CSC 525:
Computer Networks

Overlay Networks

- Overlay network
  - A network built on top of an existing network
  - Add an additional layer of indirection/virtualization to implement a new service
  - Rely on the underlay network for basic services
- Example: Internet is an overlay network
  - goal: connect local area networks
  - built on local area networks (e.g., Ethernet), phone lines
  - add an Internet Protocol header to all packets
  - Between two IP routers, there can be multiple layer-2 devices, but IP routers are not aware of them.
- Today overlay networks often refer to application level networks over the IP network.

Benefits

- Easier deployment
  - Don't have to modify underlying hardware or software
  - Only need to deploy new software on top of existing software
  - e.g., adding IP on top of Ethernet does not require modifying Ethernet protocol or driver
- Better separation of functionality

Costs

- Overhead
  - Adds a layer in protocol stack
    - Additional packet headers, processing
  - The overlay doesn't have enough knowledge of underlying network to optimize performance.
  - Sometimes, certain work is redundant across the layers.
- Complexity
  - Layering does not eliminate complexity, it only manages it
  - More layers of functionality → more possible unintended interaction between layers

Applications

- File sharing, Multicast, Unicast, Security and Privacy, Mobility, etc.
- We'll cover a number of systems. Pay attention to their differences and commonality.
  - E.g., what overlay topology does each system use, and how do they build and maintain such topologies?

Overlay Multicast

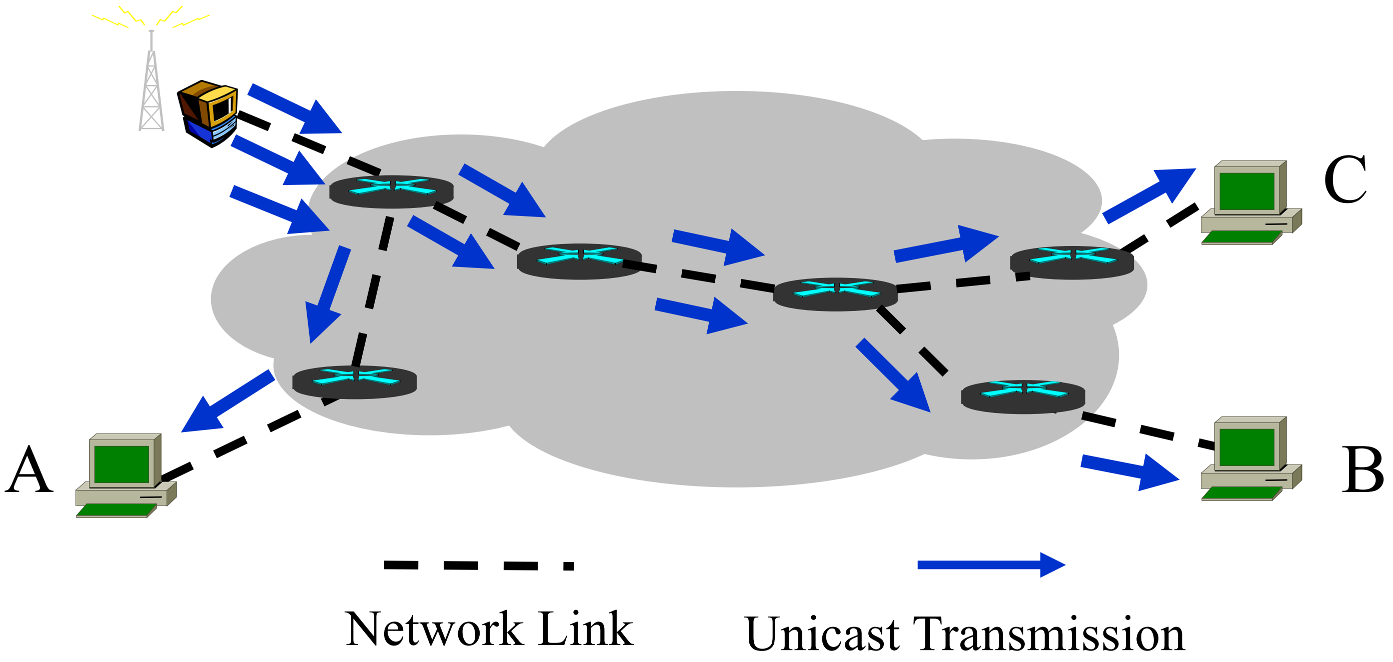
- Provide multicast functionality above the IP layer: **application layer multicast**
- Challenge: do this efficiently and scalable
- Narada
  - Support multi-source multicast
  - Involve only end hosts
  - Small groups <= a few hundreds of nodes

Concerns with IP Multicast

- Scalability with the number of groups
  - Routers maintain per-group state
  - Aggregation of multicast addresses is not very effective.
- Supporting higher level functionality is difficult
  - IP Multicast: best-effort multi-point delivery service
  - End systems responsible for handling higher level functionality
    - Reliability, congestion control, security, address allocation, charging and accounting, etc.
- Deployment is difficult and stagnant (at inter-domain).

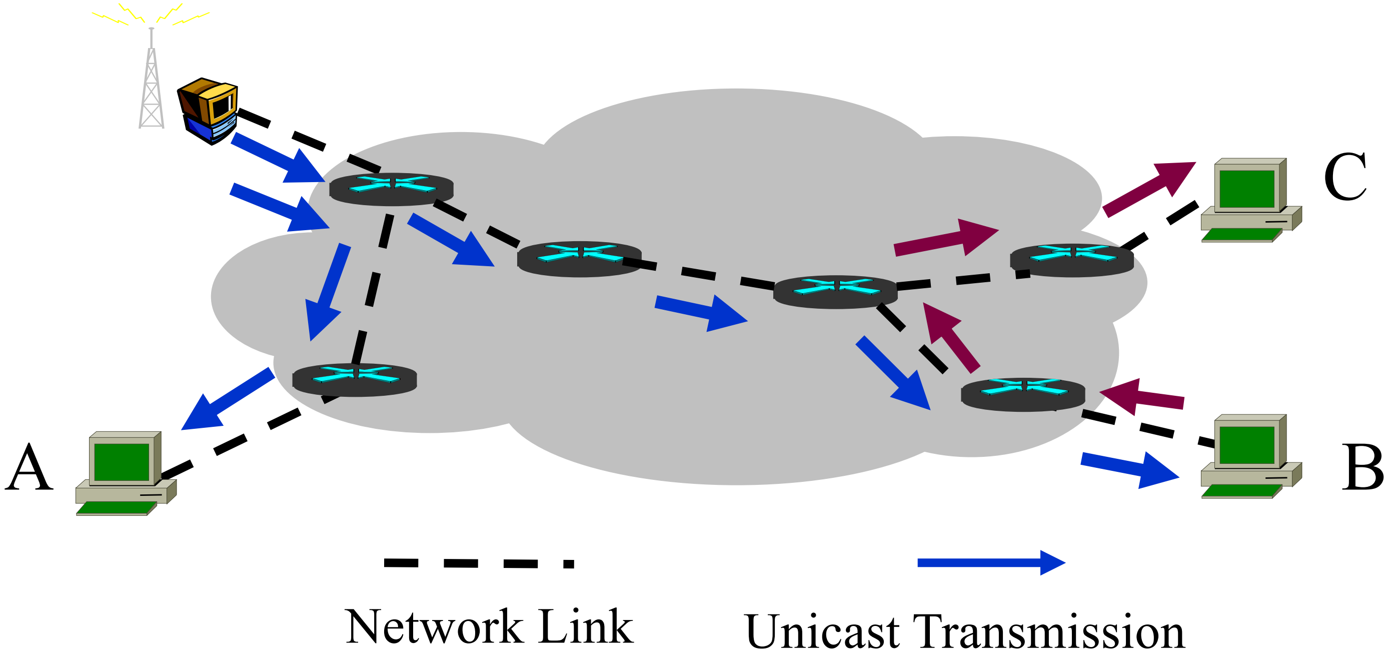
End System Multicast

- Hosts forward packets, without router support
- Build a smarter forwarding tree
  - Repetitive Unicast doesn't scale
  - Duplicate transmission and extra delay inevitable, but should be minimized.



End System Multicast

- Hosts forward packets, without router support
- Build a smarter forwarding tree
  - Repetitive Unicast doesn't scale
  - Duplicate transmission and extra delay inevitable, but should be minimized.

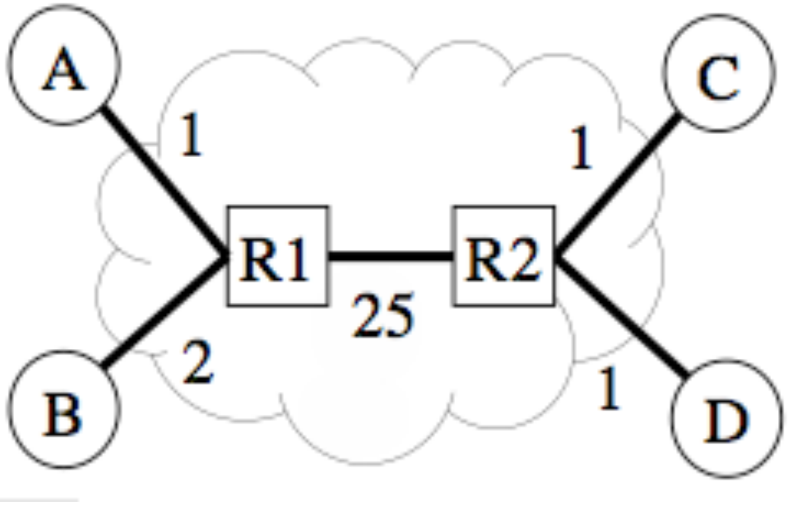


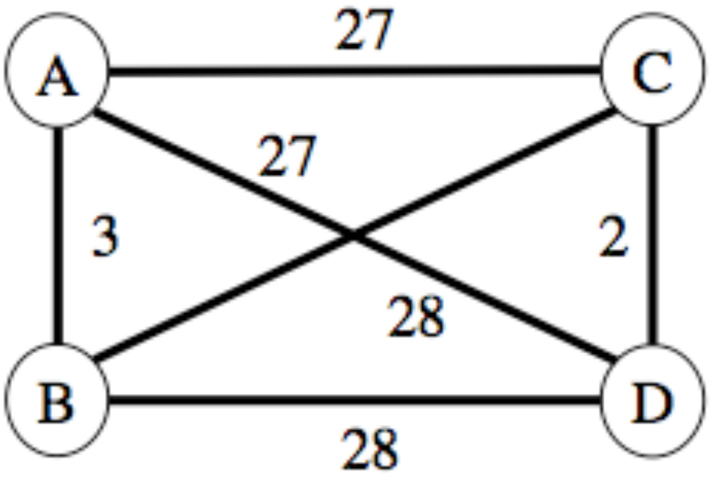
Potential Benefits

- Routers don’t maintain per-group state
  - End systems do, but they participate in much fewer groups
- Much easier to deploy
- Potentially simplify support for higher level functionality
  - Leverage computation and storage of end systems
  - For example, for buffering packets, transcoding, ACK aggregation
  - Leverage solutions for unicast congestion control and reliability

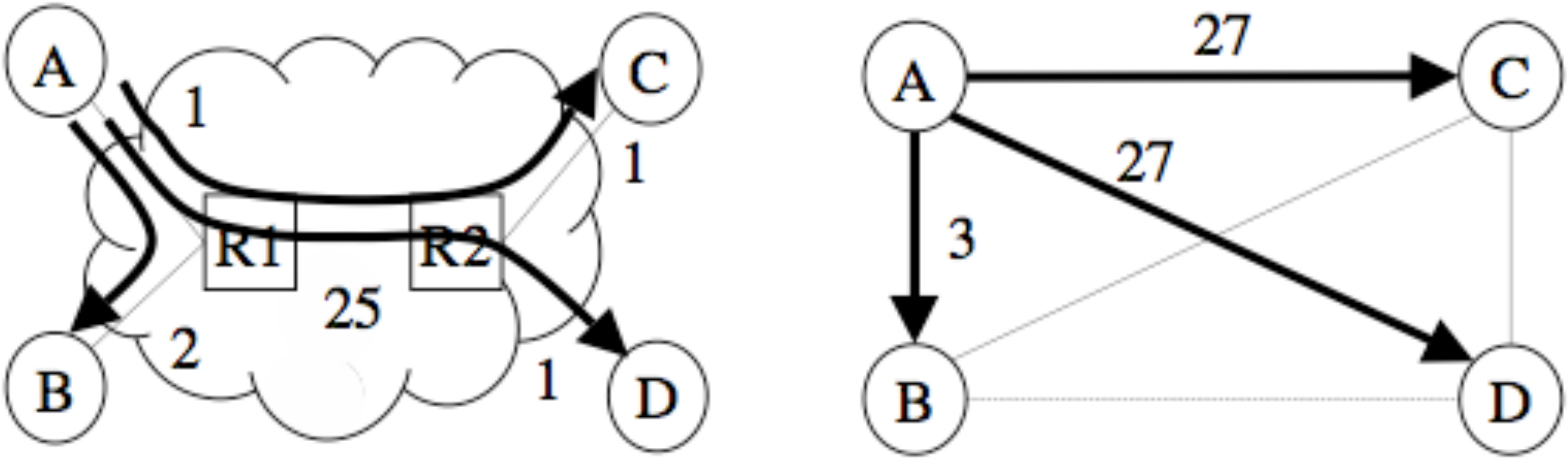
What is a good multicast overlay?

- Conflicting goals: shorter delay and small node degree
- Challenge: End hosts don't know the underlying topology

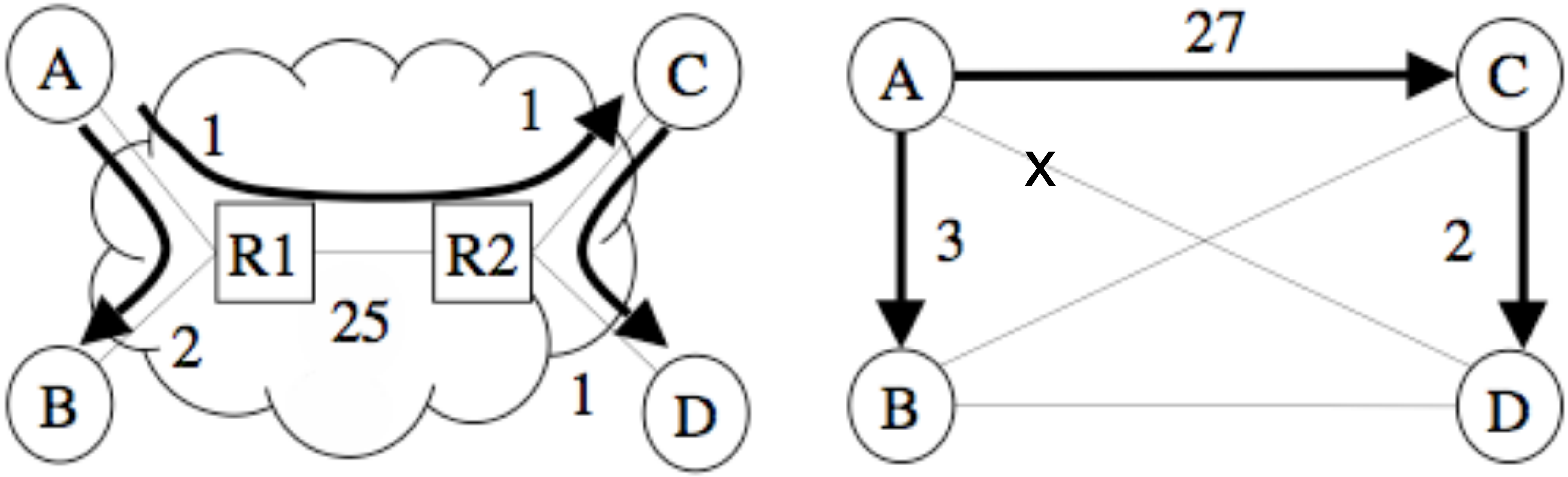




- Cannot directly run a conventional multicast routing protocol



- Need trim the complete graph first



Narada Design

Step 1

"**Mesh**": Richer overlay topology that includes all group members, but not a complete graph.

- Members have low degrees
- Shortest path delay between any pair of members along mesh is small

Step 2

- Construct per-source trees using well known multicast routing algorithms (reverse path checking)
  - Members have low degrees
  - Small delay from source to receivers

Narada Components

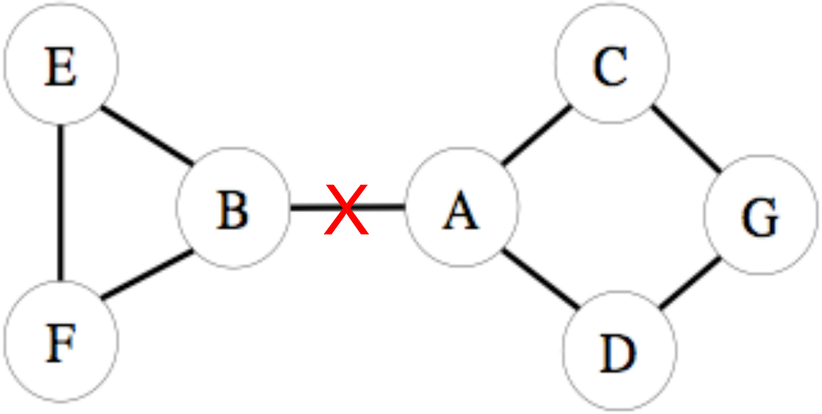
- Mesh Management:
  - Ensure mesh remains connected in face of membership changes
- Mesh Optimization:
  - Distributed heuristics for ensuring path delay between members along the mesh is small
- Forwarding tree construction:
  - Routing algorithms for constructing data-delivery trees
  - Distance vector routing, and reverse path forwarding

Mesh Management

- Every node maintains a list of **all** the group members
- Periodically exchange the list with neighbors
- A new node gets a partial list of existing members by an out-of-band mechanism.
- Need to remember dead nodes for a while.

Repairing Mesh Partition

- Partition is detected when some member entries are not refreshed as expected
- Probe these members by probability
  - Either they're dead, or establish another virtual link to repair the partition.

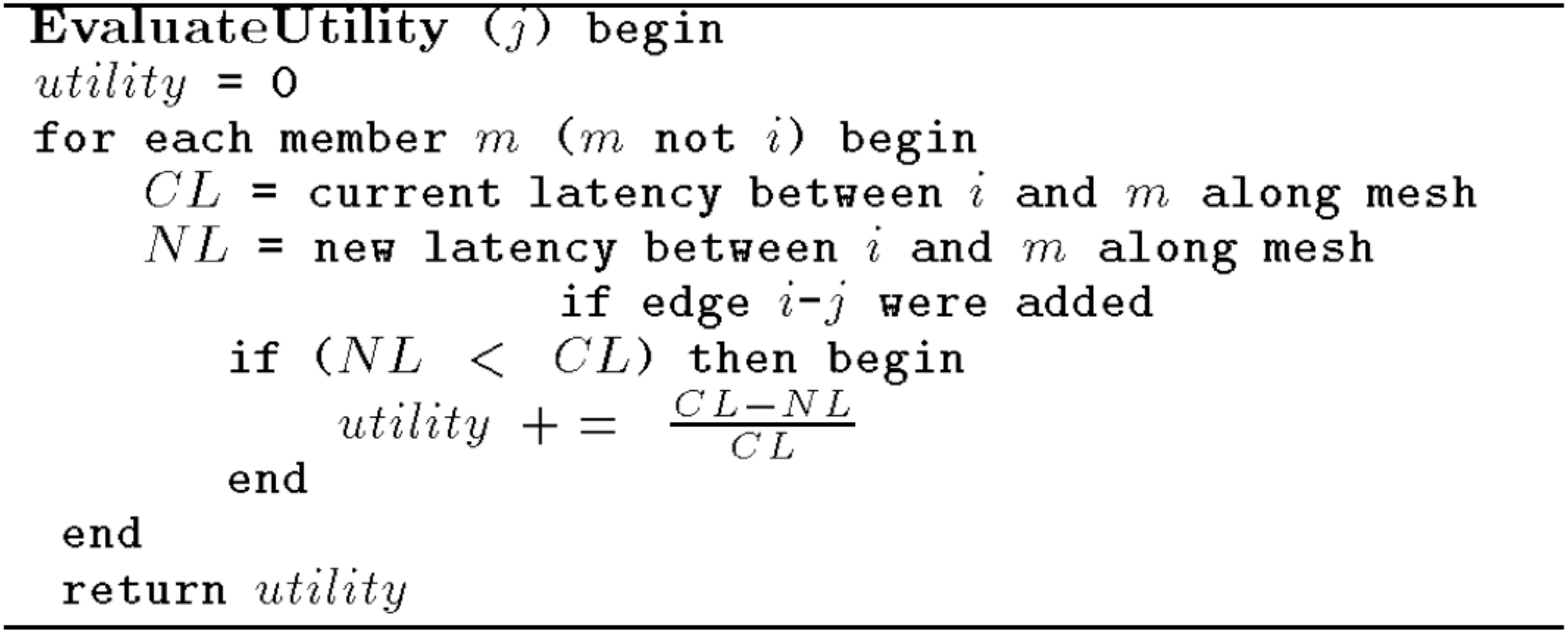


Improve Mesh Quality

- Initially a new node randomly chooses some existing members to connect to.
- Over time, add or drop virtual links to improve the mesh quality.
- Keep the node degree under a threshold.

Add Overlay Links

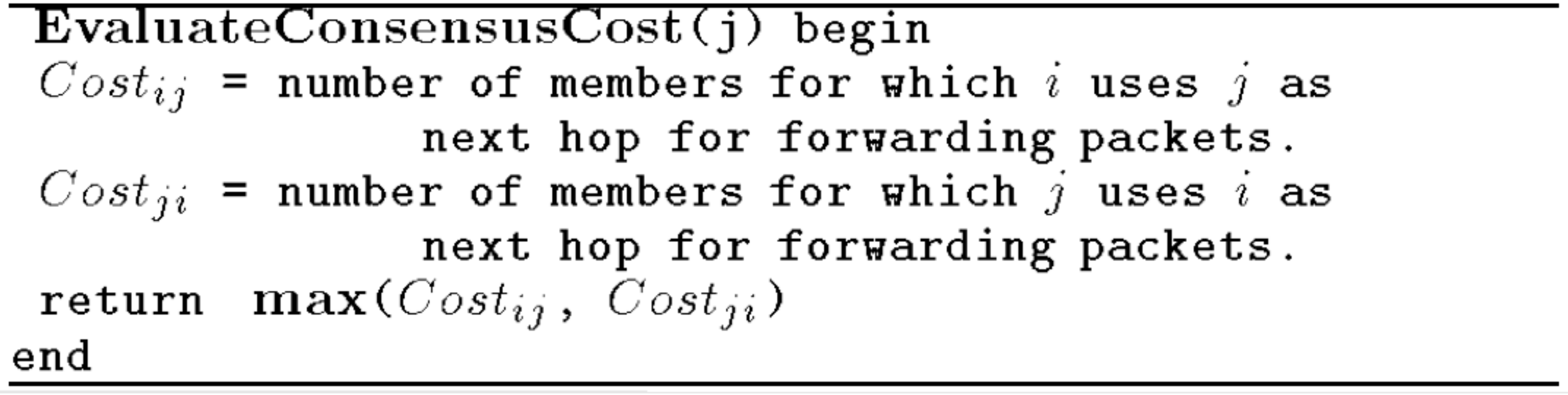
- Node i adds link i-j if *utility*(j) is greater than a threshold.



```
EvaluateUtility ($j$) begin
utility = 0
for each member $m$ ($m$ not $i$) begin
    $CL$ = current latency between $i$ and $m$ along mesh
    $NL$ = new latency between $i$ and $m$ along mesh
                    if edge $i$-$j$ were added
        if ($NL < CL$) then begin
            $utility += \frac{CL-NL}{CL}$
        end
 end
 return $utility$
```

Drop Overlay Links

- Node i drops link i-j if *consensus cost* (j) is less than a threshold.



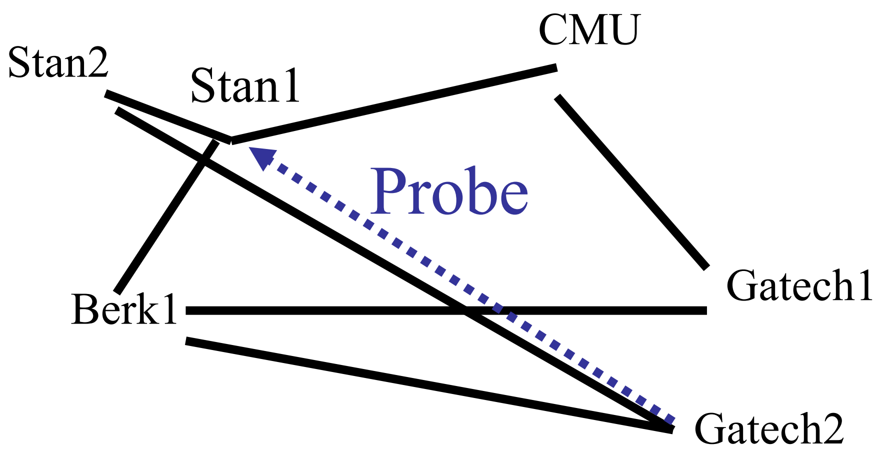
```
EvaluateConsensusCost(j) begin
 $Cost_{ij}$ = number of members for which $i$ uses $j$ as
                next hop for forwarding packets.
 $Cost_{ji}$ = number of members for which $j$ uses $i$ as
                next hop for forwarding packets.
 return  max($Cost_{ij}$, $Cost_{ji}$)
end
```

Multicast Routing

- Similar to DVMRP
  - Run distance vector protocol on top of the mesh
  - Use RPF to derive the per-source forwarding tree
- Leverage existing routing protocol.

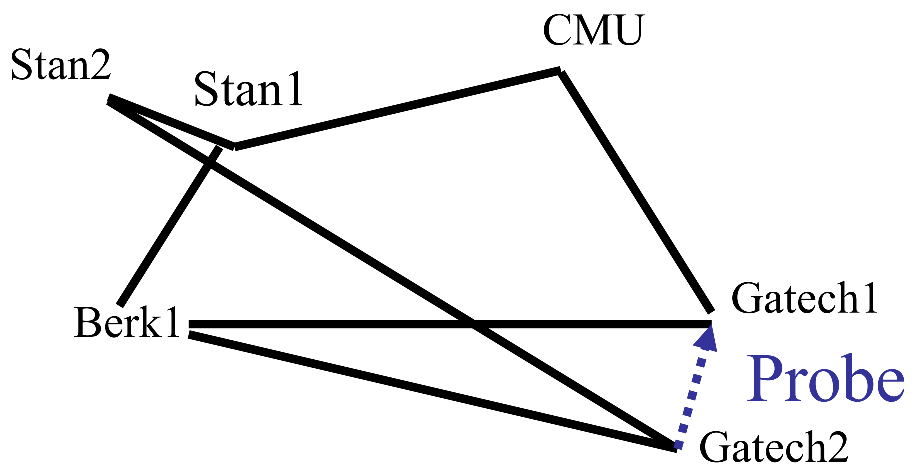
Desirable properties of heuristics

- Stability: A dropped link will not be immediately added back
- Partition Avoidance: A partition of the mesh is unlikely to be caused as a result of any single link being dropped



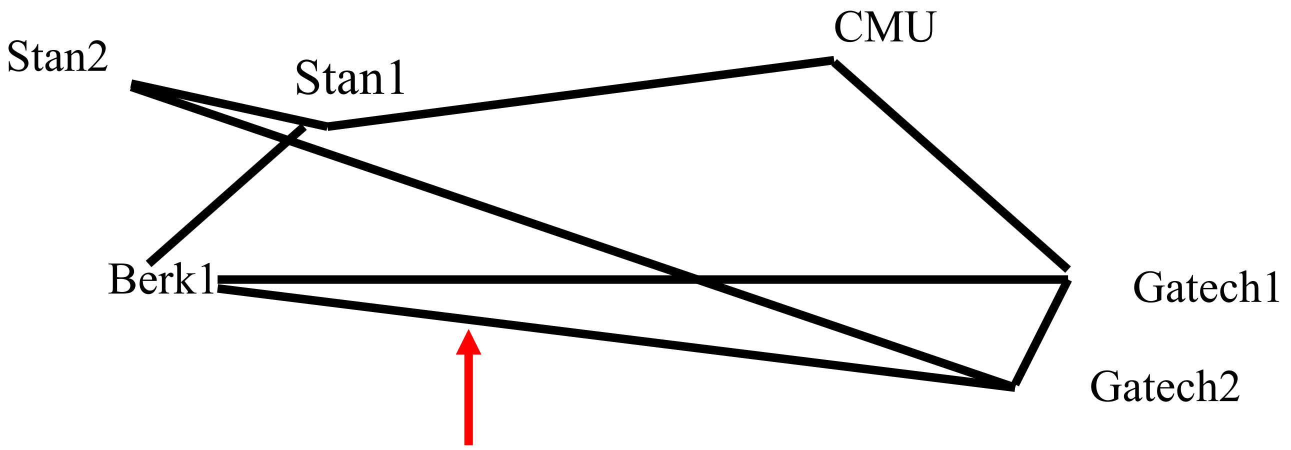
Delay improves to Stan1, CMU but marginally.

Do not add link!



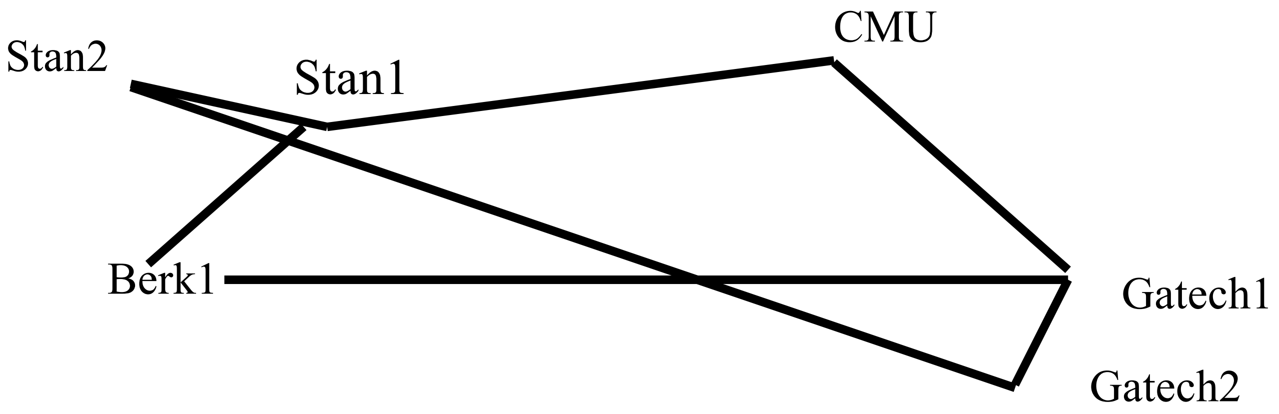
Delay improves to CMU, Gatech1 and significantly.

Add link!



Used by Berk1 to reach only Gatech2 and vice versa.

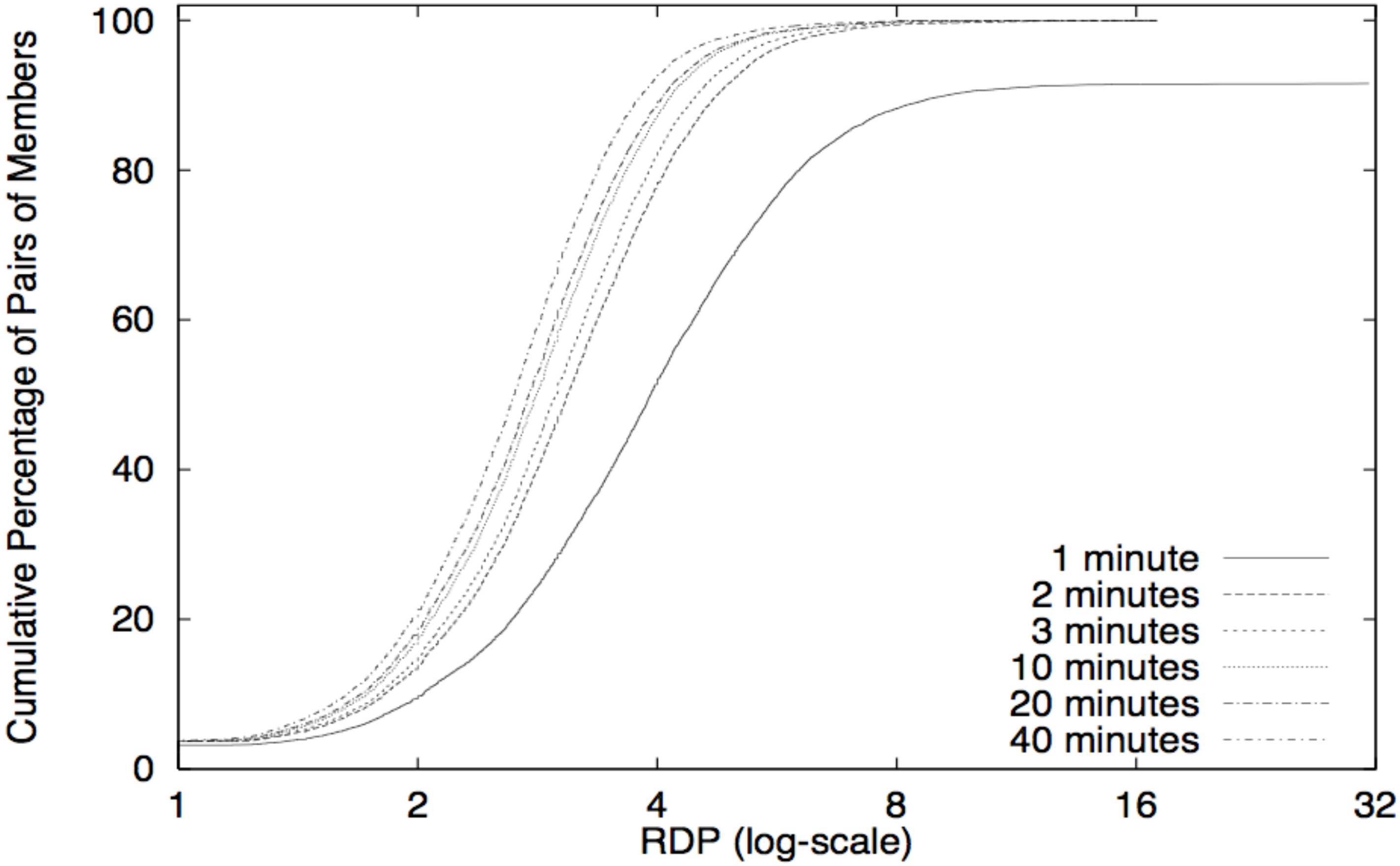
Drop!!



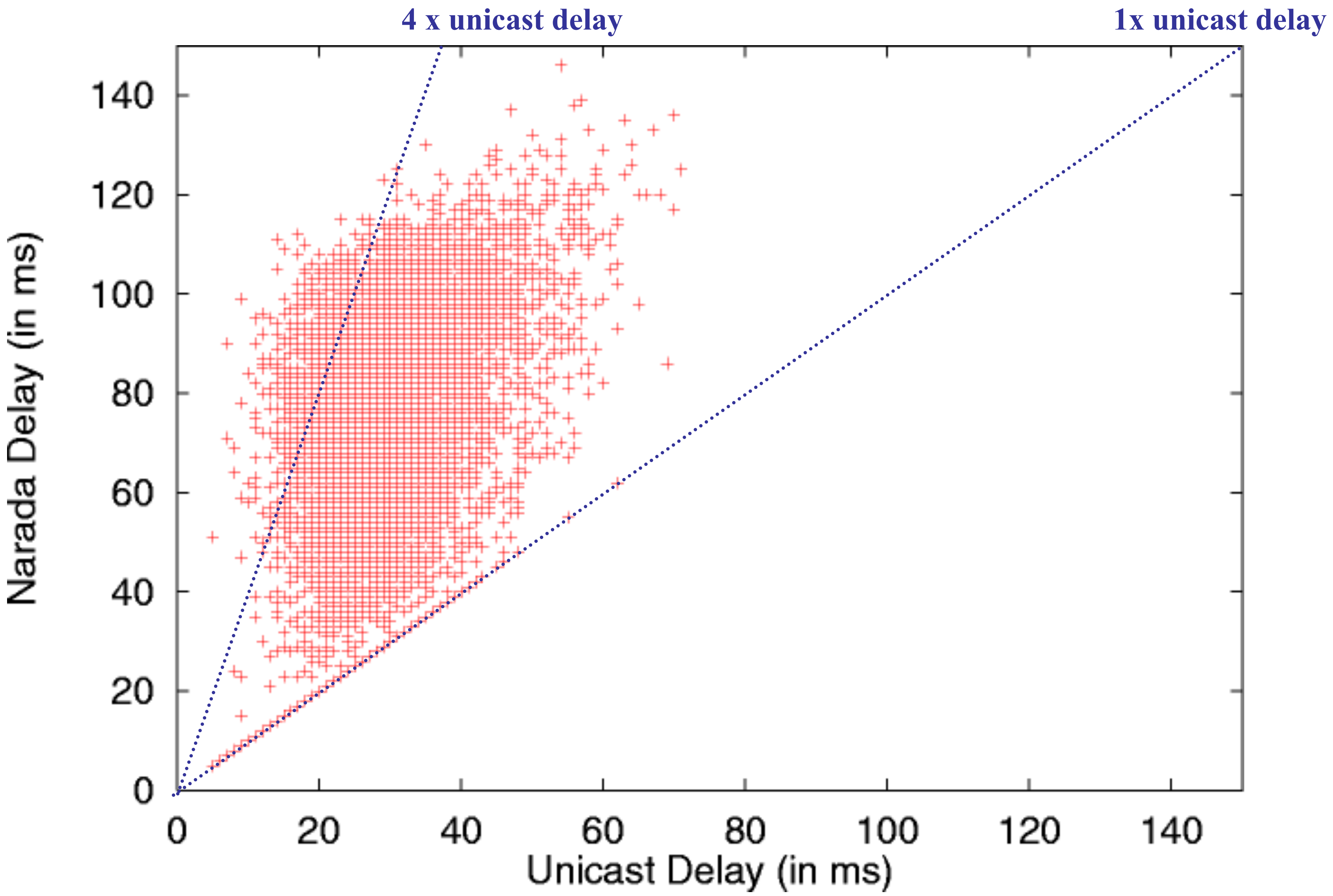
An improved mesh

Member Join

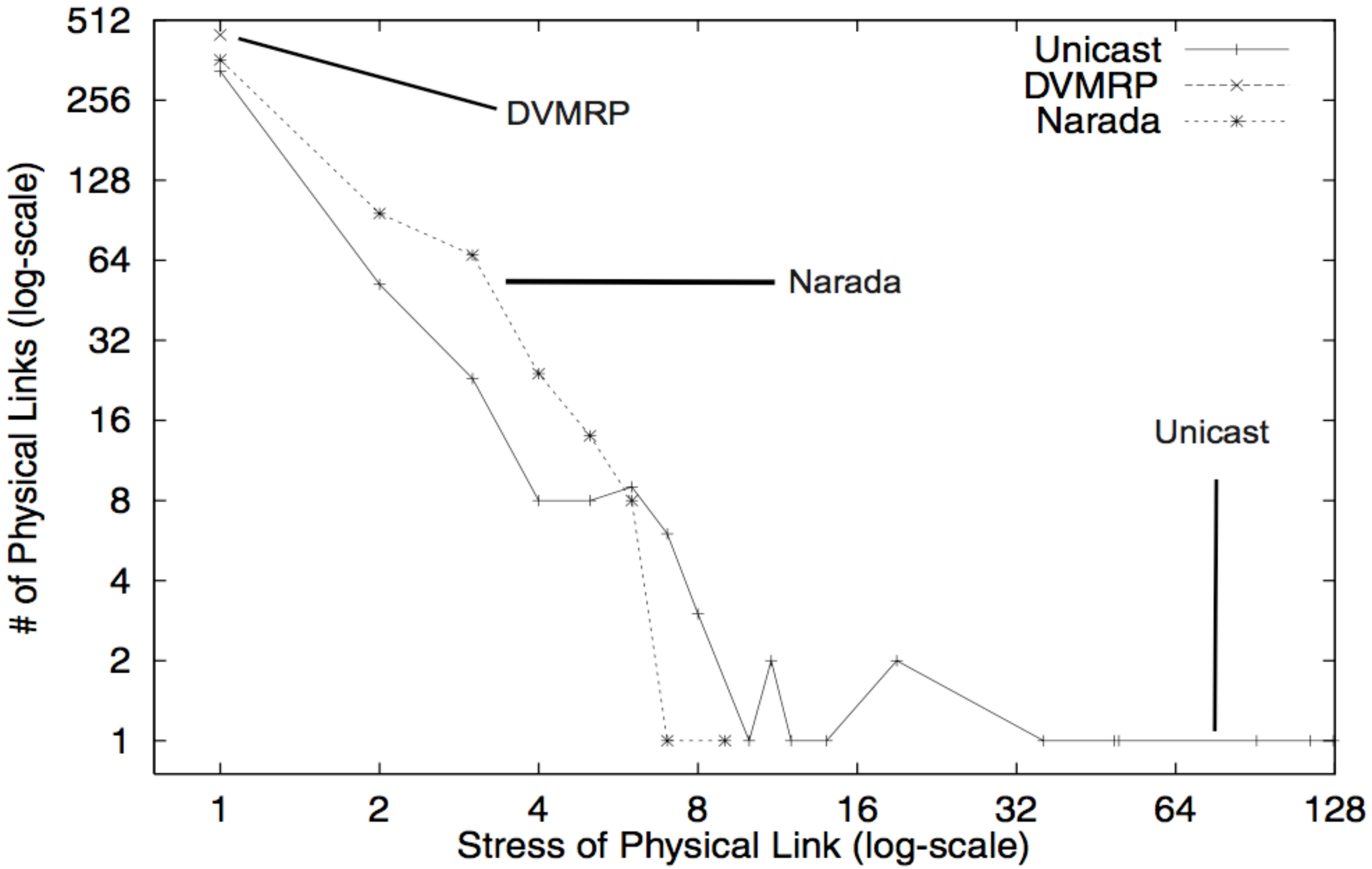
- Mesh improves over time
- But still changes after 40 minutes



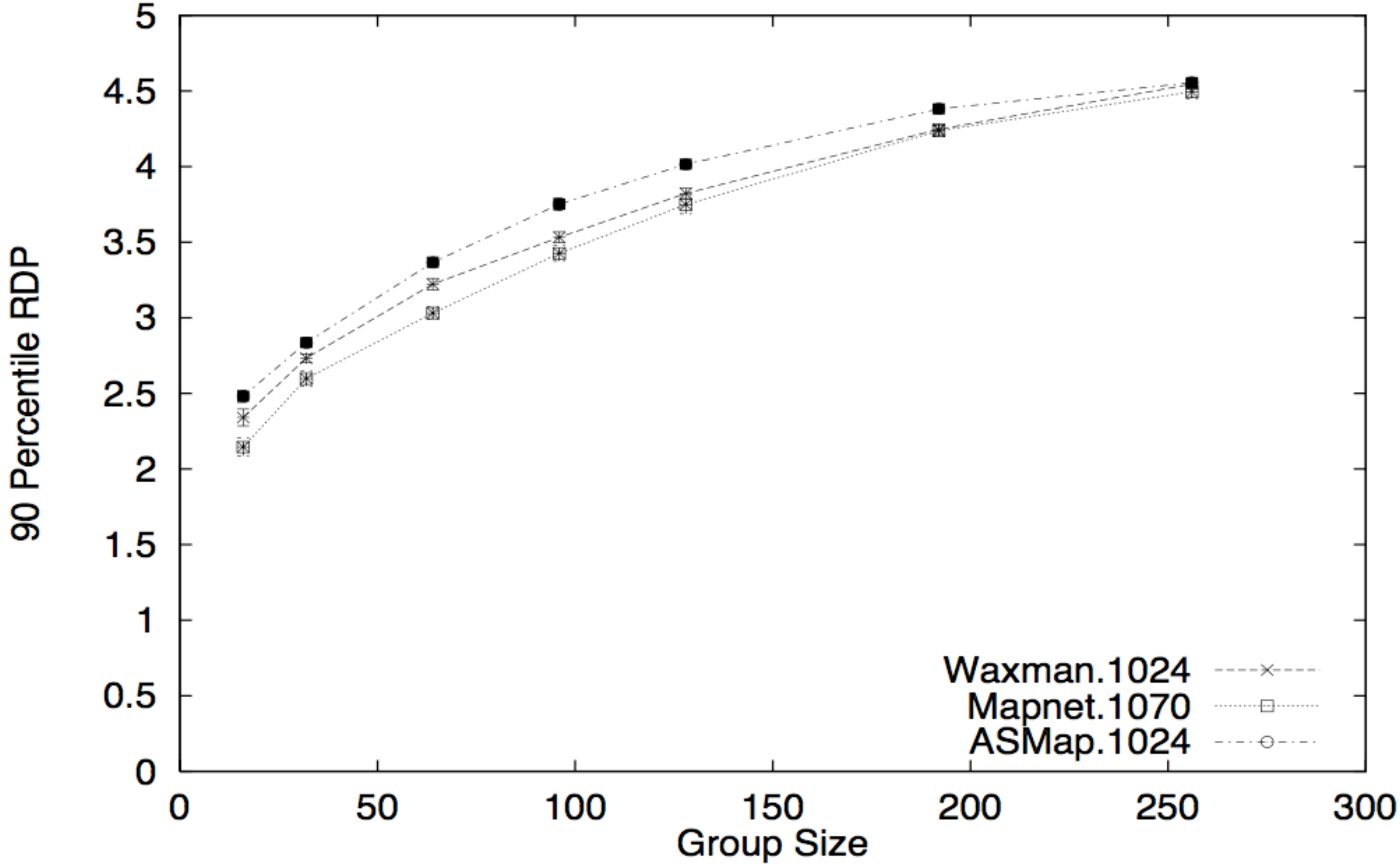
Delay


4 x unicast delay
1x unicast delay
Narada Delay (in ms)
0
20
40
60
80
100
120
140
Unicast Delay (in ms)
0
20
40
60
80
100
120
140

Link Stress


512
256
128
64
32
16
8
4
2
1
# of Physical Links (log-scale)
1
2
4
8
16
32
64
128
Stress of Physical Link (log-scale)
Unicast
DVMRP
Narada
DVMRP
Narada
Unicast

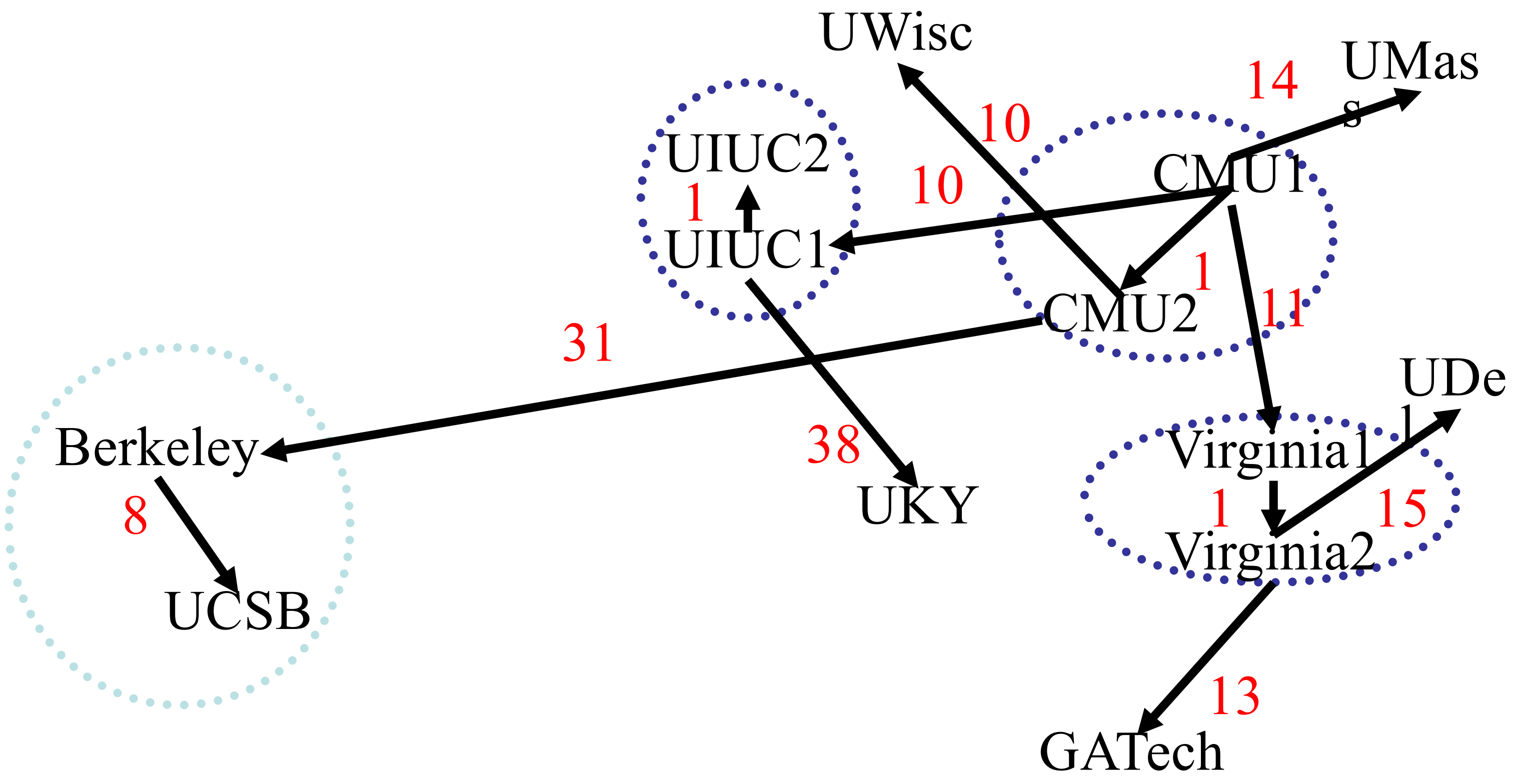
Group Size



90 Percentile RDP
Group Size
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5
0
50
100
150
200
250
300
Waxman.1024
Mapnet.1070
ASMap.1024

Internet Evaluation

- 13 hosts, all join the group at about the same time
- No further change in group membership
- Each member tries to maintain 2-4 neighbors in the mesh
- Host at CMU designated source



Conclusions

- For small-sized groups, an end-system overlay approach
  - is feasible
  - has a low performance penalty compared to IP Multicast
  - has the potential to simplify support for higher layer functionality
  - allows for application-specific customizations
- Where to implement multicast
  - IP layer or Application layer or both ?