Eliminating Bottlenecks in Overlay Multicast

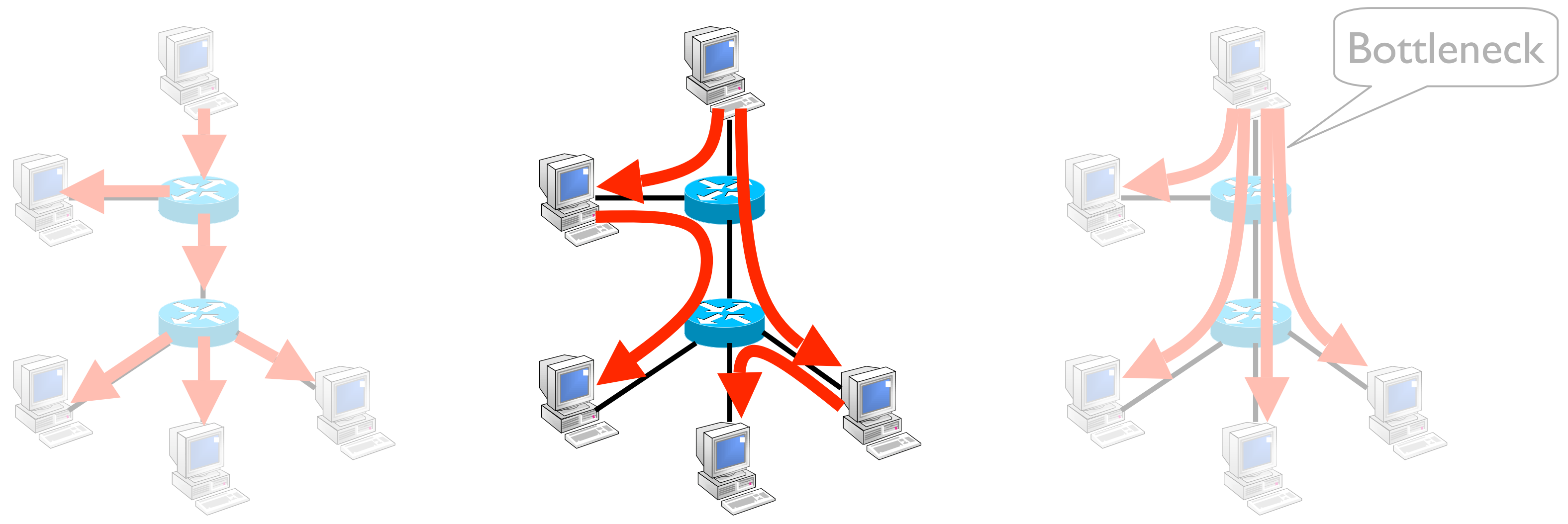
Min Sik Kim Yi Li Simon S. Lam

Department of Computer Sciences

The University of Texas at Austin

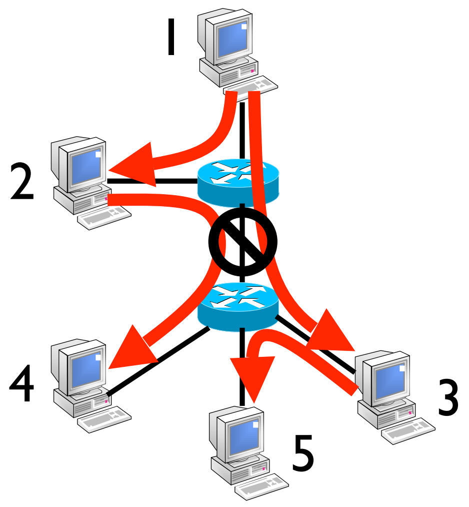
Overlay Multicast

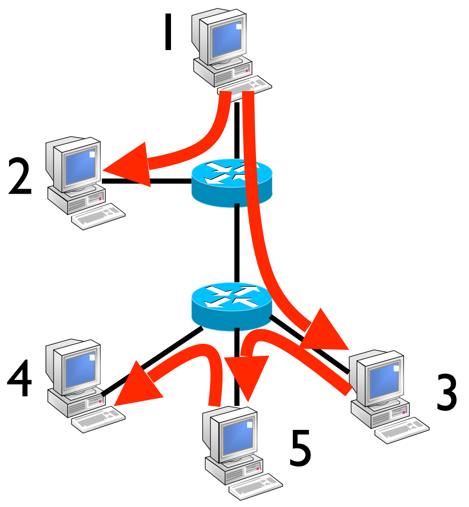
IP multicast overlay multicast N unicast



- How to construct an efficient overlay tree?

Eliminating Shared Bottlenecks



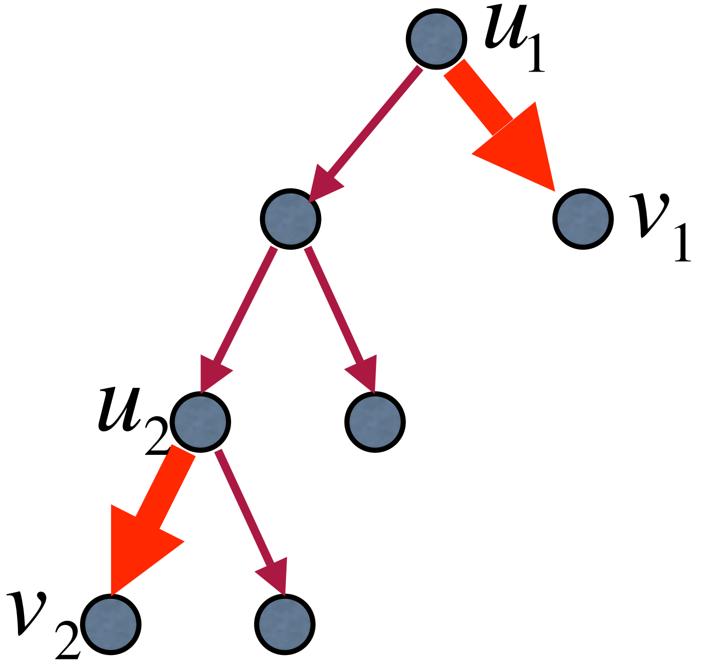


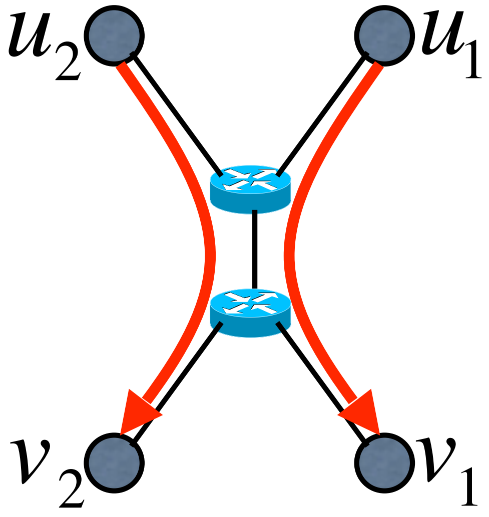
- □ Shared congestion detection
- □ Bottleneck Elimination
- □ Why difficult?

Outline

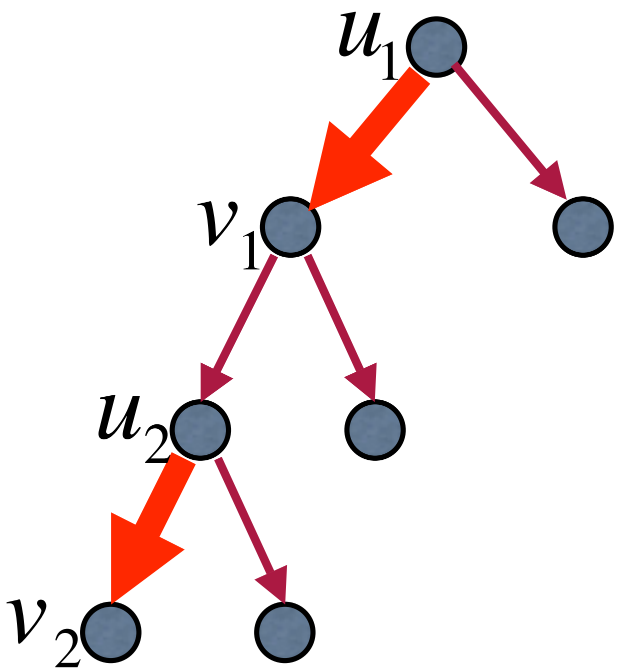
- ☑ Introduction
- □ Types of Shared Bottlenecks
- □ Bottleneck Elimination
- □ Performance Evaluation
- □ Summary

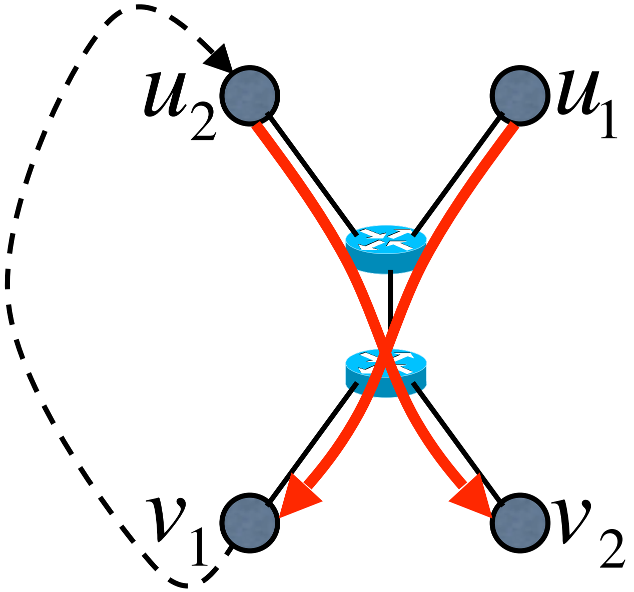
Inter-path Shared Bottleneck





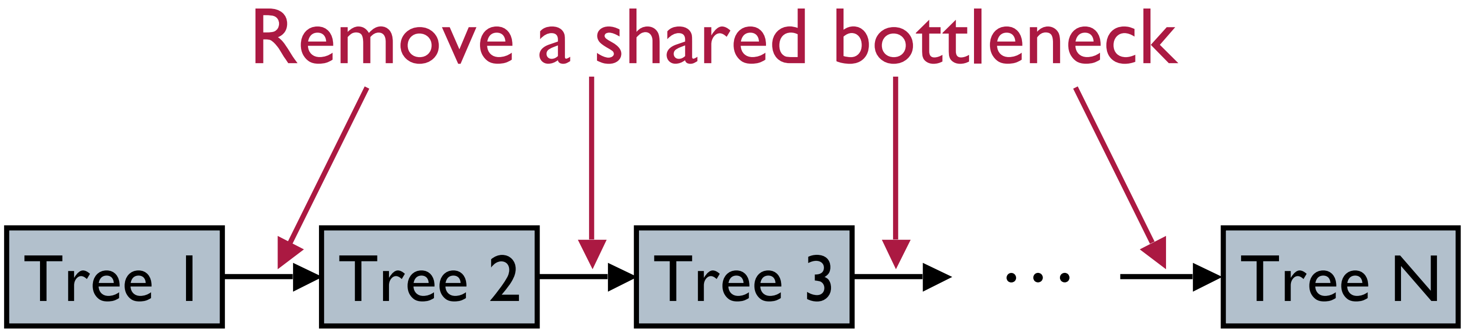
Intra-path Shared Bottleneck





Algorithm

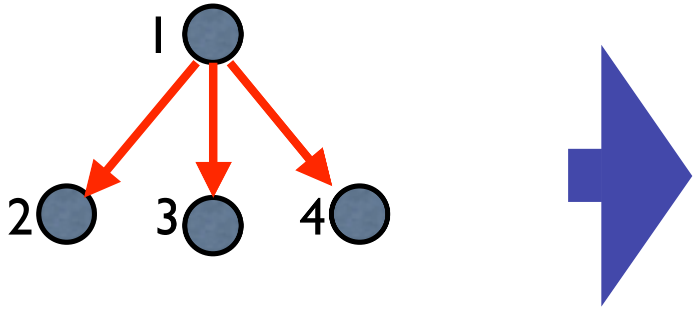
- Eliminate both types of shared bottlenecks
- No tree oscillation

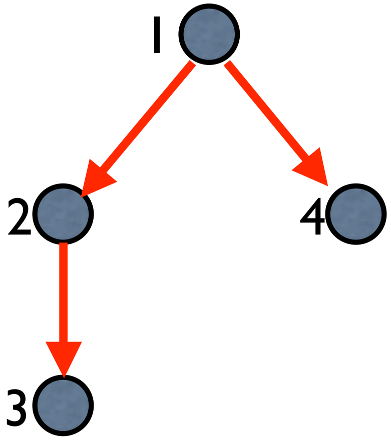


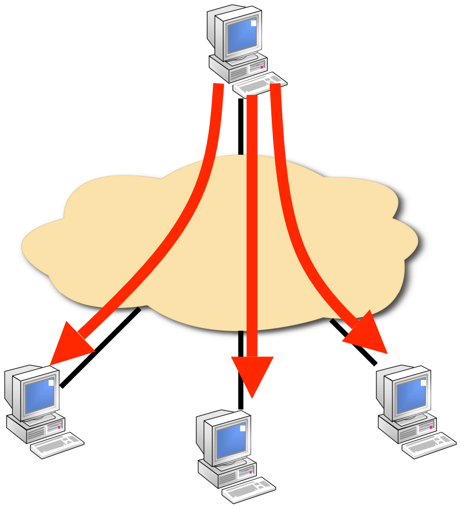
Remove a shared bottleneck

- One-way process
- Finite steps

Example





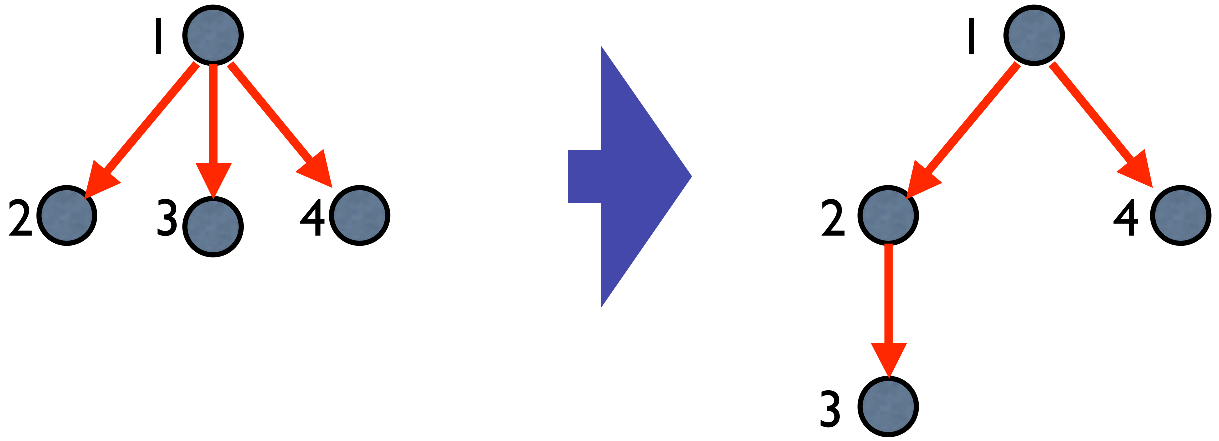


- □ # of leaves ↓
- □ Height ↑

Leaf Distance Vector

- □ Depths of all leaf nodes in descending

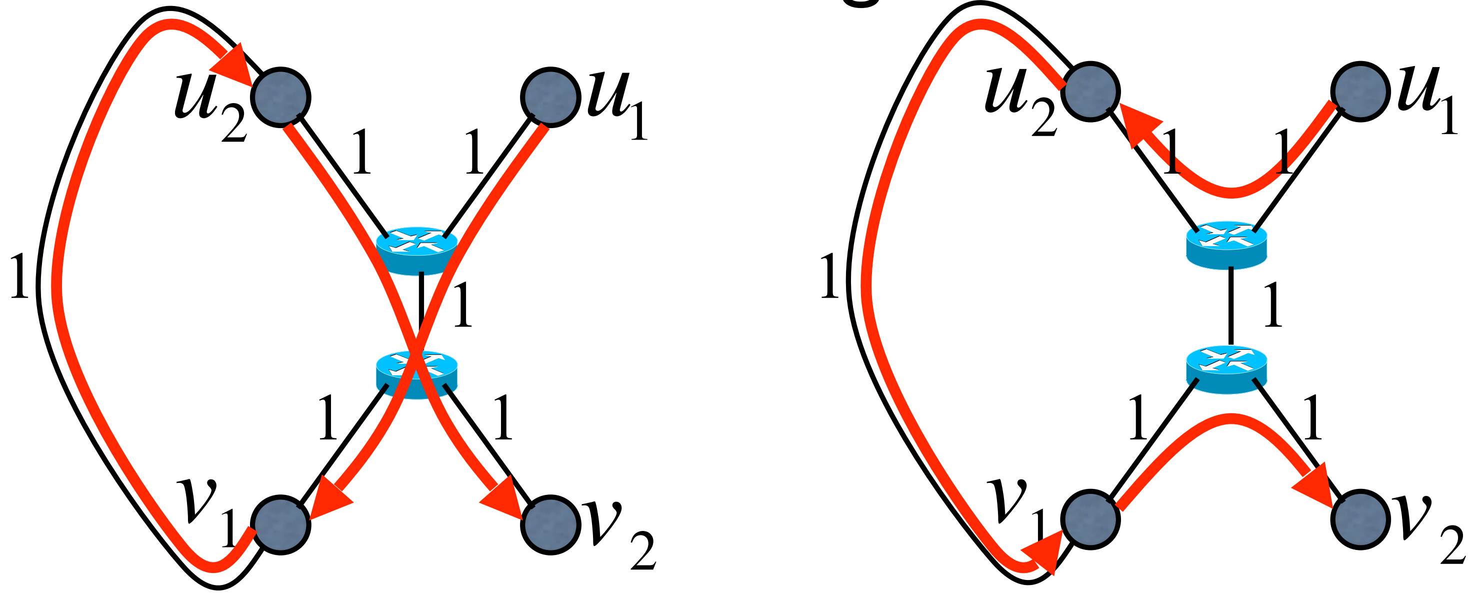
$$D \prec D' \Longleftrightarrow \begin{cases} |D| > |D'| \\ \quad \text{or} \\ |D| = |D'| \text{ and } D \text{ precedes } D' \text{ lexicographically} \end{cases}$$



$$D = (1,1,1) \quad \prec \quad D' = (2,1)$$

Total Cost

- □ Sum of costs of all edges in the tree



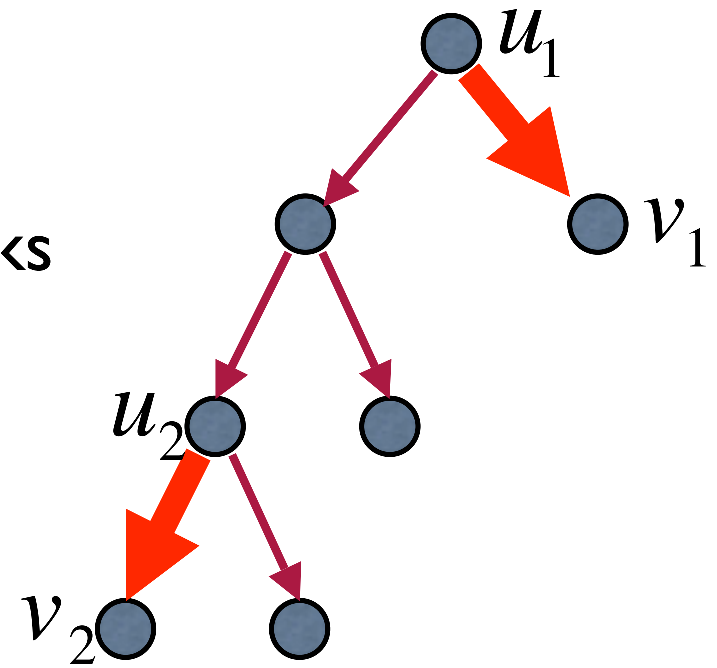
$$C = \underset{u_1v_1}{3} + \underset{v_1u_2}{1} + \underset{u_2v_2}{3} = 7 > C' = \underset{u_1u_2}{2} + \underset{u_2v_1}{1} + \underset{v_1v_2}{2} = 5$$

Proof Sketch

- Bottleneck elimination algorithms
  - Remove-Inter-Path-Shared-Bottleneck
  - Remove-Intra-Path-Shared-Bottleneck
- Applying either algorithm causes
  - D ↑
  - C ↓
- Only a finite # of changes

Protocol

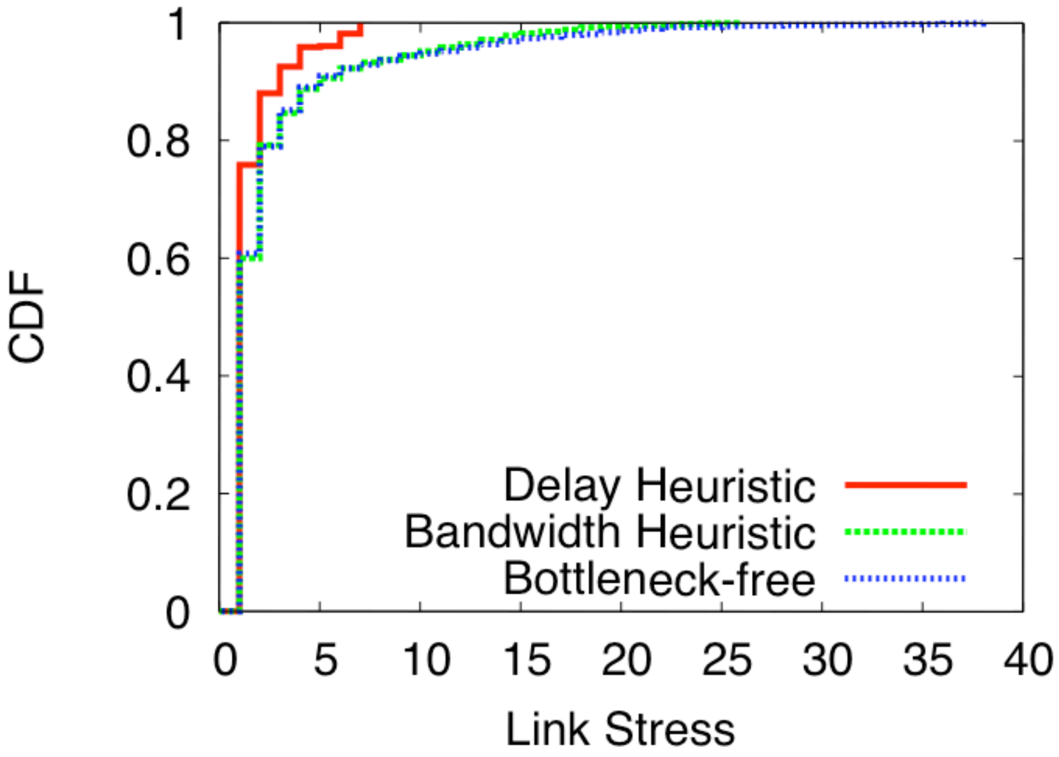
- □ Measure delay for congested edges
- □ Detect shared congestion
- □ Eliminate shared bottlenecks
- □ Forward remaining bottlenecks



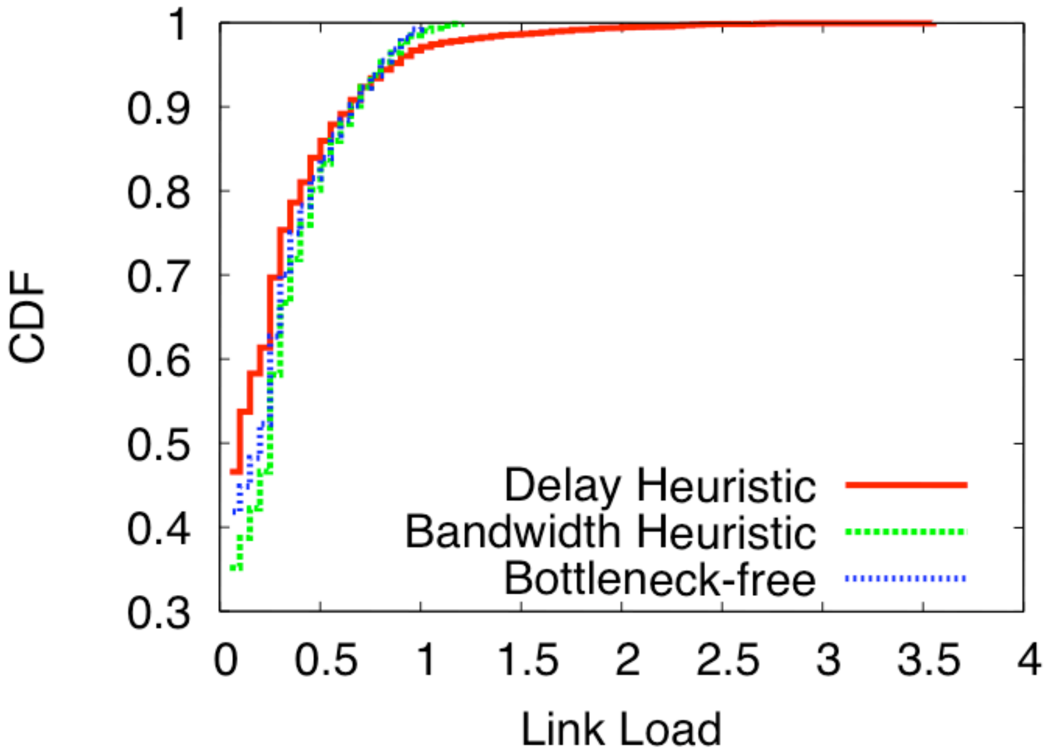
Performance Evaluation

- Comparison
  - Delay heuristic
  - Bandwidth heuristic
  - Bottleneck-free
- Metrics
  - Links stress
  - Link load
  - Relative delay penalty
  - Receiving rate

Link Stress

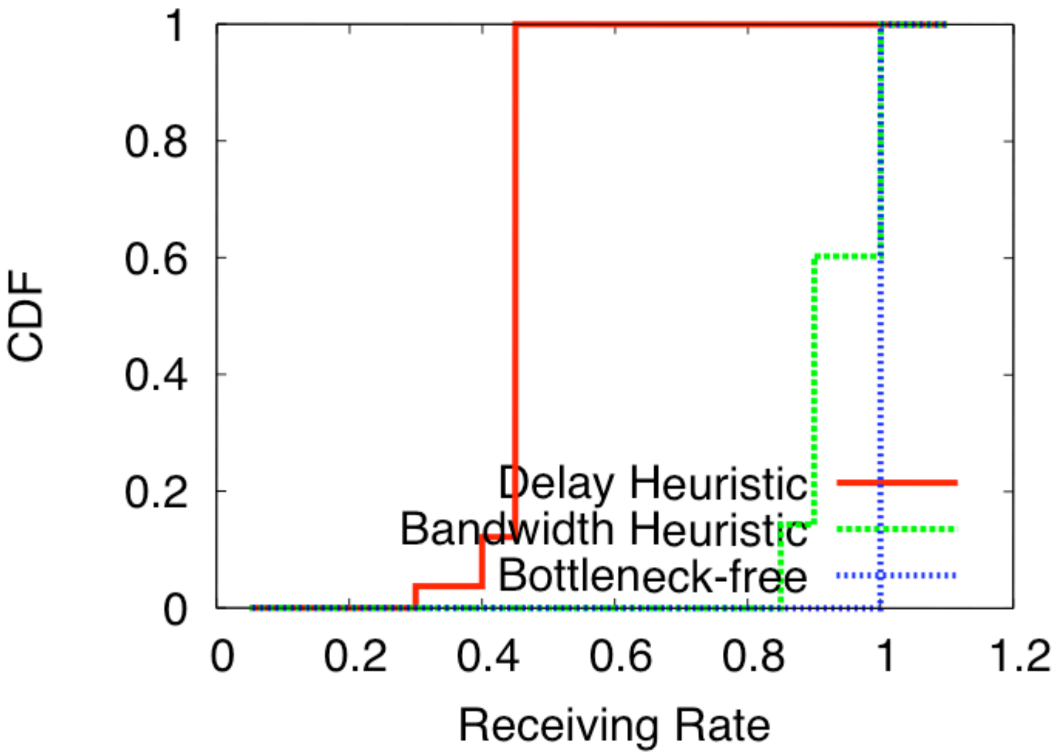

1
0.8
0.6
0.4
0.2
0
CDF
0
5
10
15
20
25
30
35
40
Link Stress
Delay Heuristic
Bandwidth Heuristic
Bottleneck-free

Link Load

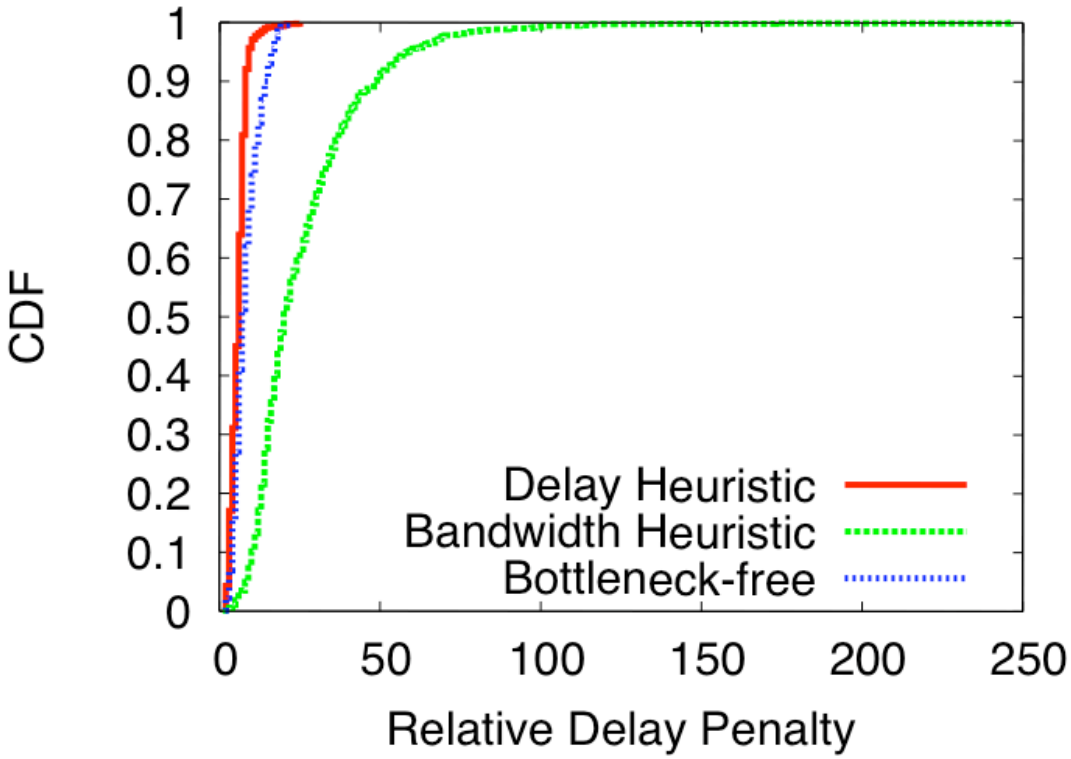


CDF
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0
0.5
1
1.5
2
2.5
3
3.5
4
Link Load
Delay Heuristic
Bandwidth Heuristic
Bottleneck-free

Receiving Rate



Relative Delay Penalty



Summary

- Overlay multicast creates shared bottlenecks
- Proposed tree construction algorithm
  - Eliminates all shared bottlenecks
  - Provides full receiving rate
  - Low link load, low delay