CS 552
Multicast

Richard Martin

(slide credits B. Nath, I. Stoica)

Motivation

- Many applications requires one-to-many communication
  - E.g., video/audio conferencing, news dissemination, file updates, etc.
- Using unicast to replicate packets not efficient
  → thus, IP multicast needed
  - What about the end-to-end arguments?

Semantic

- Open group semantic
  - A group is identified by a location-independent address
  - Any source (not necessary in the group) can multicast to all members in a group
- Advantages:
  - Query an object/service when its location is not known
- Disadvantage
  - Difficult to protect against unauthorized listeners

Multicast Service Model

- Built around the notion of group of hosts:
  - Senders and receivers need not know about each other
- Sender simply sends packets to “logical” group address
- No restriction on number or location of receivers
  - Applications may impose limits
- Normal, best-effort delivery semantics of IP

Multicast Service Model (cont'd)

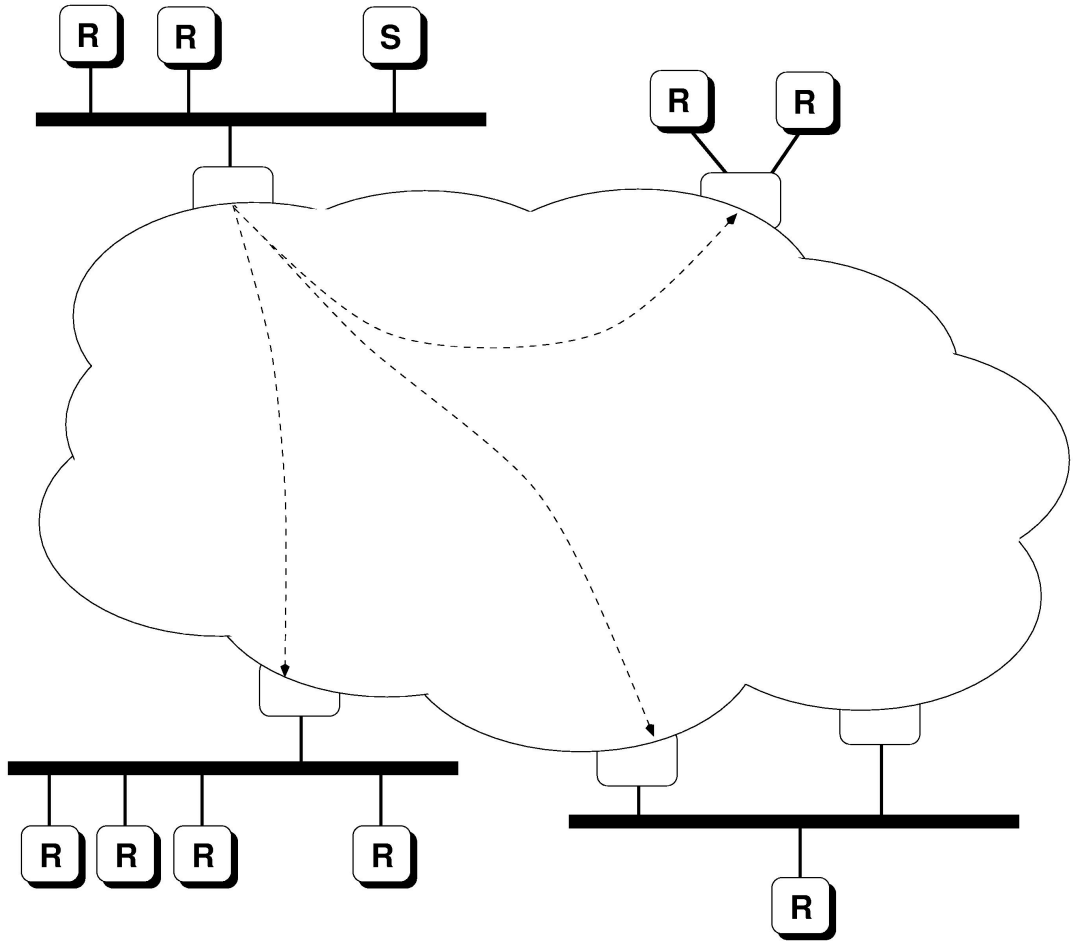
- Dynamic membership
  - Hosts can join/leave at will
- No synchronization or negotiation
  - Can be implemented a higher layer if desired

Key Design Goals

1. Delivery efficiency as good as unicast
2. Low join latency
3. Low leave latency

Network Model

- Interconnected LANs
- LANs support link-level multicast
- Map globally unique multicast address to LAN-based multicast address (LAN-specific algorithm)



Problem

- Multicast delivery widely available on individual LANs
  - Example: Ethernet multicast
- But not across interconnection of LANs

Routing Strategies

- Flooding
- Shared Spanning Tree
- Source-Based Spanning Trees
- Deering and Cheriton paper
  - Seminal work
- Reverse Path Forwarding (RPF)
- Truncated Reverse Path Broadcast (TRPB)
- Reverse Path Multicasting (RPM)

Flooding

- Same as unicast algorithm
  - A router copies a packet and transmits it on all outbound links (except the one the packet came in on)
  - Routers keep a list of sequence numbers
  - If a packet with the same sequence number has already been seen, drop the packet
- How long to keep the sequence #?
- Not scalable to … ?
  - Gnutella?

Source based trees

- Instead of building one shared spanning tree for all multicast packets, use a separate spanning tree for each source

- Each source-based spanning tree is explicitly constructed using the shortest paths from the source to all other destinations

Source based trees

- Advantages:
  - Packets follow shortest paths to all destinations
  - No duplicate packets are generated in the network
- Disadvantages:
  - Source Based Trees must be explicitly set up
  - Multicast routing tables can grow very large, since they carry separate entries for each source
  - Packets still arrive where they aren't wanted

Deering & Cheriton ’89

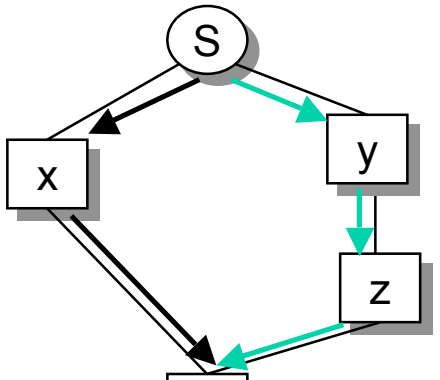
- Single spanning-tree (SST)
- Distance-vector multicast (DVM)
- Link-state multicast (LSM)
- Also: Sketches hierarchical multicast

Distance Vector Multicast Routing

- An elegant extension to Distance Vector routing
- Use shortest path DV routes to determine if link is on the source-rooted spanning tree

Reverse Path Flooding (RPF)

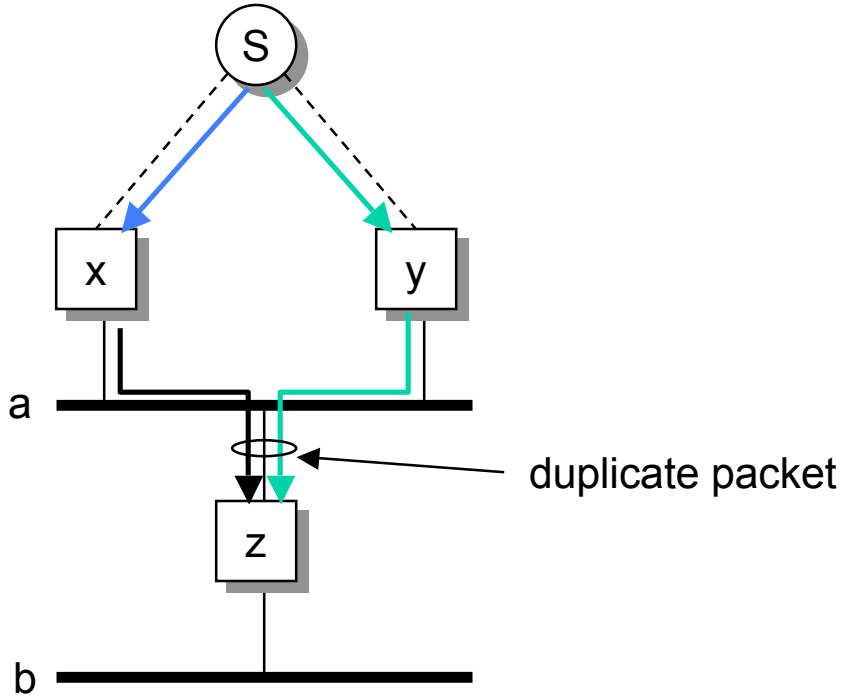
- A router forwards a broadcast packet from source (S) iff it arrives via the shortest path from the router back to S
- Packet is replicated out all but the incoming interface
- Reverse shortest paths easy to compute → just use info in DV routing tables
  - DV gives shortest reverse paths
  - Works if costs are symmetric



Forward packets that arrives on shortest path from "t" to "S" (assume symmetric routes)

RPF Limitation

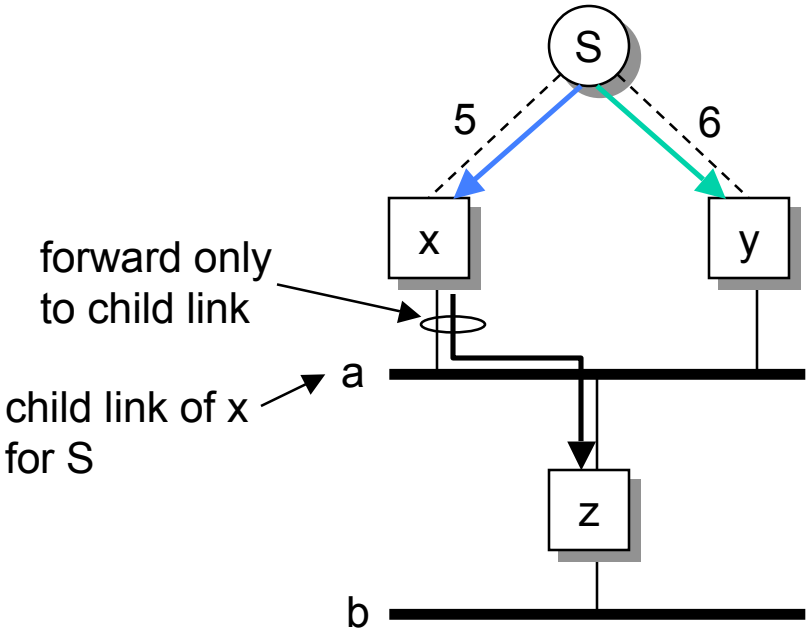
– Flooding can cause a given packet to be sent multiple times over the same link



- Solution: Reverse Path Broadcasting

Reverse Path Broadcasting (RPB)

- Basic idea: forward a packet from S only on child links for S
- Child link of router R for source S: link that has R as parent on the shortest path from the link to S



Identify Child Links

- Routing updates identify parent
- Since distances are known, each router can easily figure out if it's the parent for a given link
- In case of tie, lower address wins

Problem

- This is still a broadcast algorithm – the traffic goes everywhere
- First order solution: Truncated RPB

Truncated RBP

- Don't forward traffic onto network with no receivers
  1. Identify leaves
  2. Detect group membership in leaf

Reverse Path Multicast (RPM)

- Prune back transmission so that only absolutely necessary links carry traffic
- Use on-demand pruning so that router group state scales with number of active groups (not all groups)

Basic RPM Idea

- Prune (Source,Group) at leaf if no members
  - Send Non-Membership Report (NMR) up tree
- If all children of router R prune (S,G)
  - Propagate prune for (S,G) to parent R
- On timeout:
  - Prune dropped
  - Flow is reinstated
  - Down stream routers re-prune
- Note: again a soft-state approach

Details

- How to pick prune timers?
  - Too long → large join time
  - Too short → high control overhead
- What do you do when a member of a group (re)joins?
  - Issue prune-cancellation message (grafts)
- Both Non-Membership Report (prune) and graft messages are positively acknowledged (why?)

RPM Scaling

- State requirements:
  - O(Sources □ Groups) active state
- How to get better scaling?
  - Hierarchical Multicast
  - Core-based Trees

Core Based Trees (CBT)

- Paper: Ballardie, Francis, and Crowcroft,
  - "Core Based Trees (CBT): An Architecture for Scalable Inter-Domain Multicast Routing", SIGCOMM 93
- Similar to Deering's Single-Spanning Tree
- Unicast packet to core router and bounce it back to multicast group
- Tree construction is receiver-based
  - One tree per group
  - Only nodes on tree involved
- Reduce routing table state from O(S x G) to O(G)

Core based trees

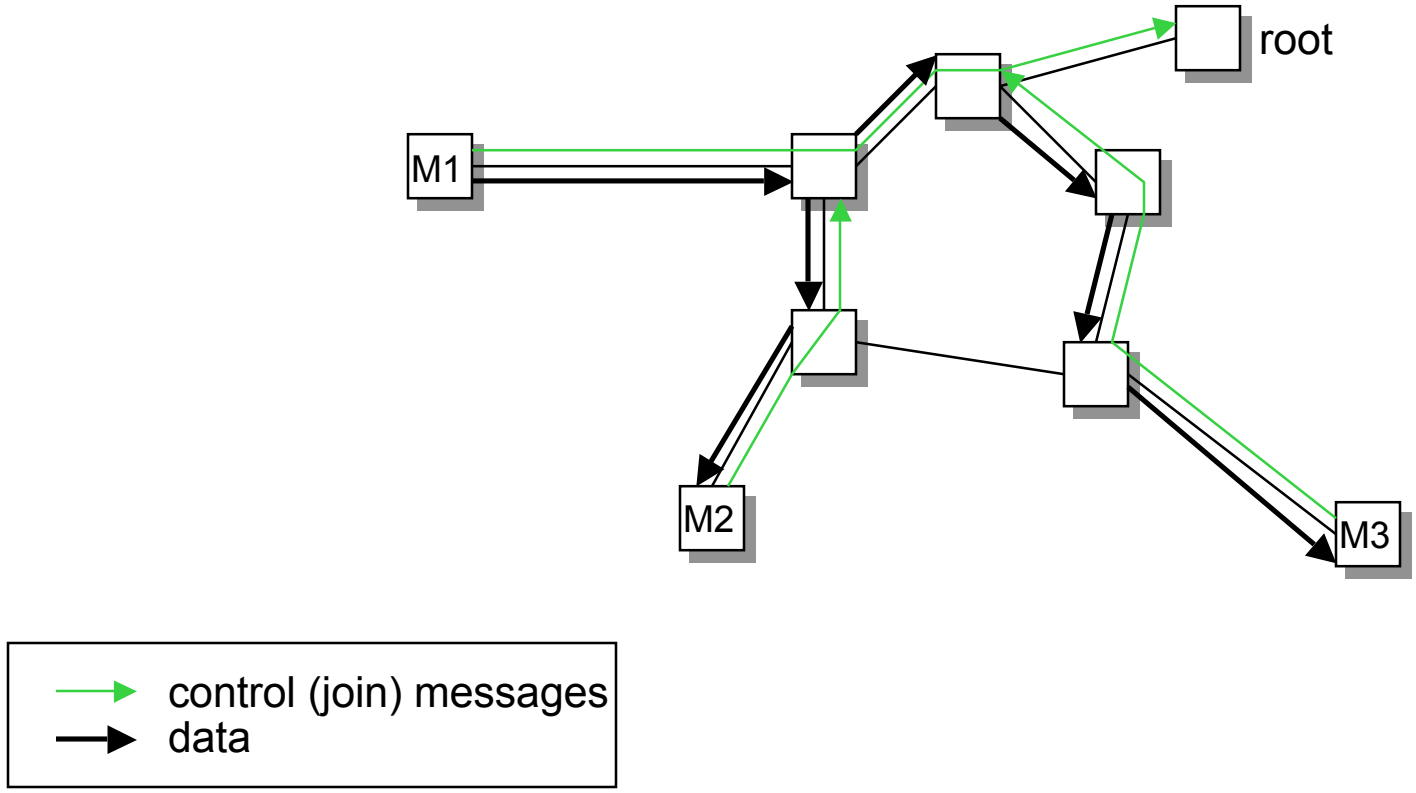
use a shared tree that connects all receivers in each multicast

- One special router in the shared tree is called the “core router”
- When a receiver wishes to join a multicast group, it sends a “join request” message toward the core router
- – As join message passes through non-core routers, branches are added to the shared tree
- When a sender wishes to send packets to a multicast group, it send the packet toward the core router.
- – The first router (core or non-core) to see the packet will intercept the packet and multicast it to all receivers on the shared tree

- Sender unicasts packet for group G to core C
- G is included as IP option, core strips option, punches new header (S, G)

Example

- Group members: M1, M2, M3
- M1 sends data



CBT: Advantages & Disadvantages

- Advantages:
  - Smaller router tables, so more scalable
  - only one entry per multicast group
  - not one entry per (source, group) pair like RPM
  - Senders do not need to join a group to send to it
- Disadvantages:
  - Shared trees are not as optimal as source-based trees
  - Core routers can become bottlenecks

IP Multicast Revisited

- Despite many years of research and many compelling applications, and despite the fact that the many of today routers implement IP multicast, this is still not widely deployed
- Why?

Possible Explanations [Holbrook & Cheriton '99]

- Violates ISP input-rate-based billing model
  - No incentive for ISPs to enable multicast!
- No indication of group size (again needed for billing)
- Hard to implement sender control → any node can send to the group (remember open group semantic?)
- Multicast address scarcity

Summary

- Deering's DV-RMP an elegant extension of DV routing
- CBT addresses some of the DV-RMP scalability concerns but is sub-optimal and less robust
- Protocol Independent Multicast (PIM)
  - Sparse mode similar to CBT
  - Dense mode similar to DV-RPM
- Lesson: economic incentives plays a major role in deploying a technical solution
  - See EXPRESS work

Scalable Application Layer Multicast

Suman Banerjee, Bobby Bhattacharjee, Christopher Kommareddy
Department of Computer Science,University of Maryland, College Park, MD 20742, USA

Application-Layer multicast

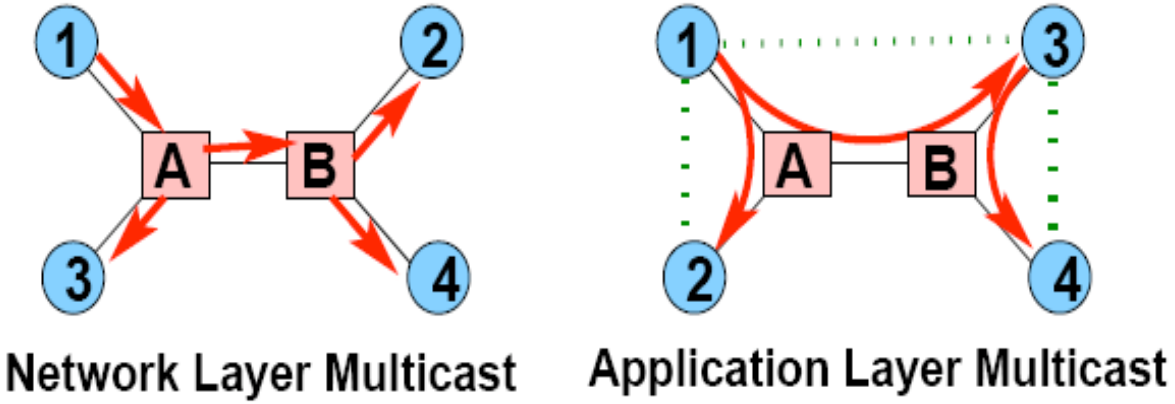
- Failure of in-network multicast
- Change too hard
  - Protocols
  - Existing Router infrastructure
  - Business models
- Solution:
  - Move multicast to edges of the network.

NICE protocol

- Introduction
- Solution overview
- Protocol description
- Simulation experiment
- Implementation
- Conclusion

Application-Layer Multicast

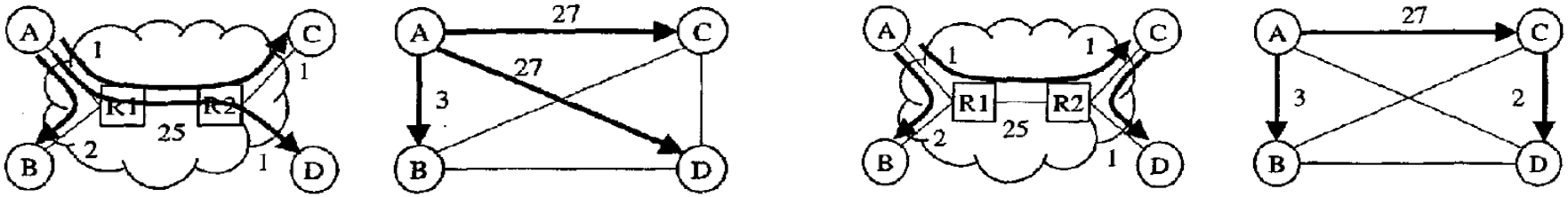
- Multicasting is an efficient way for packet delivery in one-many data transfer applications



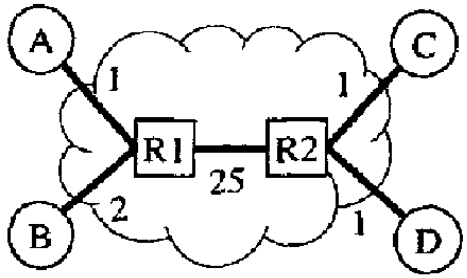
native multicast where data packets are replicated at routers inside the network, in application-layer multicast data packets are replicated at end hosts

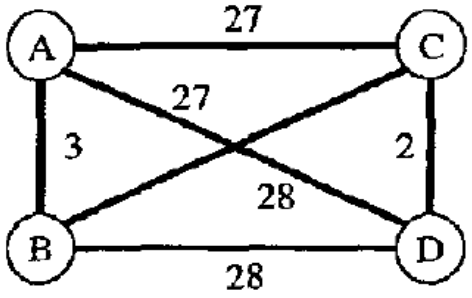
New metrics

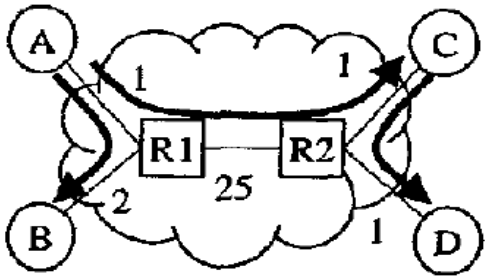
- Two intuitive measures of “goodness” for application layer multicast overlays, namely *stress* and *stretch*
- The stress metric is defined per-link and counts the number of identical packets sent by a protocol over each underlying link in the network.
  EX: stress of link A-R1 is 2 (__)
- The stretch metric is defined per-member and is the ratio of path-length from the source to the member along the overlay to the length of the direct unicast path.
  EX: <A,D> = 29/27 , <A,B> = <A,C> = 1

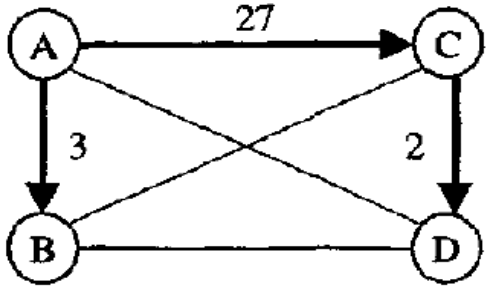


Introduction (Previous work-Narada Protocol)





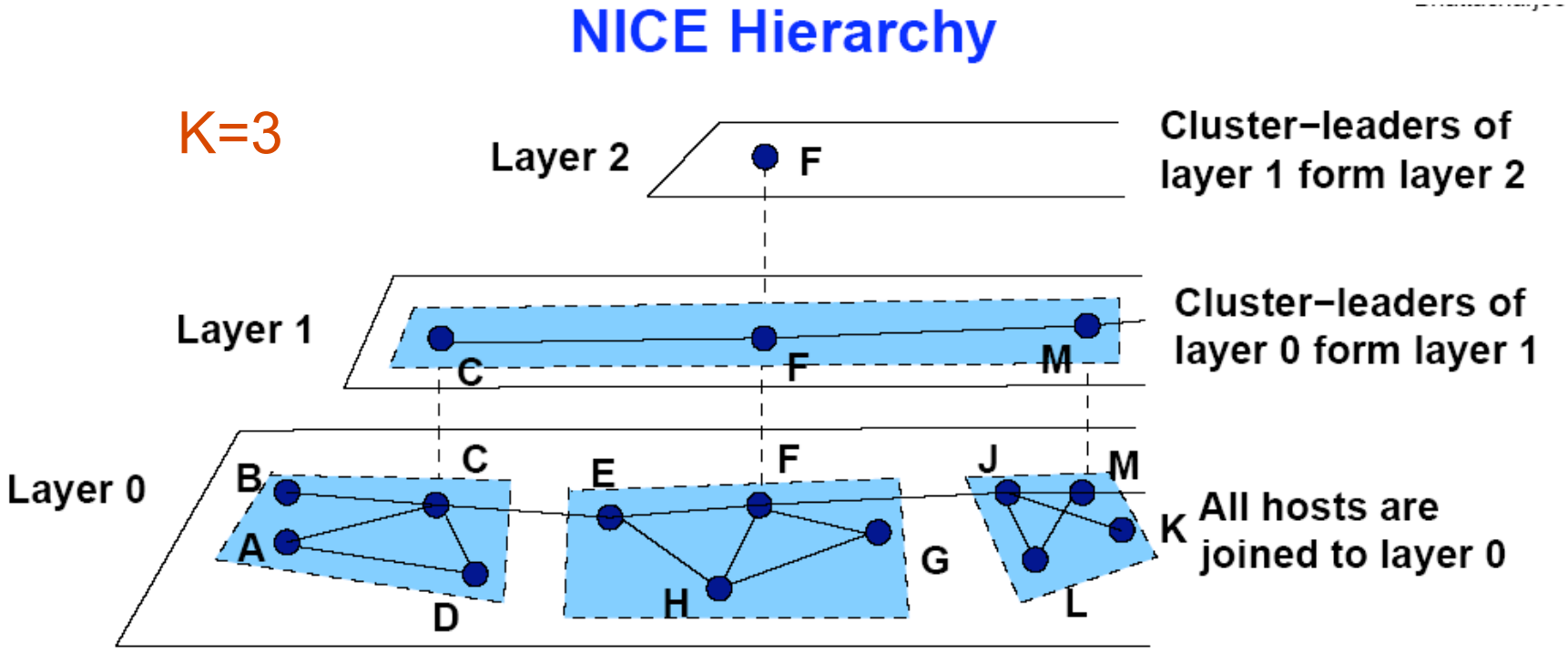




- Mesh based approach
- Shortest path spanning tree
- Maintaining group state of all members

Solution overview

(Hierarchical Arrangement of Members)

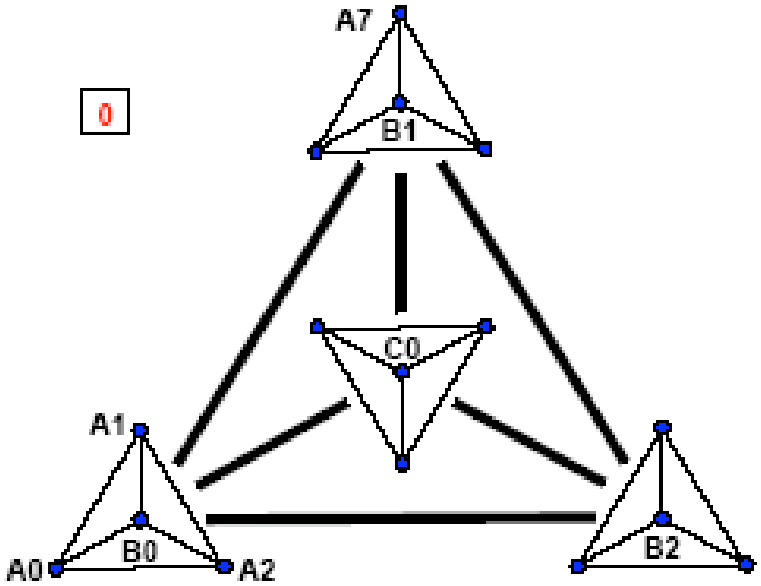


- clusters of size between K and 3K-1

Solution overview

(**Control and Data Paths**)

- Control path is the path between peers in a cluster
- The neighbors on the control topology exchange periodic soft state refreshes and do not generate high volumes of traffic.
- In the worst case, both state and traffic overhead is $O(k \log N)$.

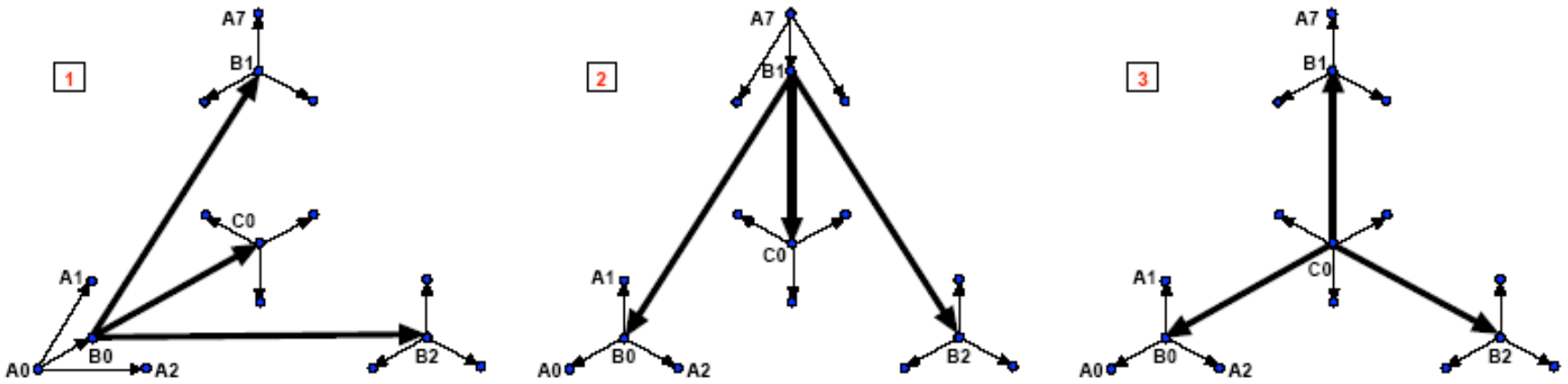


Solution overview

(Control and Data Paths)

- in the NICE protocol we choose the data delivery path to be a tree.

- (1) A0 is the source
  (2) A7 is the source
  (3) C0 is the source



Solution overview

(Invariants)

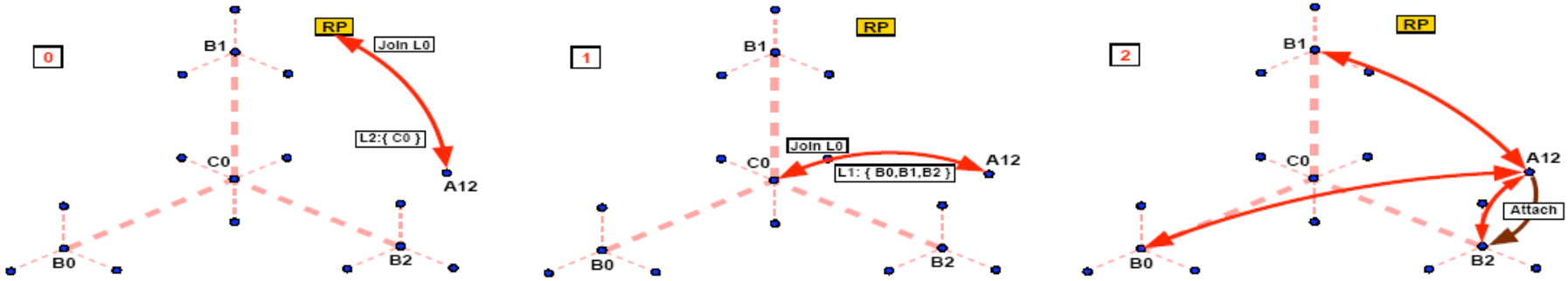
Specifically the protocol described in the next section maintains the following set of invariants:

- At every layer, hosts are partitioned into clusters of size between K and 3K-1
- All hosts belong to an L_0 cluster, and each host belongs to only a single cluster at any layer
- The cluster leaders are the centers of their respective clusters and form the immediate higher layer.

Protocol description

(New Host Joins)

- The new host A12 contacts RP (Rendezvous Point) first.
- RP responds with a host in highest layer:A12 now contacts member in it highest layer. Host C0 then informs all the members of its cluster i.e. B0,B1 and B2.
- A12 then contacts each of these members with the join query to identify the closest member among them , and iteratively uses this procedure to find its cluster.



Protocol description

(Cluster Maintenance and Refinement)

Cluster Split and Merge

A cluster-leader periodically checks the size of its cluster, and appropriately splits or merges the cluster when it detects a size bound violation.

- Split:
  - It is done when Size > 3k-1
  - Forms equal sized clusters
  - Chooses a new leader

- Merge:
  - Done when size < k
  - Merges with neighboring clusters at the same layer
  - Chooses new leader

Protocol description

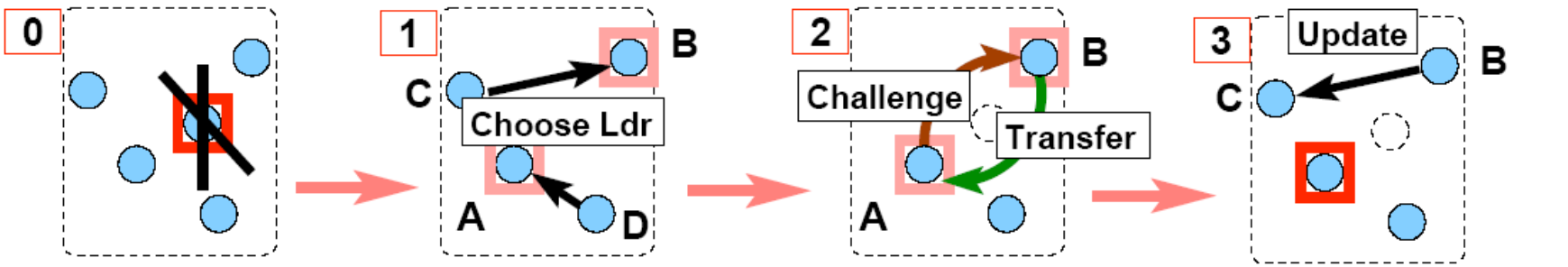
(Cluster Maintenance and Refinement)

Refining Cluster Attachments

- When a member is joining a layer, it may not always be able to locate the closest cluster in that layer (e.g. due to lost join query or join response, etc.)
- Each member periodically probes all members in its super-cluster , to identify the closest member to itself in the super-cluster

Protocol description

(Host Departure and Leader Selection)



Leader lost

Each member independently chooses new leader

Competing leaders will reconcile

Single leader selected

Performance Metrics

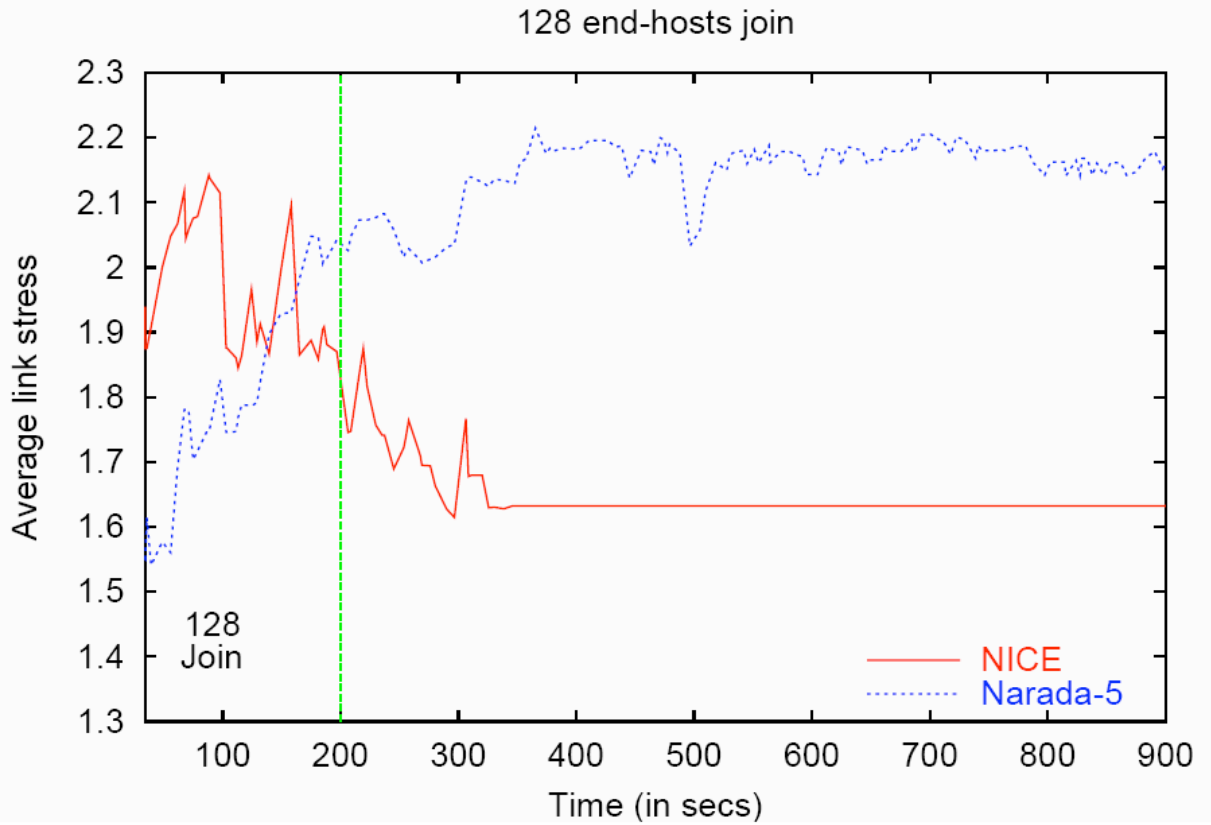
- Quality of data path:
  > Stretch
  > Stress
  > Latency

- Failure recovery:
  Measured fraction of (remaining) members that correctly receive the data packets sent from the source as the group membership changed.

- Control traffic overhead*:*
  Byte overheads at routers and end-hosts.

Stress

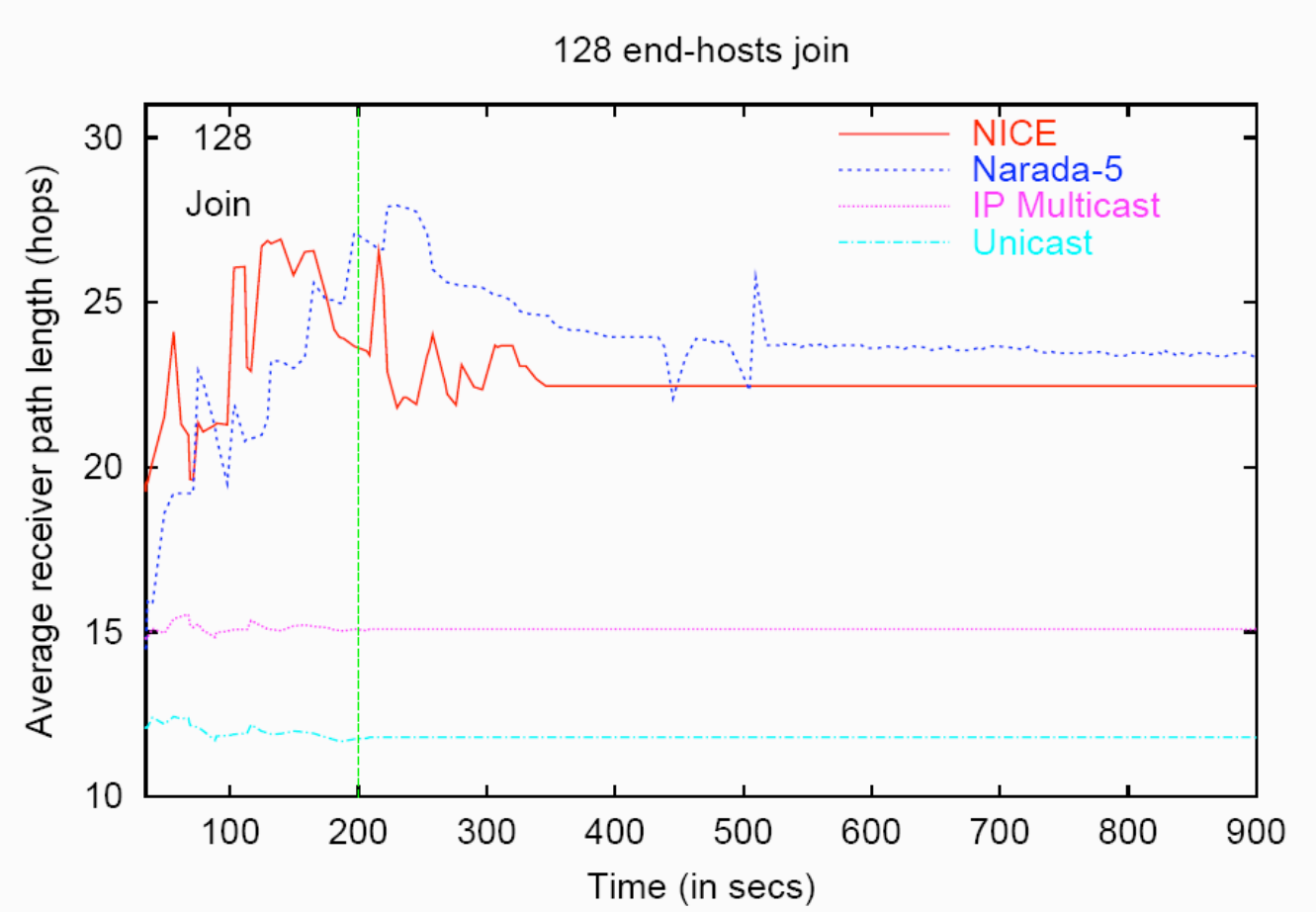
(simulation)



- Nice protocol converges to a stable value within 350 seconds.
- Narada uses fewer number of links on the topology than NICE
- Nice reduces the average link stress.

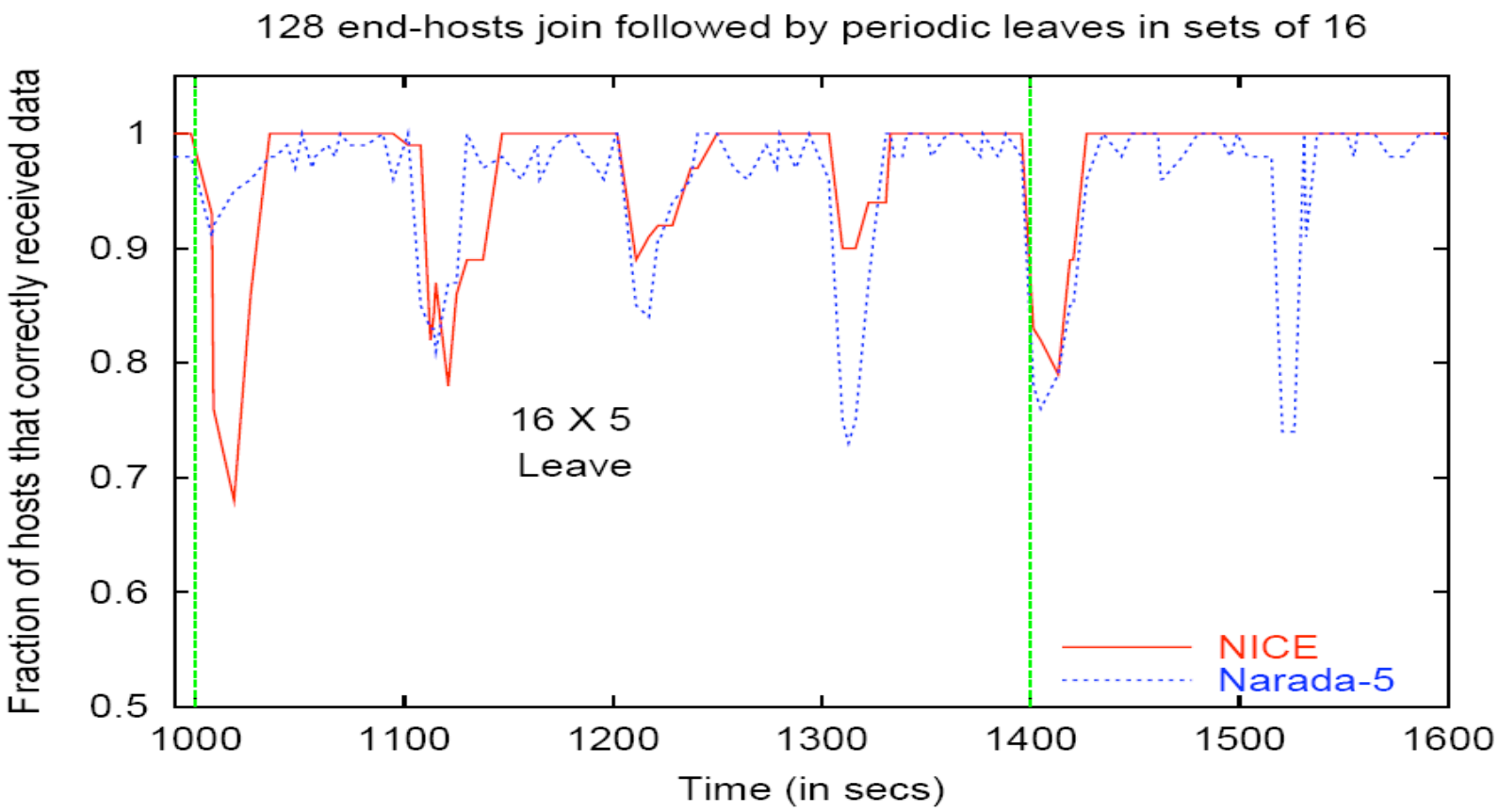
Path Length Experiments

(simulation)



- the conclusion is that the data path lengths to receivers were similar for both protocols.

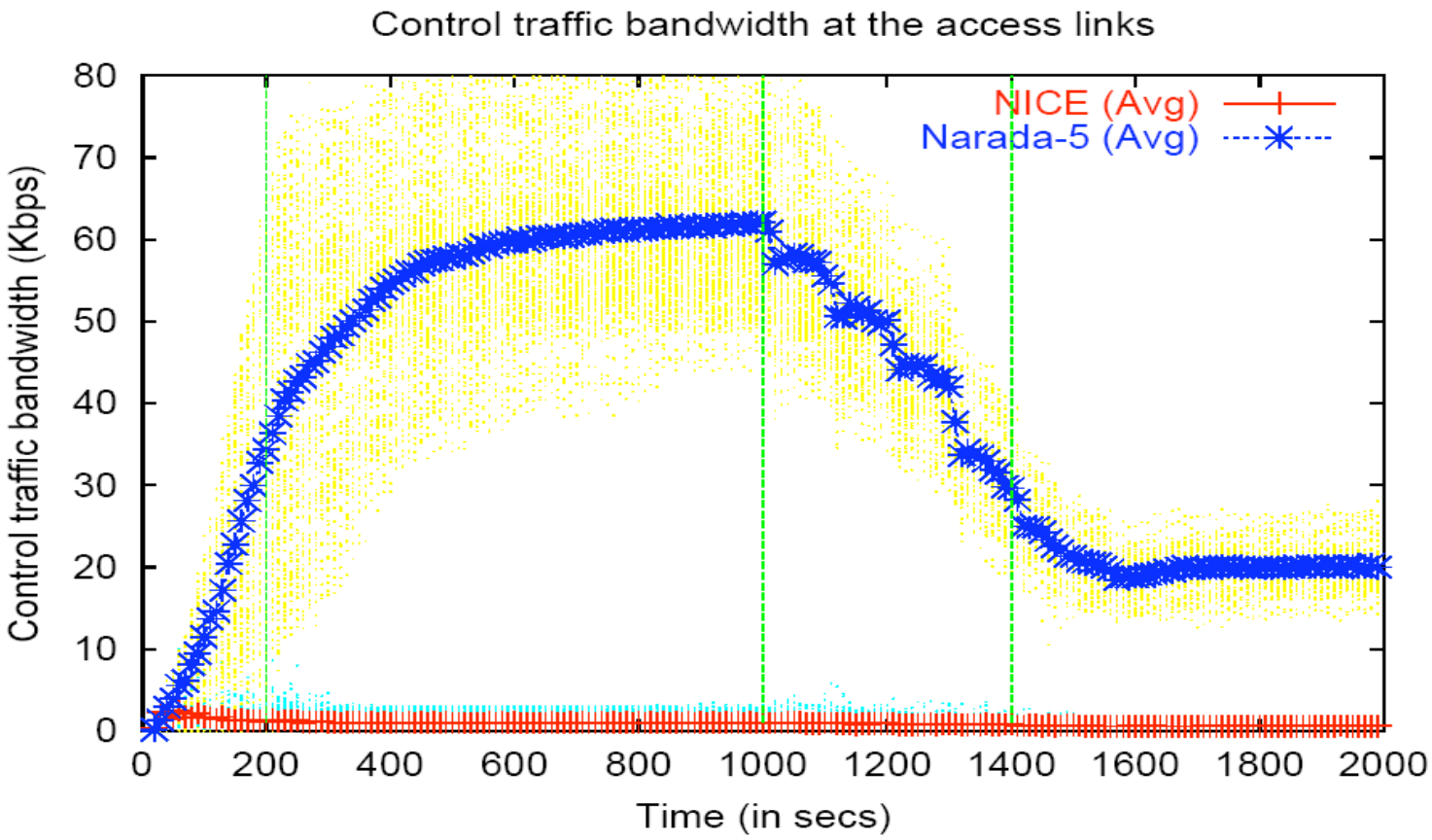
Packet loss experiments



- **Both protocols have similar performance.**

Control Traffic

(simulation)



- Nice had a lower average overhead

Results Overview

Group Size	Router Stress		Link Stress		Path Length		Bandwidth Overheads (Kbps)	
	Narada-5	NICE	Narada-5	NICE	Narada-5	NICE	Narada-30	NICE
8	**1.55** (1.30)	3.51 (3.30)	**1.19** (0.39)	3.24 (2.90)	25.14 (9.49)	**12.14** (2.29)	**0.61** (0.55)	1.54 (1.34)
16	**1.84** (1.28)	2.34 (2.16)	**1.34** (0.76)	1.86 (1.39)	**19.00** (7.01)	20.33 (6.75)	2.94 (2.81)	**0.87** (0.81)
32	**2.13** (2.17)	2.42 (2.60)	**1.54** (1.03)	1.90 (1.82)	20.42 (6.00)	**17.23** (5.25)	9.23 (8.95)	**1.03** (0.95)
64	2.68 (3.09)	**2.23** (2.25)	1.74 (1.53)	**1.63** (1.39)	22.76 (5.71)	**20.62** (7.40)	26.20 (28.86)	**1.20** (1.15)
128	3.04 (4.03)	**2.36** (2.73)	2.06 (2.64)	**1.63** (1.56)	**21.55** (6.03)	21.61 (7.75)	65.62 (92.08)	**1.19** (1.29)
256	3.63 (7.52)	**2.31** (3.18)	2.16 (3.02)	**1.63** (1.63)	**23.42** (6.17)	24.67 (7.45)	96.18 (194.00)	**1.39** (1.76)
512	4.09 (10.74)	**2.34** (3.49)	2.57 (5.02)	**1.62** (1.54)	24.74 (6.00)	**22.63** (6.78)	199.96 (55.06)	**1.93** (3.35)
1024	-	2.59 (4.45)	-	1.77 (1.77)	-	25.83 (6.13)	-	2.81 (7.22)
1560	-	2.83 (5.11)	-	1.88 (1.90)	-	24.99 (6.96)	-	3.28 (9.58)
2048	-	2.92 (5.62)	-	1.93 (1.99)	-	24.08 (5.36)	-	5.18 (18.55)

Table 1: Data path quality and control overheads for varying multicast group sizes (simulation)

- Path lengths and failure recovery similar for NARADA and NICE
- Stress (and variance of stress) is lower with NICE
- NICE has much lower control overhead

Testbed implementation

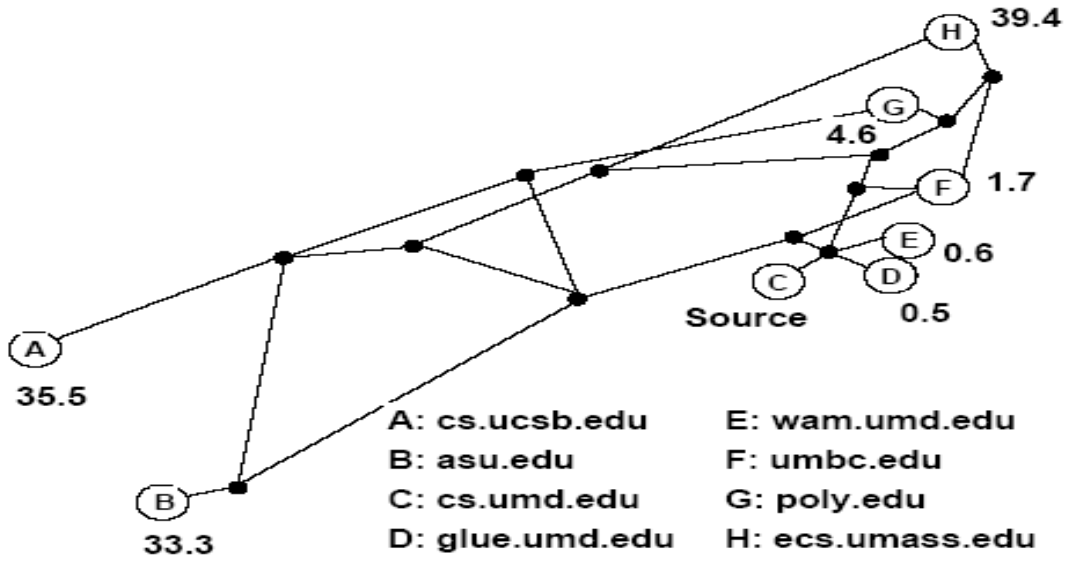


Figure 13: Internet experiment sites and direct unicast latencies from C

- 32 to 100 member groups distributed across 8 different sites.
- The number of members at each site was varied between 2 and 30
- indicate the typical direct unicast latency (in milliseconds) from the site C

Testbed experiments

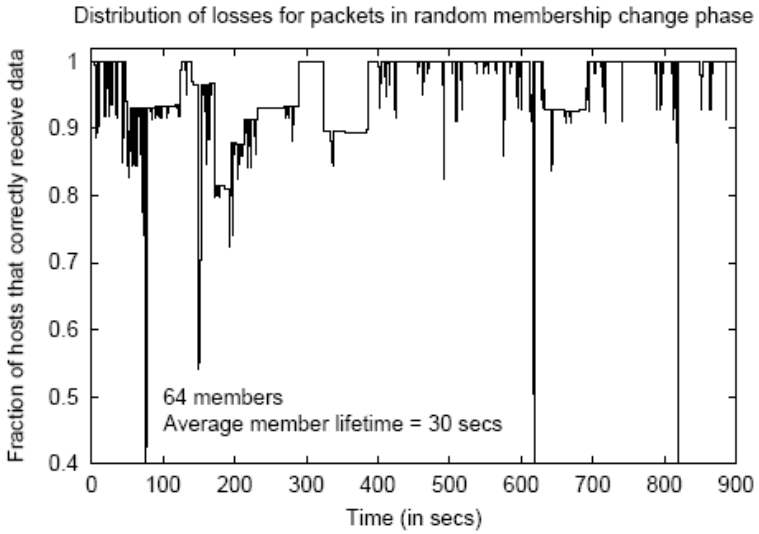


Figure 17: Fraction of members that received data packets as group membership continuously changed (testbed)

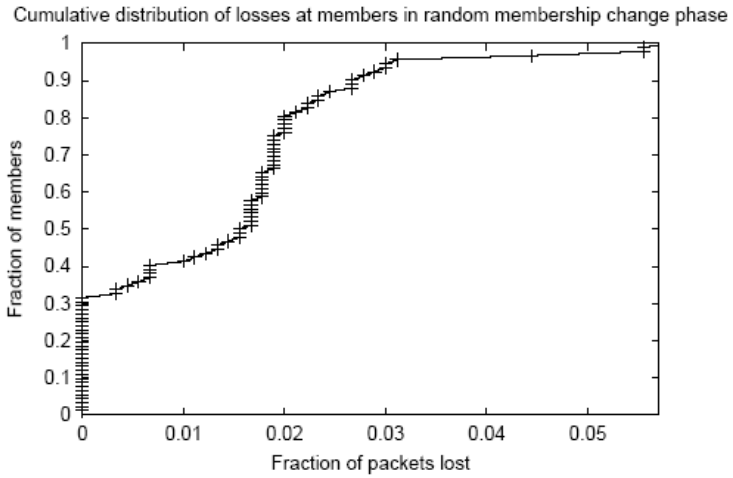


Figure 18: Cumulative distribution of fraction of packets lost for different members out of the entire sequence of 900 packets during the rapid membership change phase (testbed)

Group Size	Stress		Stretch		Control overheads (Kbps)	
	Mean	Max.	Mean	Max.	Mean	Max.
32	1.85	8.0	1.08	1.61	0.84	2.34
64	1.73	8.0	1.14	1.67	0.77	2.70
96	1.86	9.0	1.04	4.63	0.73	2.65

Table 2: Average and maximum values of of the different metrics for different group sizes(testbed)