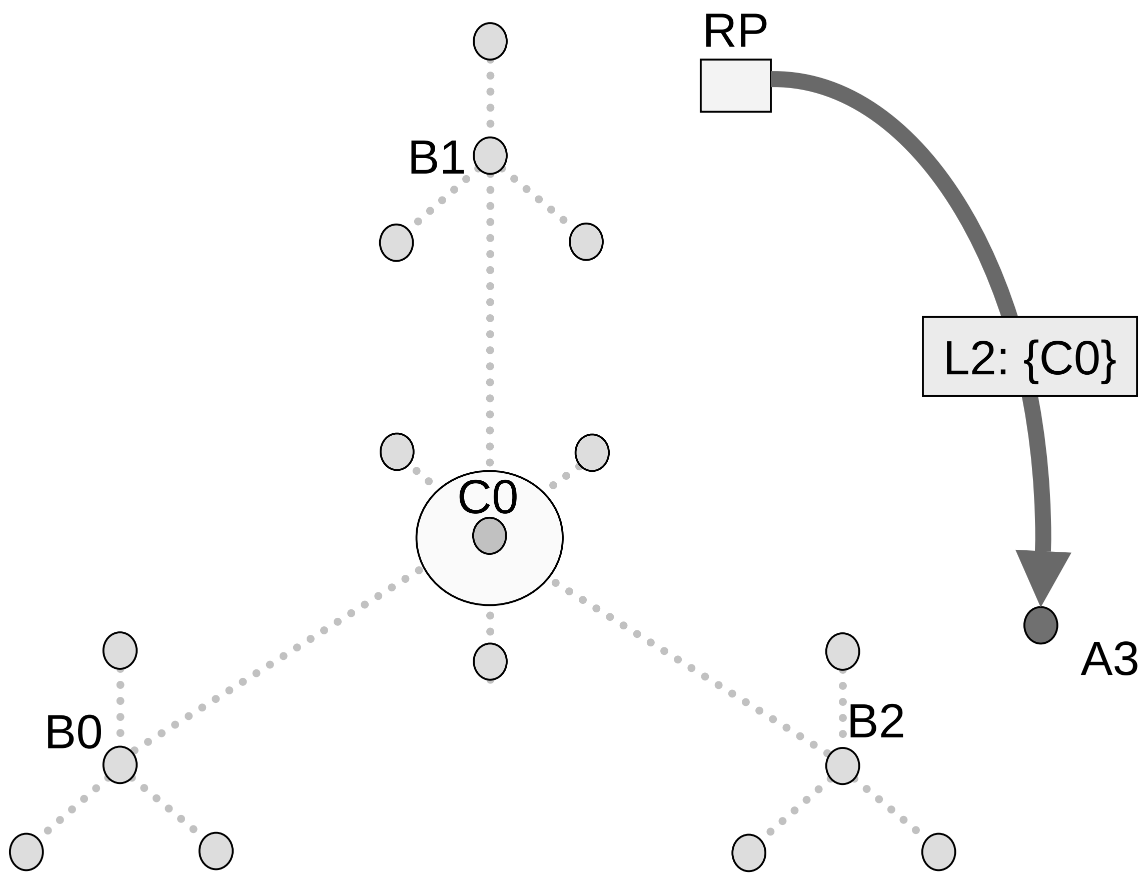


Assume a Rendezvous Point

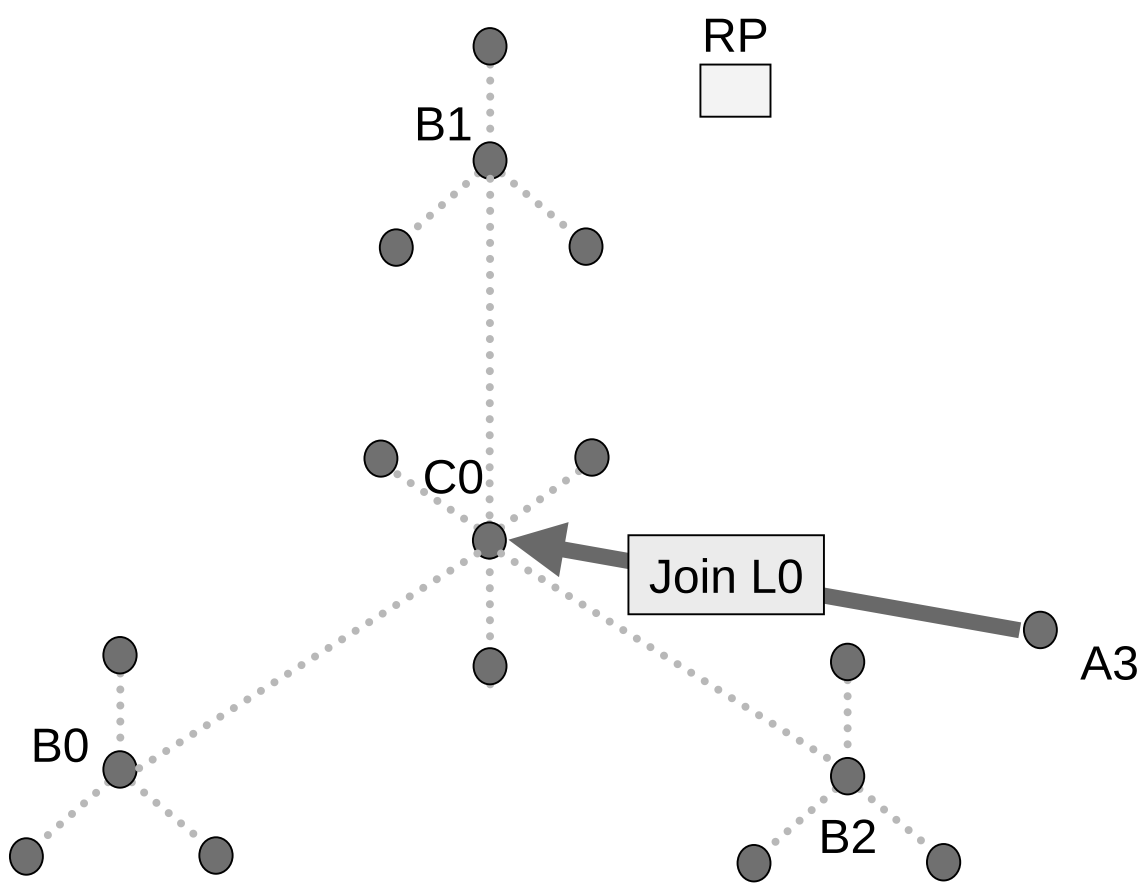
Join Procedure



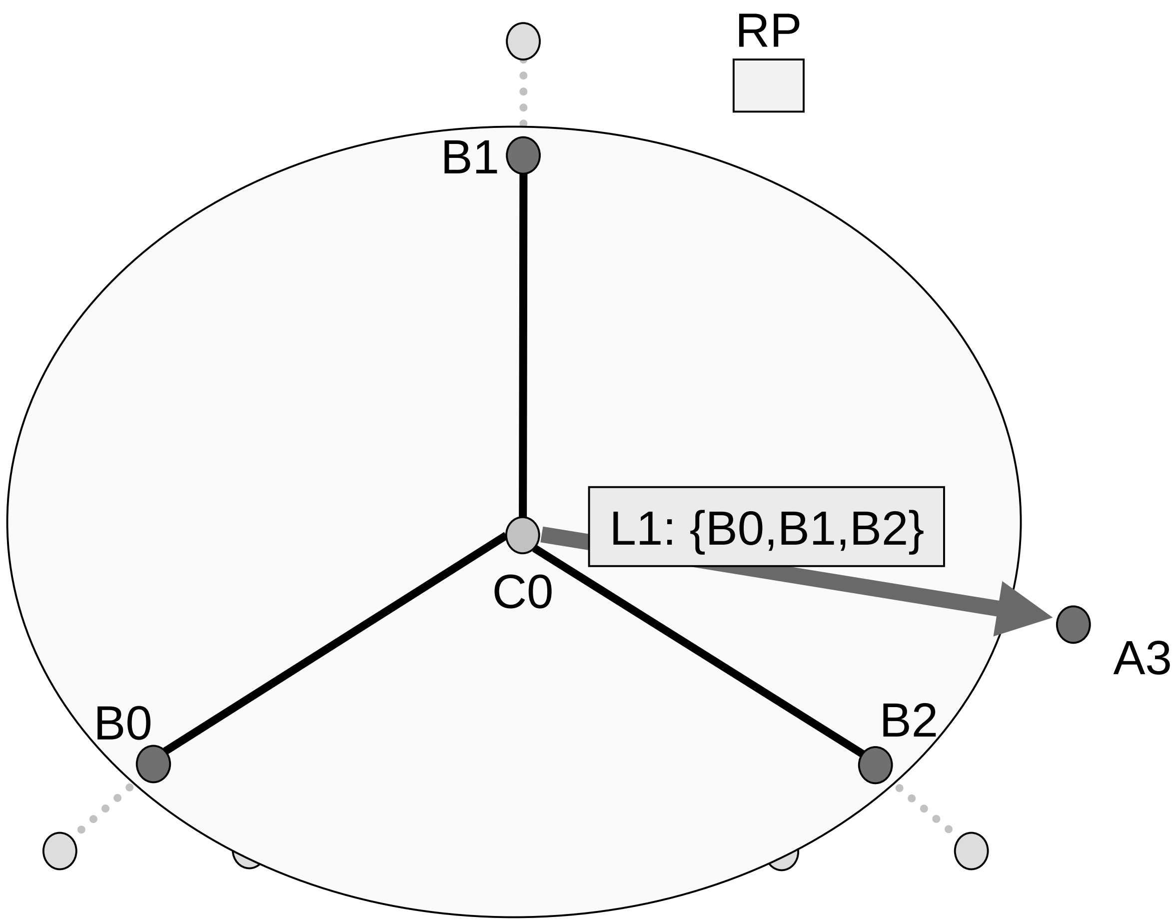
Join Procedure



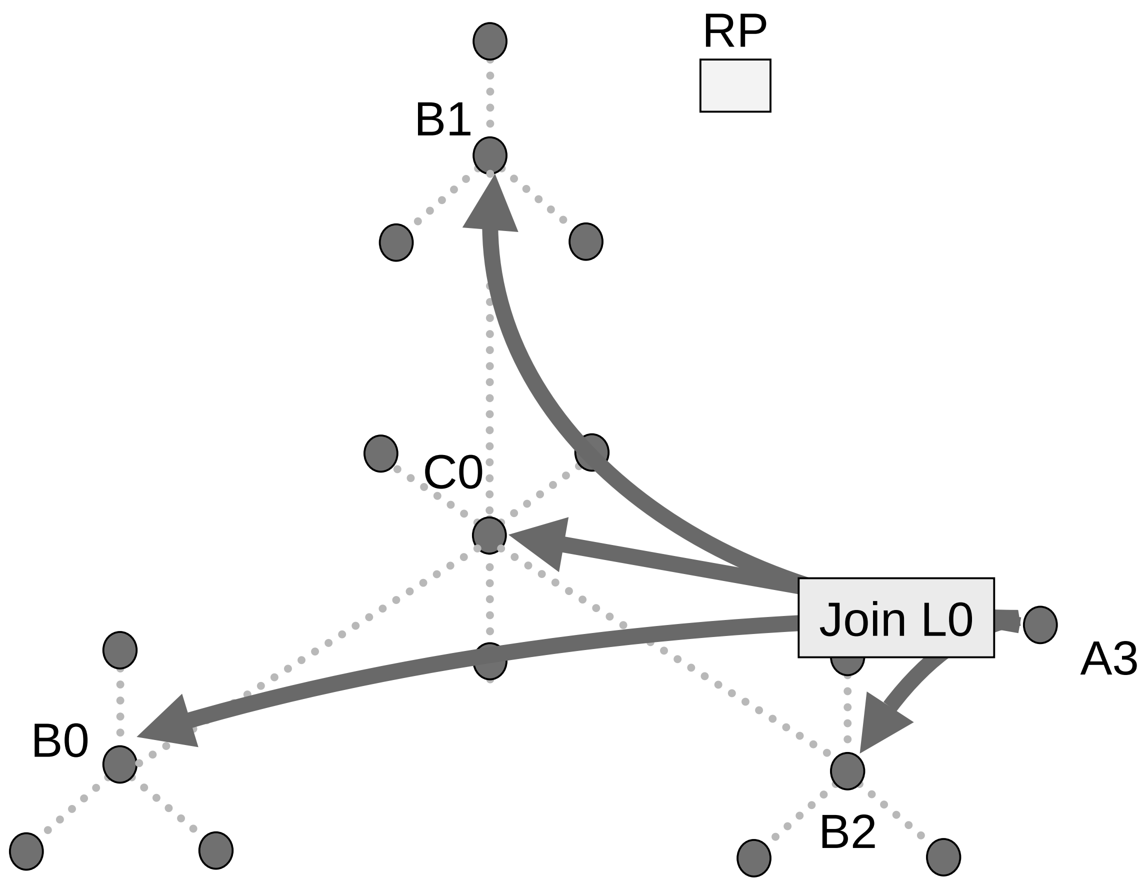
Join Procedure



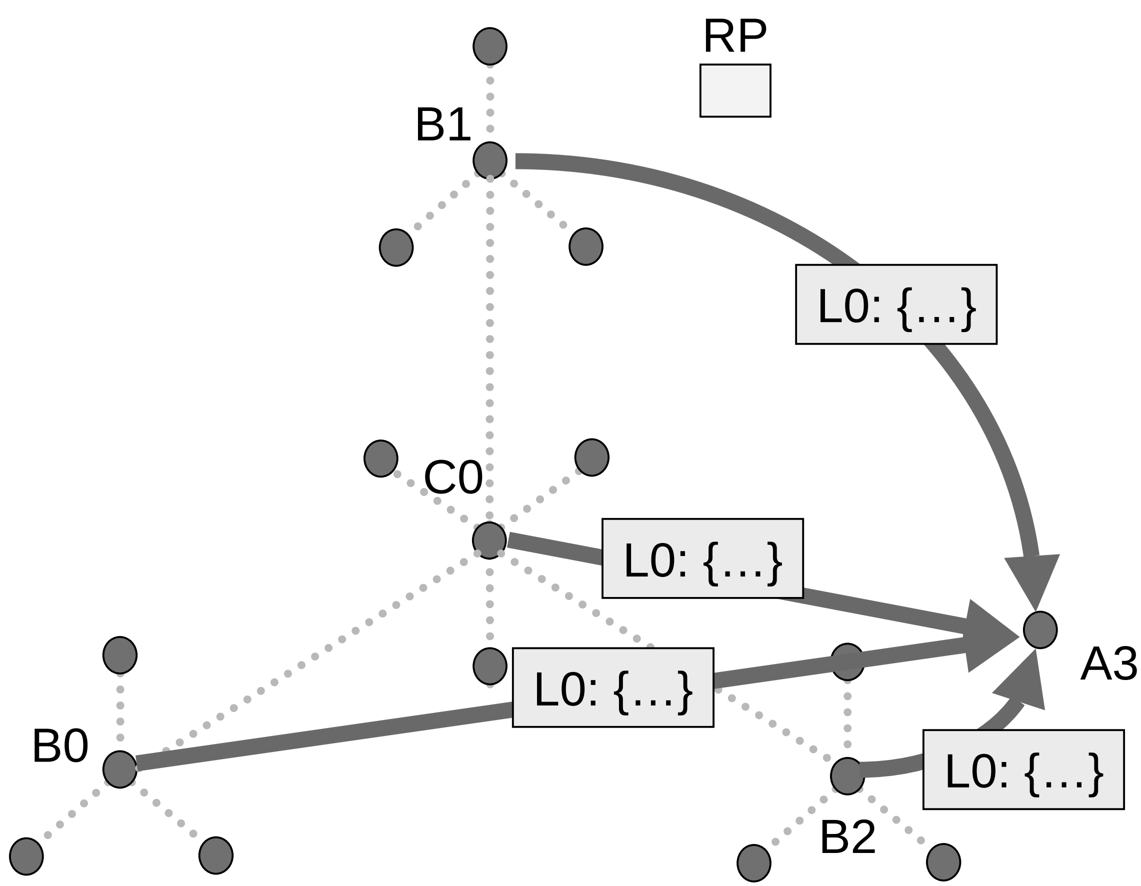
Join Procedure



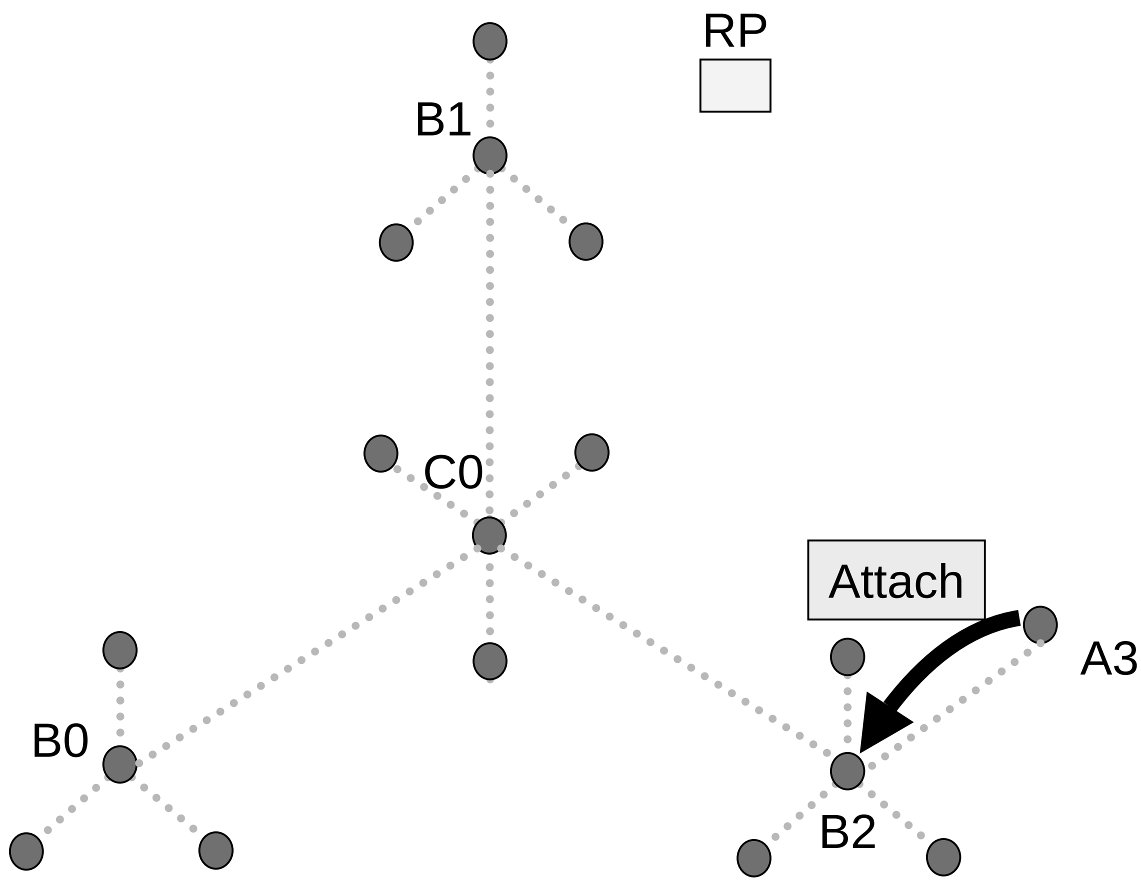
Join Procedure



Join Procedure



Join Procedure

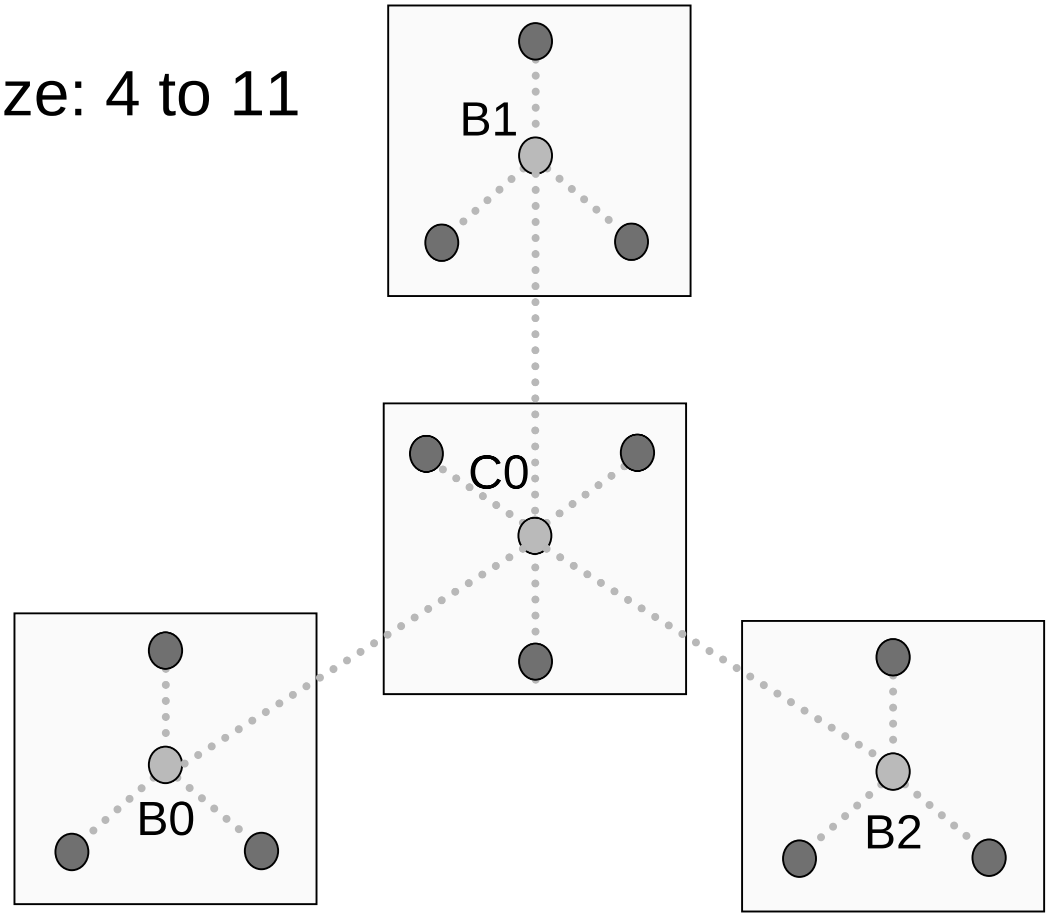


Overhead: O(log N) RTTs and O(k log N) messages

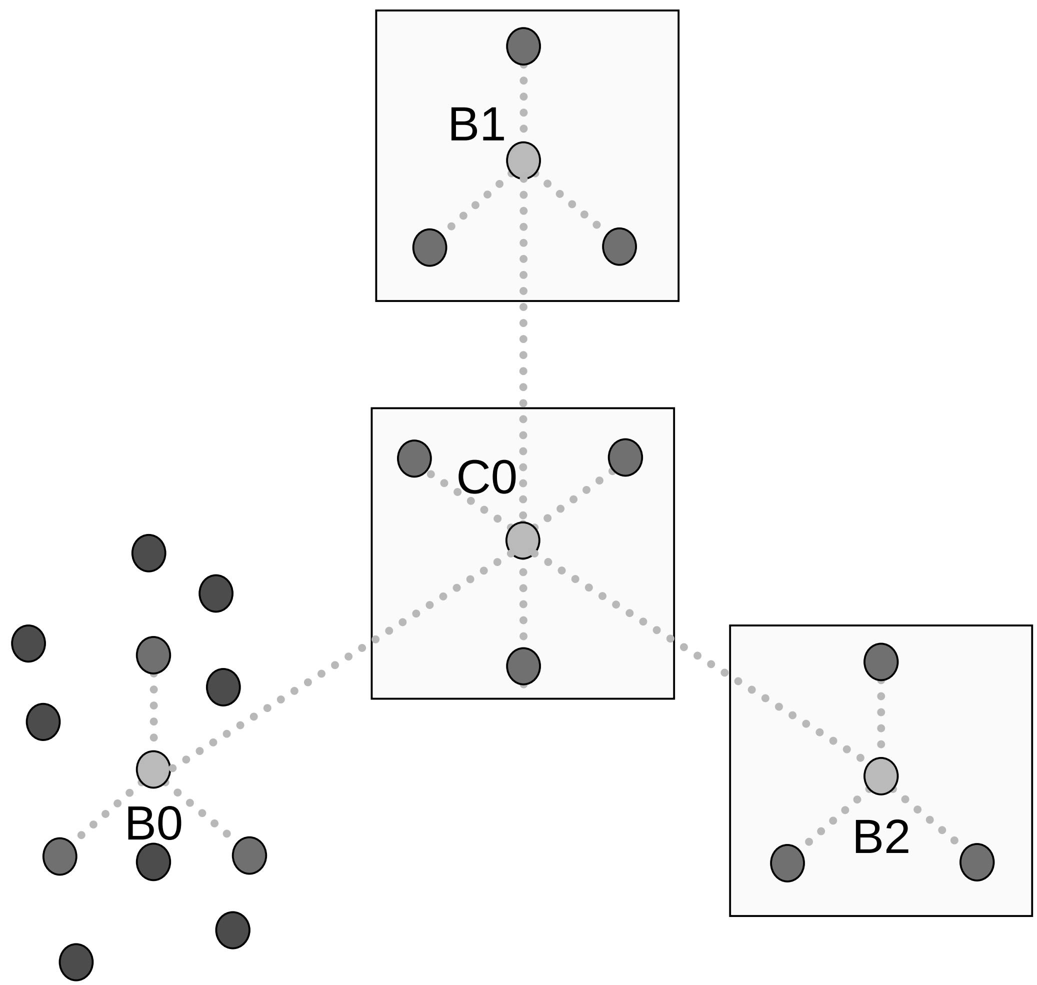
- Optimizations possible

Cluster Split

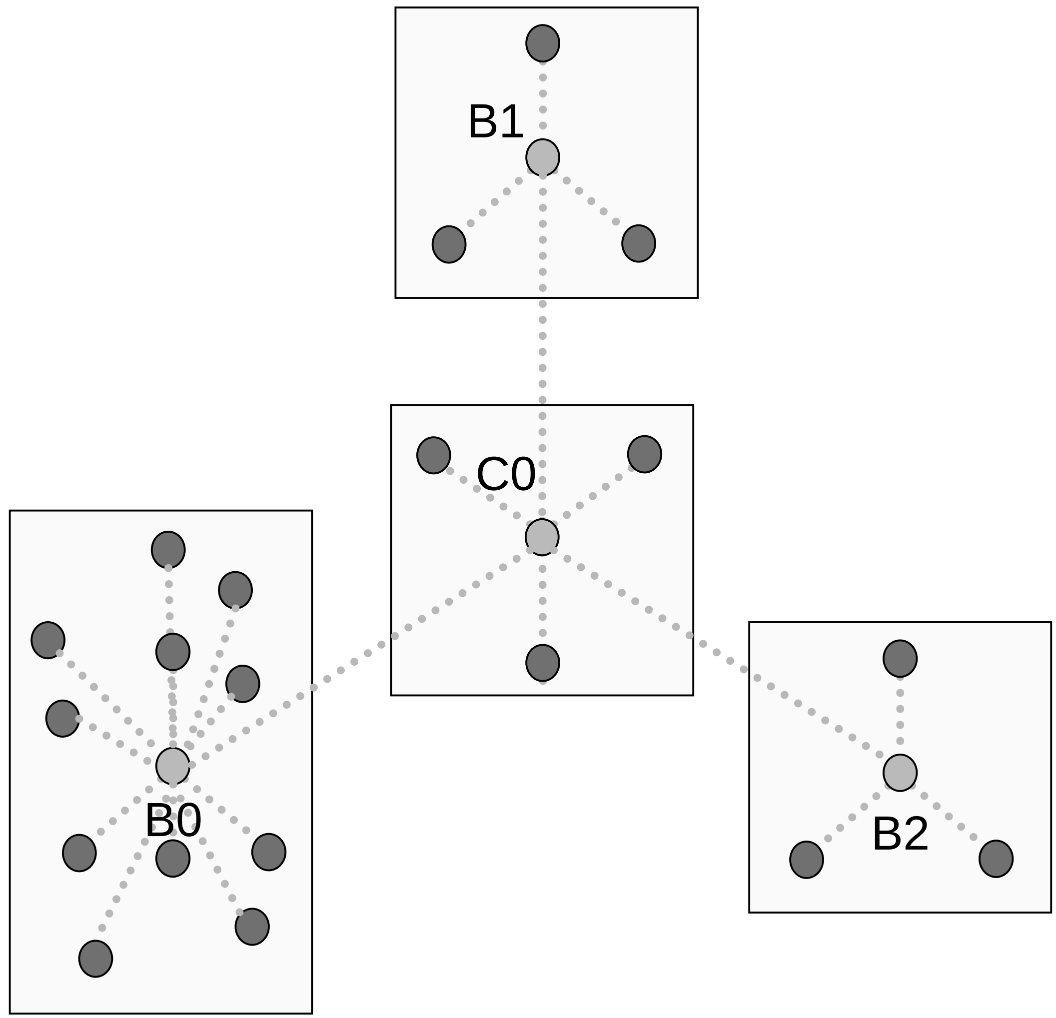
Cluster size: 4 to 11



Cluster Split

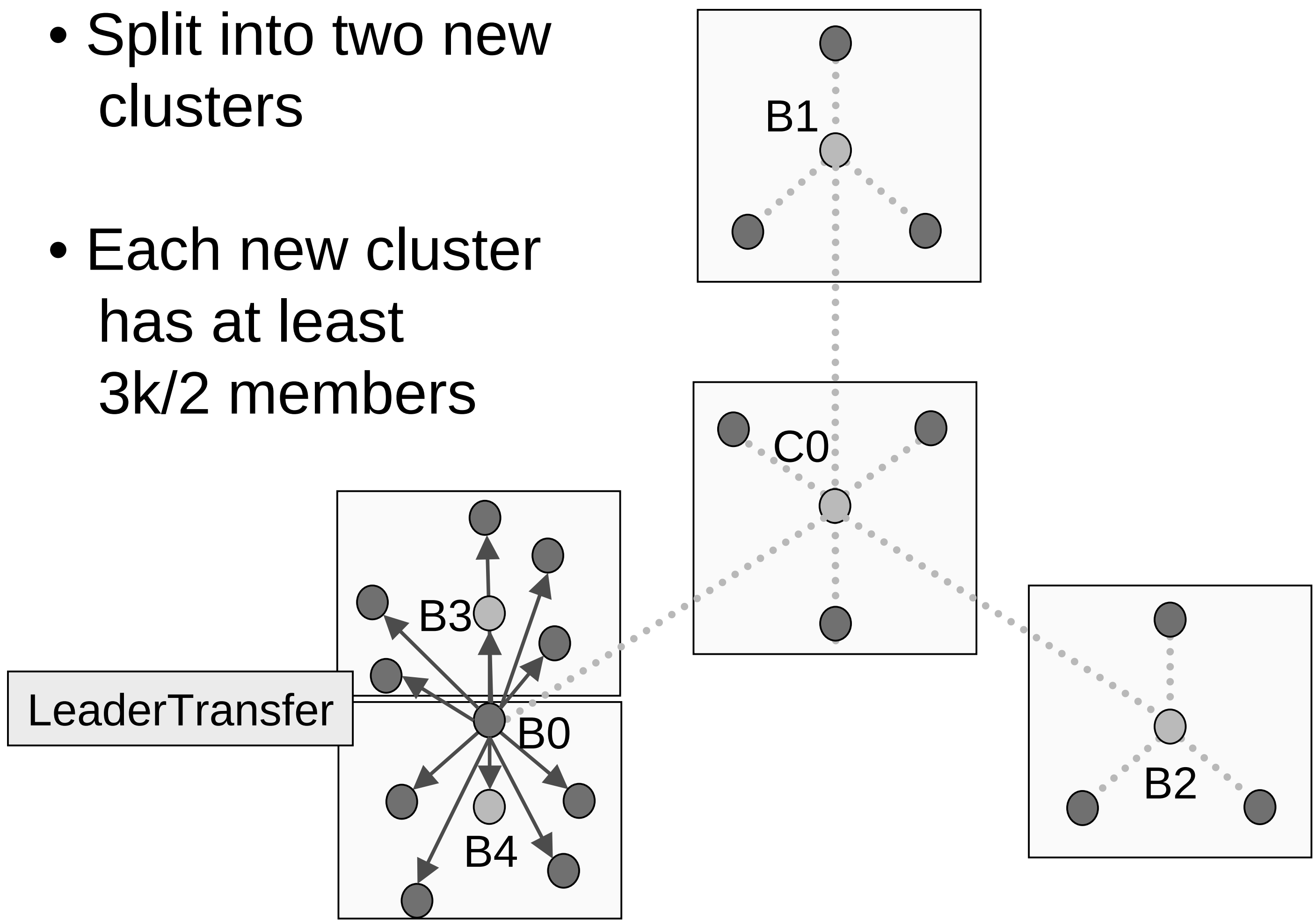


Cluster Split

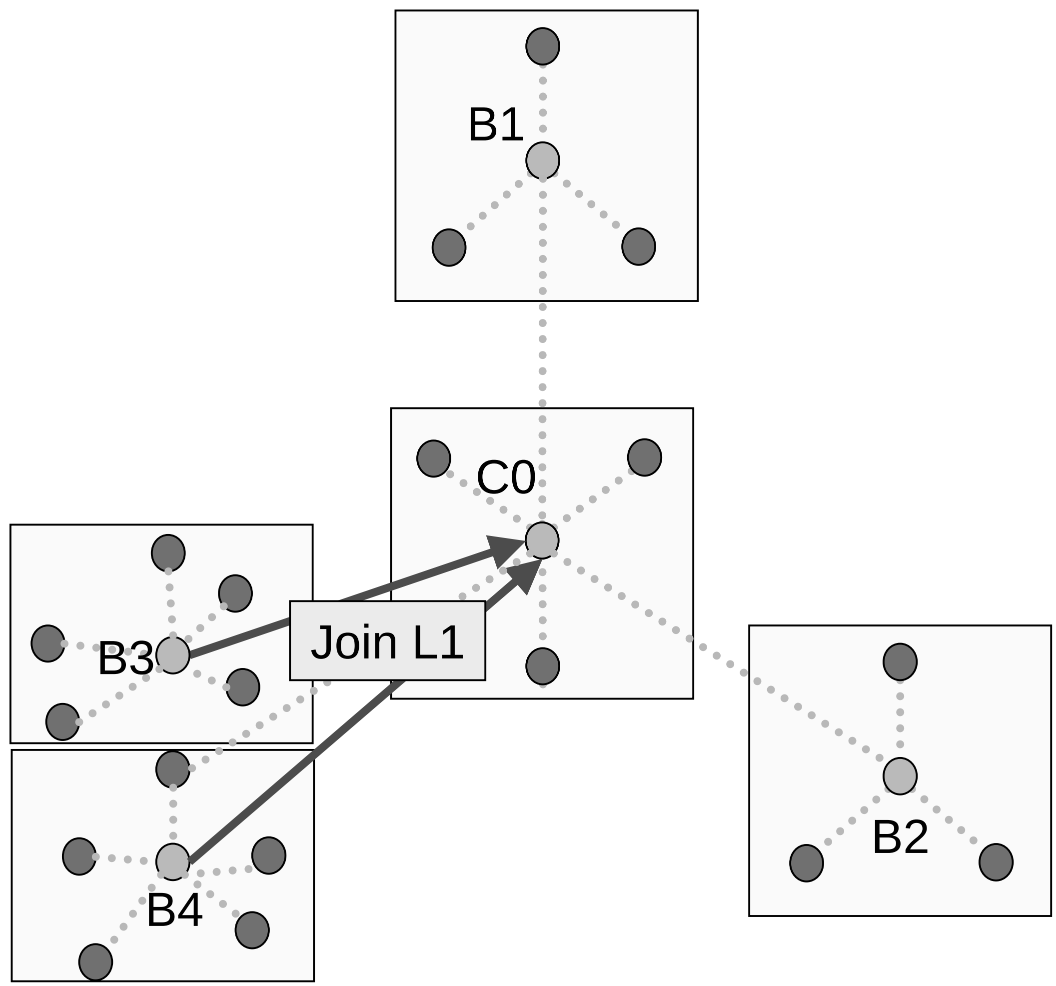


Cluster Split

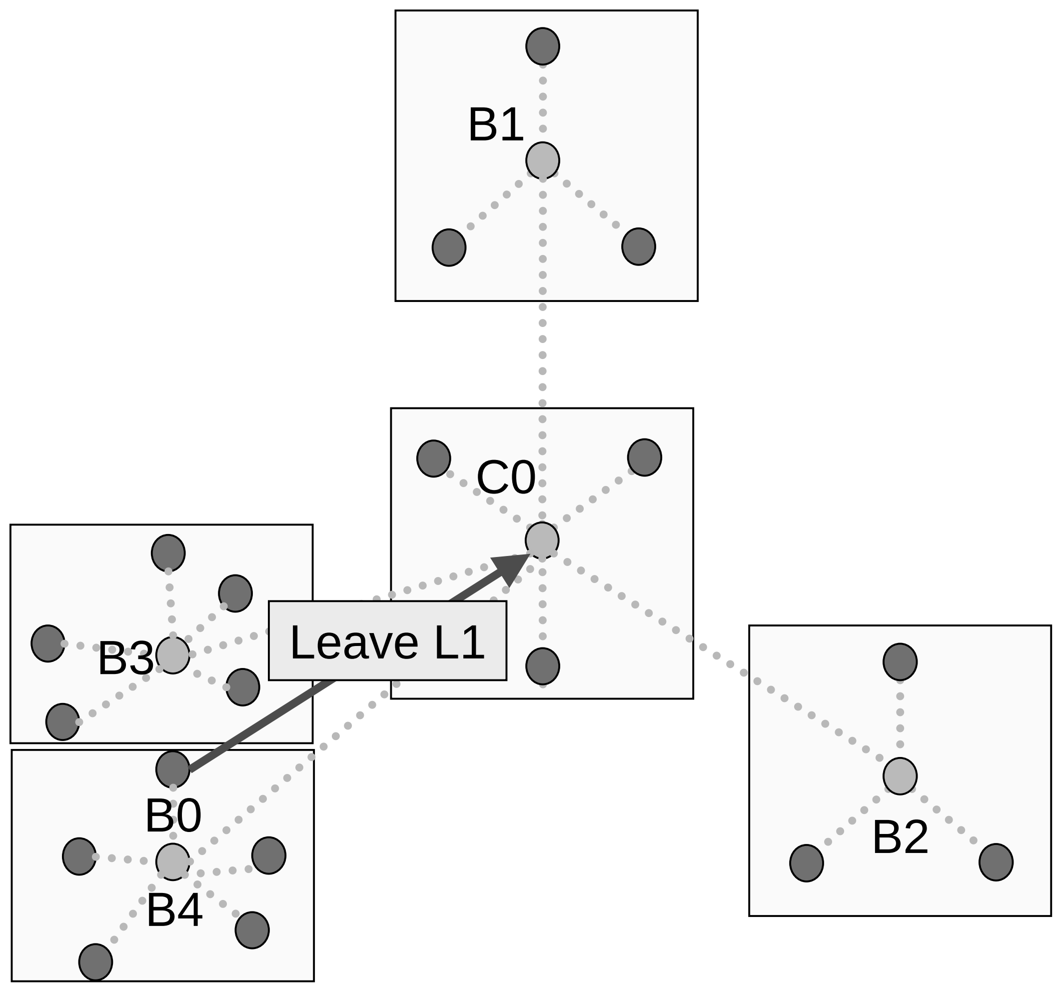
- Split into two new clusters
- Each new cluster has at least 3k/2 members



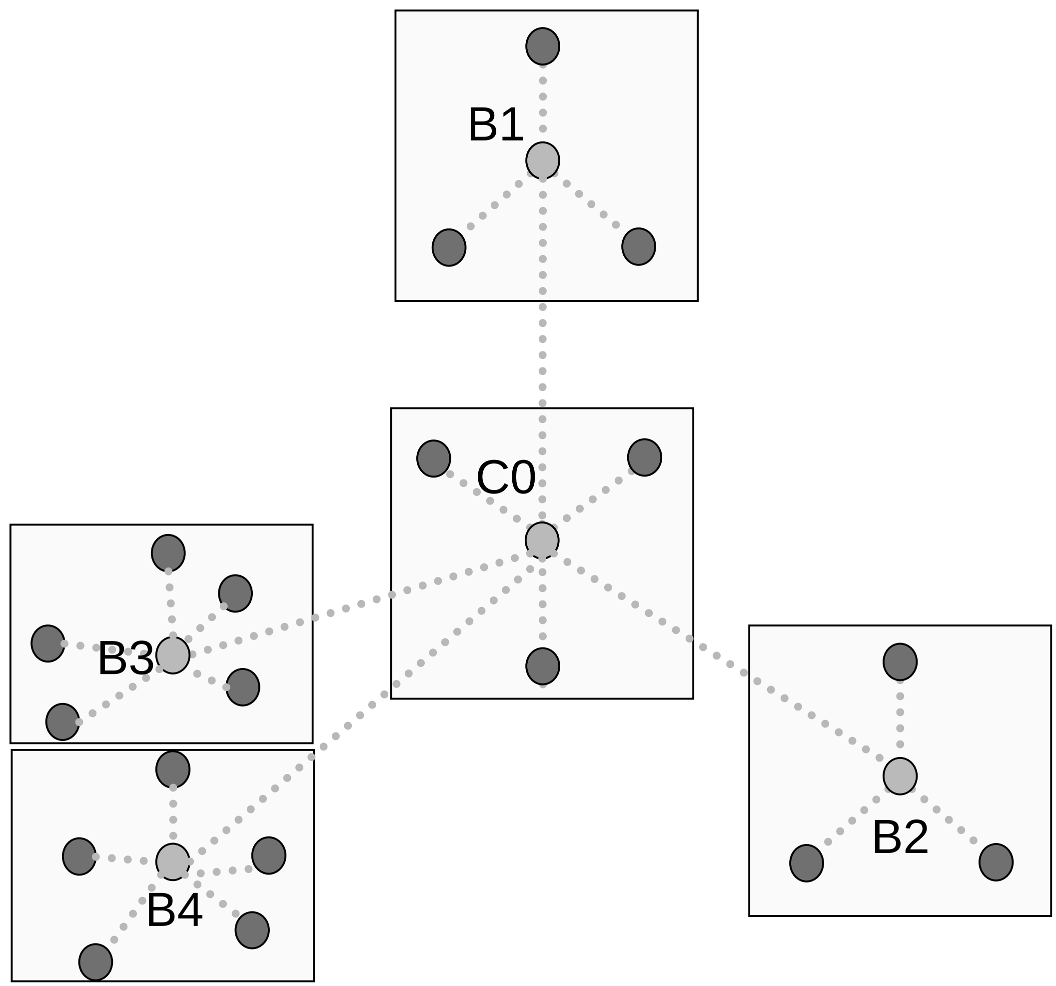
Cluster Split



Cluster Split



Cluster Split



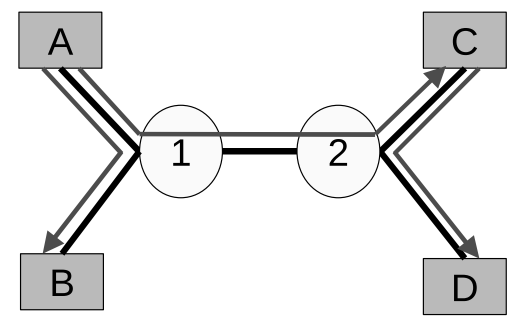
Results

- Simulations
  - 10,000 node Transit-Stub graphs
  - Group sizes upto 2048
  - Comparisons with Narada [CMU]
- Wide-area Experiments
  - Members at 8 sites
  - Group sizes upto 96
- Dynamic joins and (ungraceful) leaves
- Constant rate data source

Evaluation Metrics

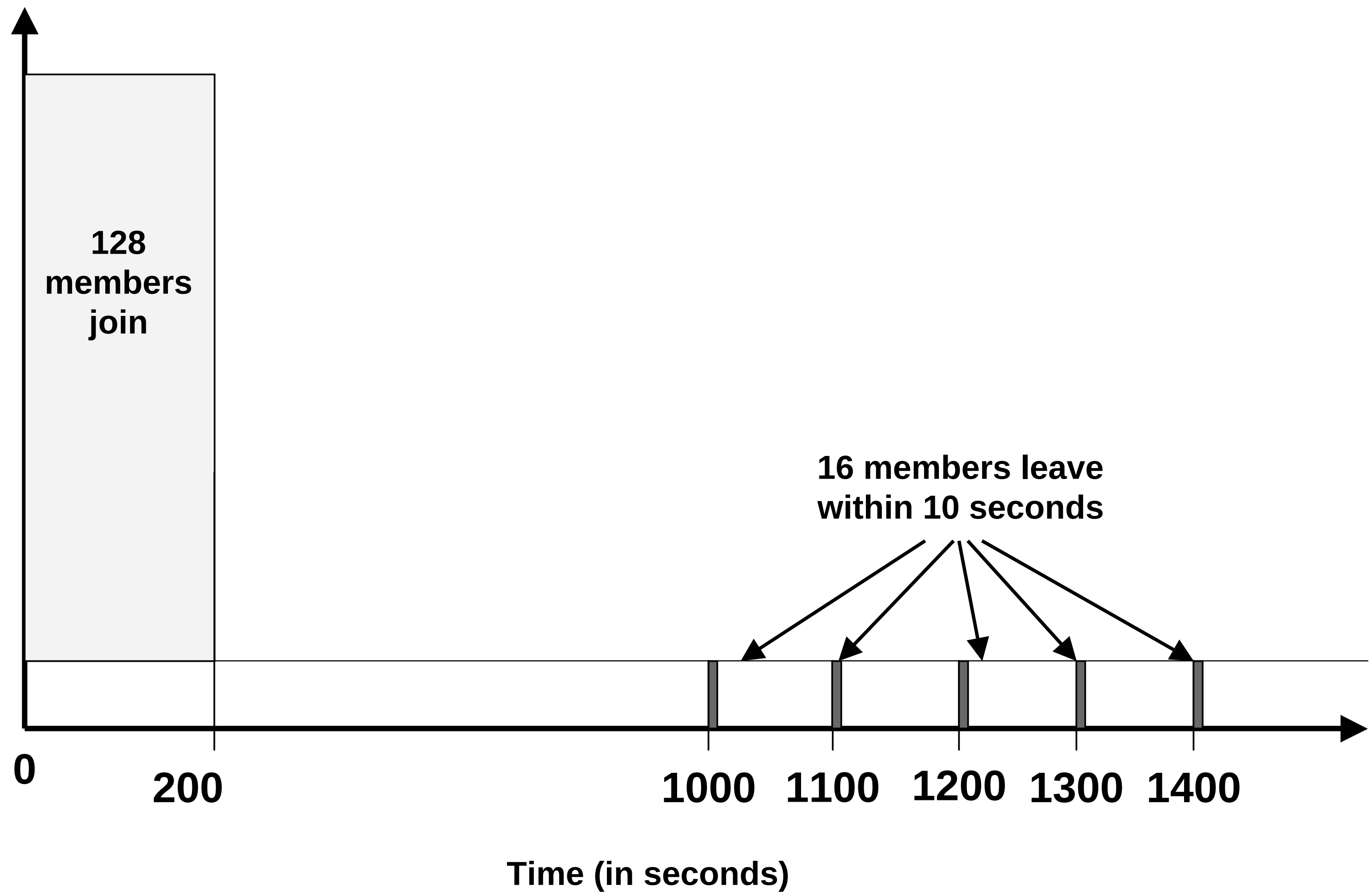
- Tree Quality: Stress
  - Number of copies of the same data packet on a link/router
  - Example: Stress on link [A-1] = 2

Application-layer Multicast



- Tree Quality: Stretch
  - Ratio of the overlay latency to the direct unicast latency
  - Example: Stretch for receiver D = 5/3
- State at end-hosts
  - Control overheads
- Robustness
  - Host failures

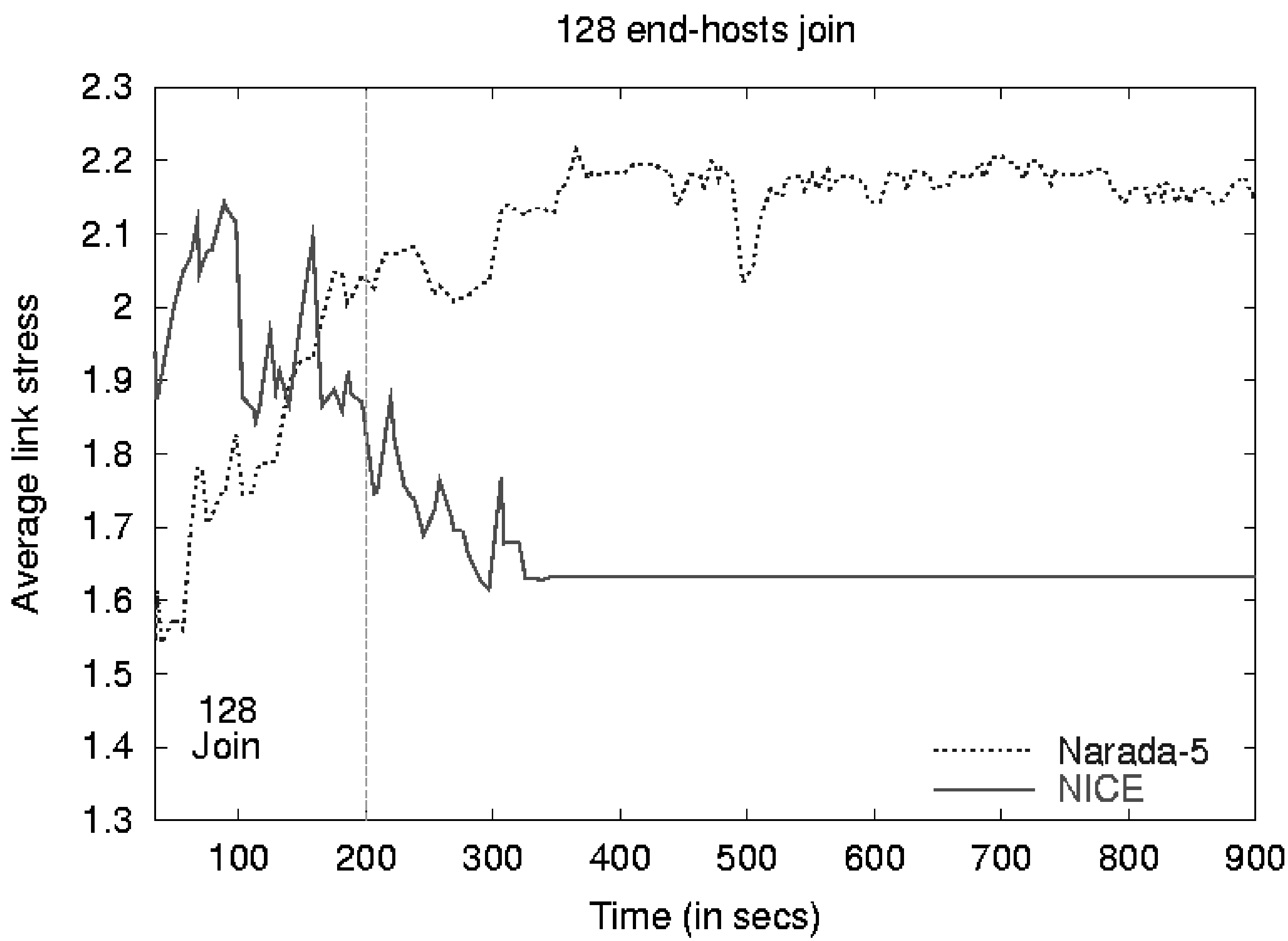
Example Scenario



Tree Quality: Stress

First 200 seconds …

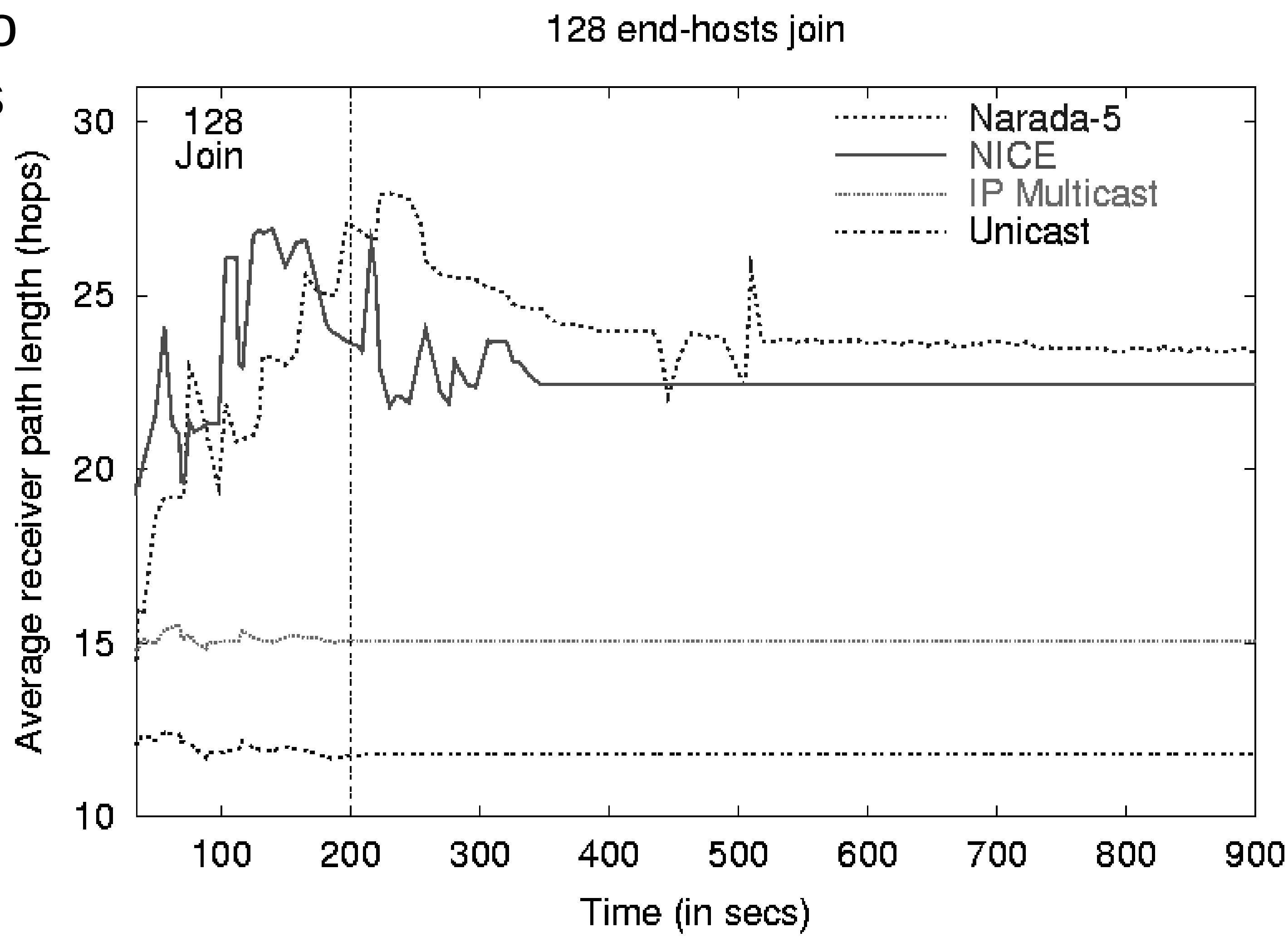
Resource usage at links



Tree Quality: Stretch

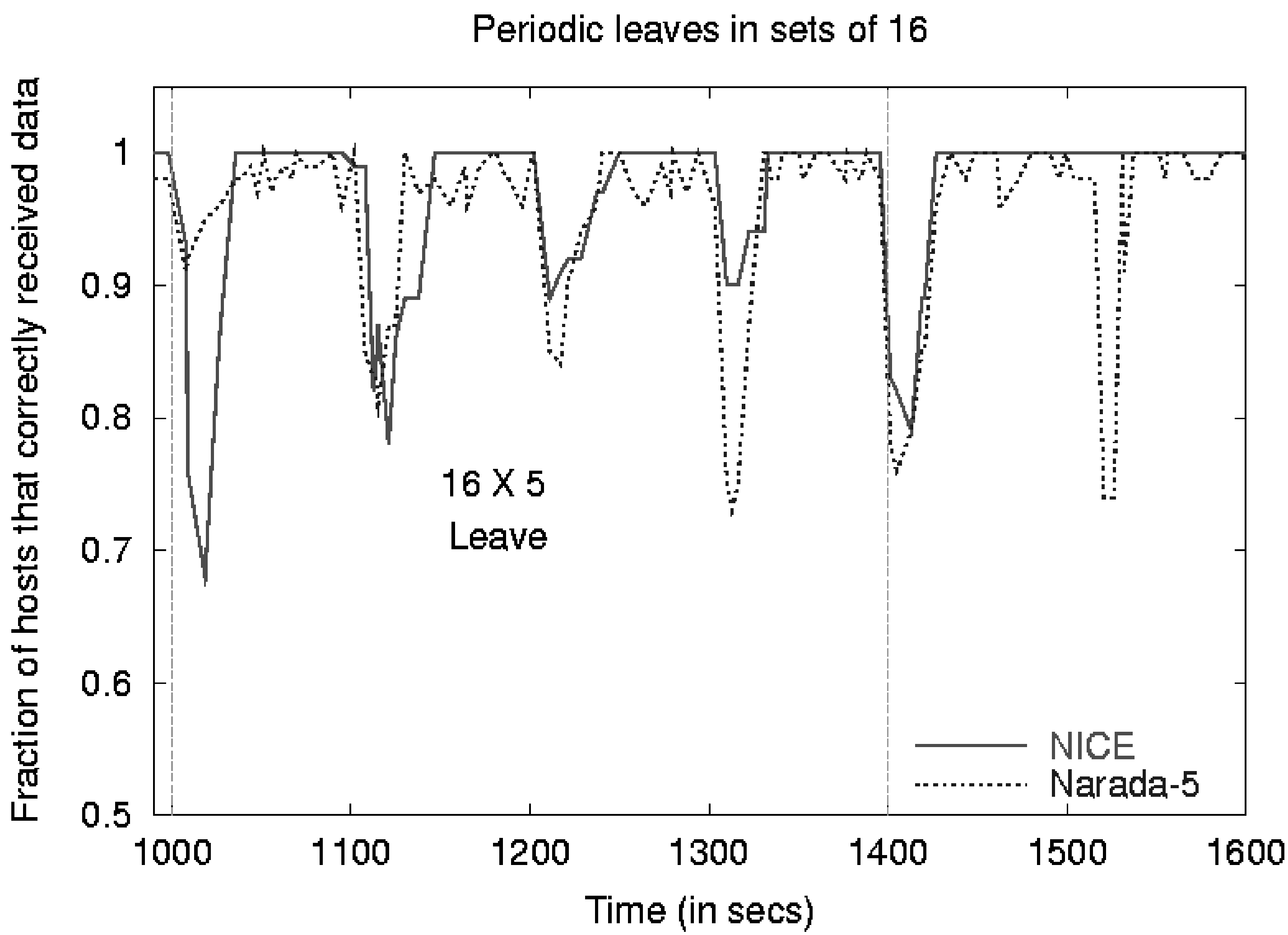
First 200 seconds …

End-to-end latency to receivers

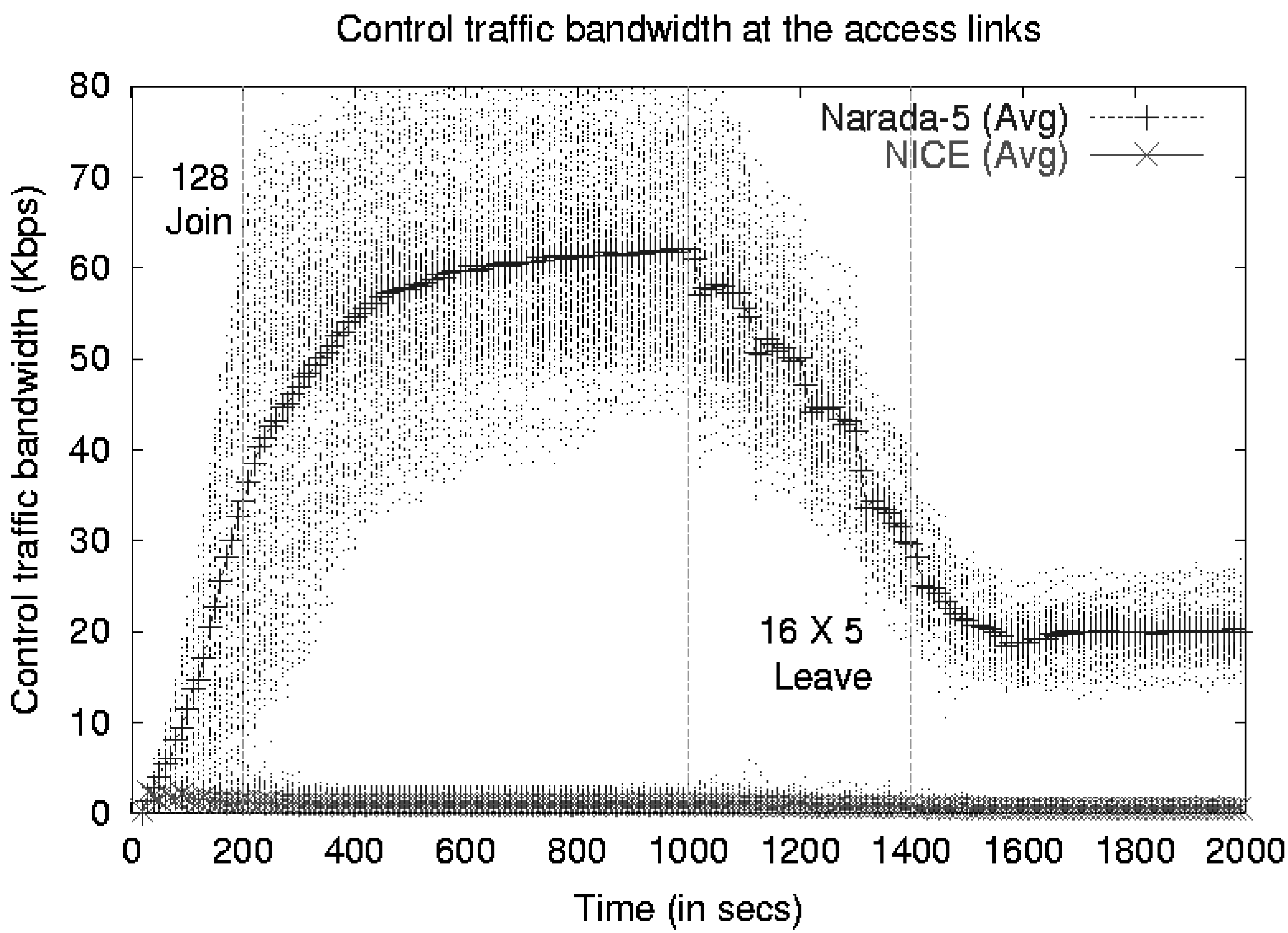


Failure Recovery

After 1000 secs



Control Overheads

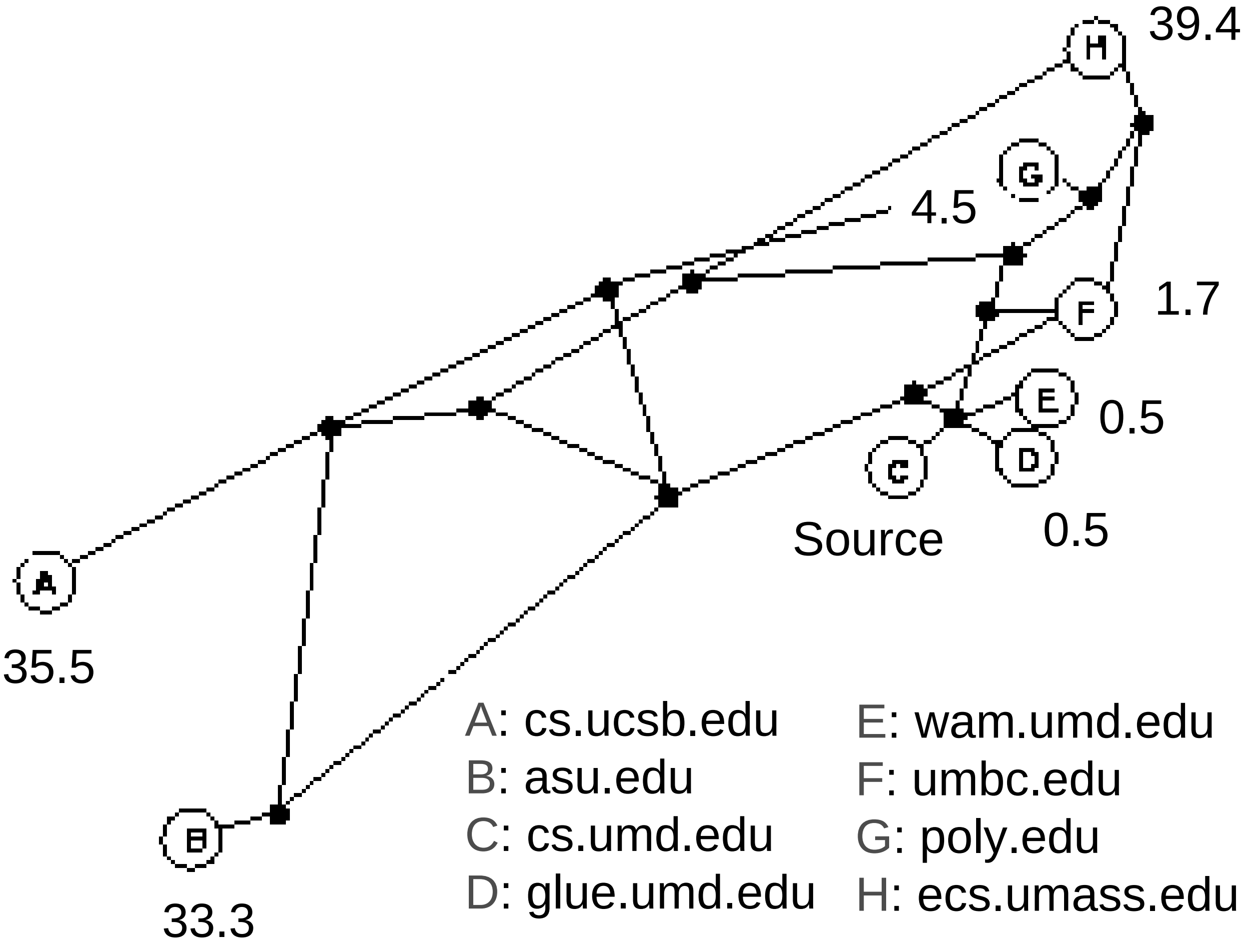


Control Overheads

Group Size	Narada-30	NICE
32	**9.23**	**1.03**
128	**65.62**	**1.19**
512	**199.96**	**1.93**
1024	**-**	**2.81**
1560	**-**	**3.28**
2048	**-**	**5.18**

Bandwidth overheads averaged over all network routers

Wide-area Testbed

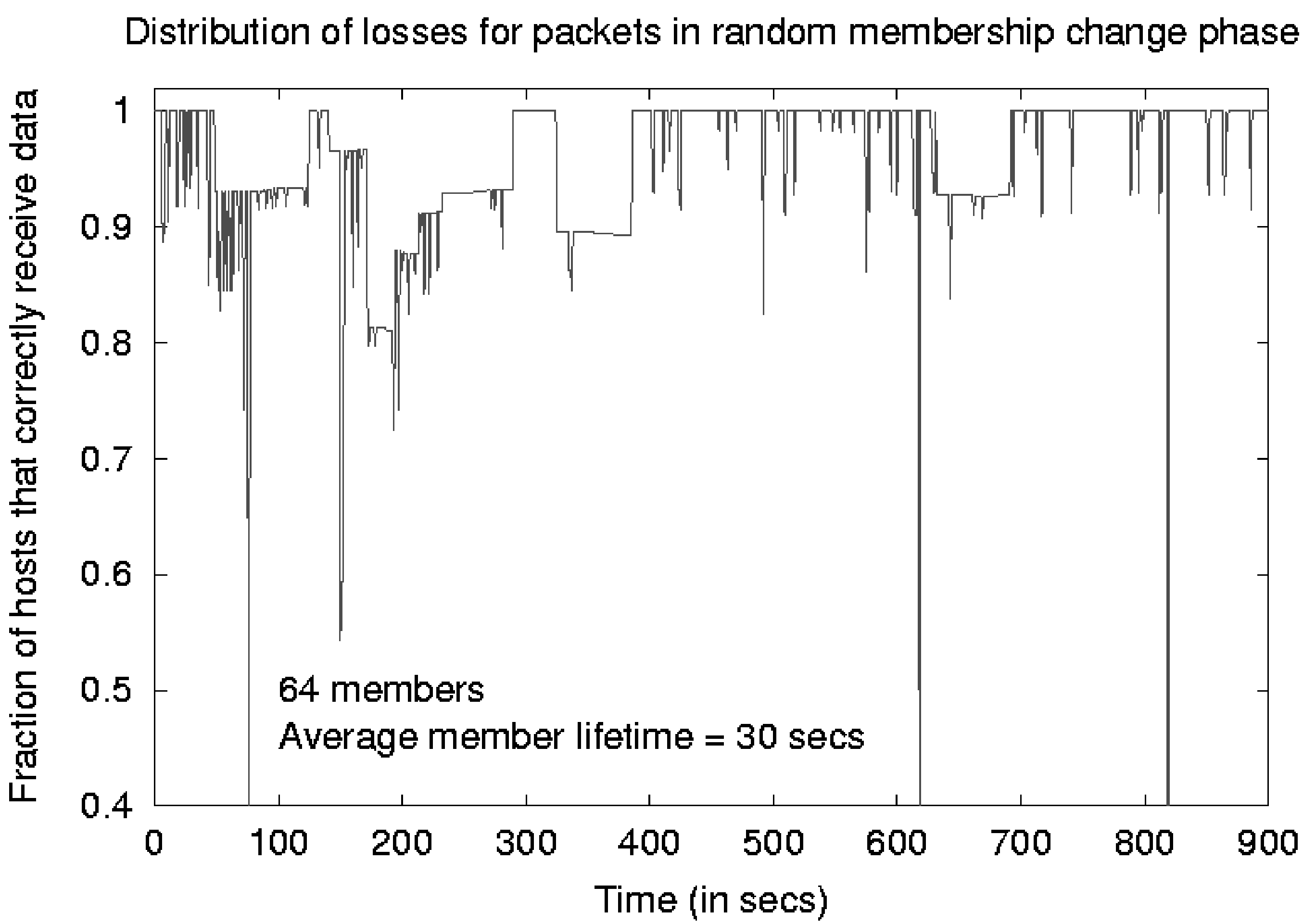


A: cs.ucsb.edu
B: asu.edu
C: cs.umd.edu
D: glue.umd.edu

E: wam.umd.edu
F: umbc.edu
G: poly.edu
H: ecs.umass.edu

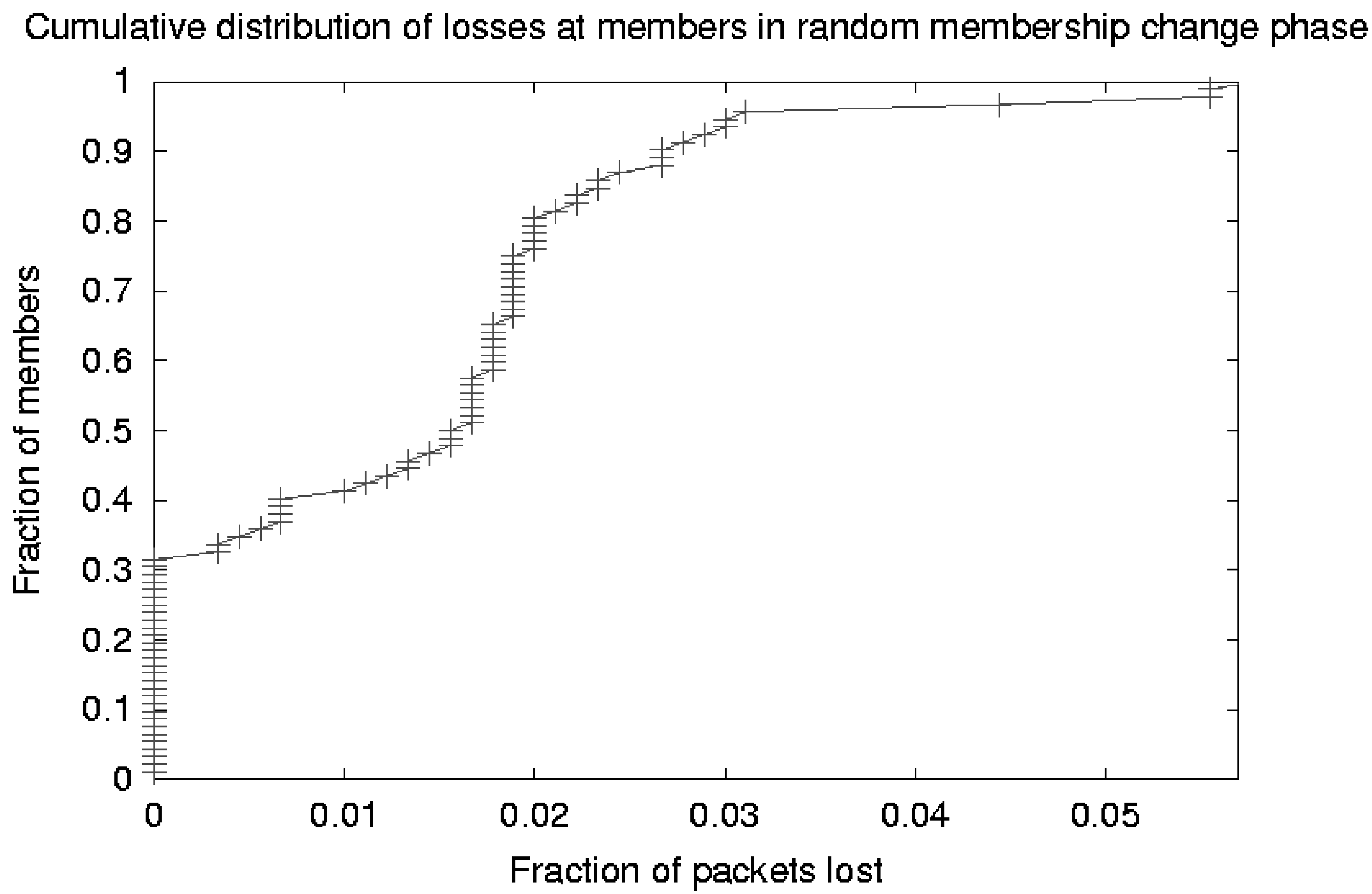
Failure Recovery

Includes the effects of network losses



Failure Recovery

Includes the effects of network losses



Related Work

- Mesh-first
  - Narada, Gossamer
- Tree-first
  - Yoid, HMTP
- Implicit
  - Scribe, Bayeux, CAN-multicast, Delaunay-Triangulation

"A Comparative Study of Application Layer Multicast Protocols",
S. Banerjee and B. Bhattacharjee
- Available at: http://www.cs.umd.edu/~suman/publications.html

Current Work

- Detailed analysis of tree quality
  - Stress and stretch
- Implementing applications
  - Video delivery

Conclusions

- NICE scales to large member groups
  - Low control overhead
  - Does not sacrifice tree quality or robustness

- Scalability using hierarchy

http://www.cs.umd.edu/projects/nice