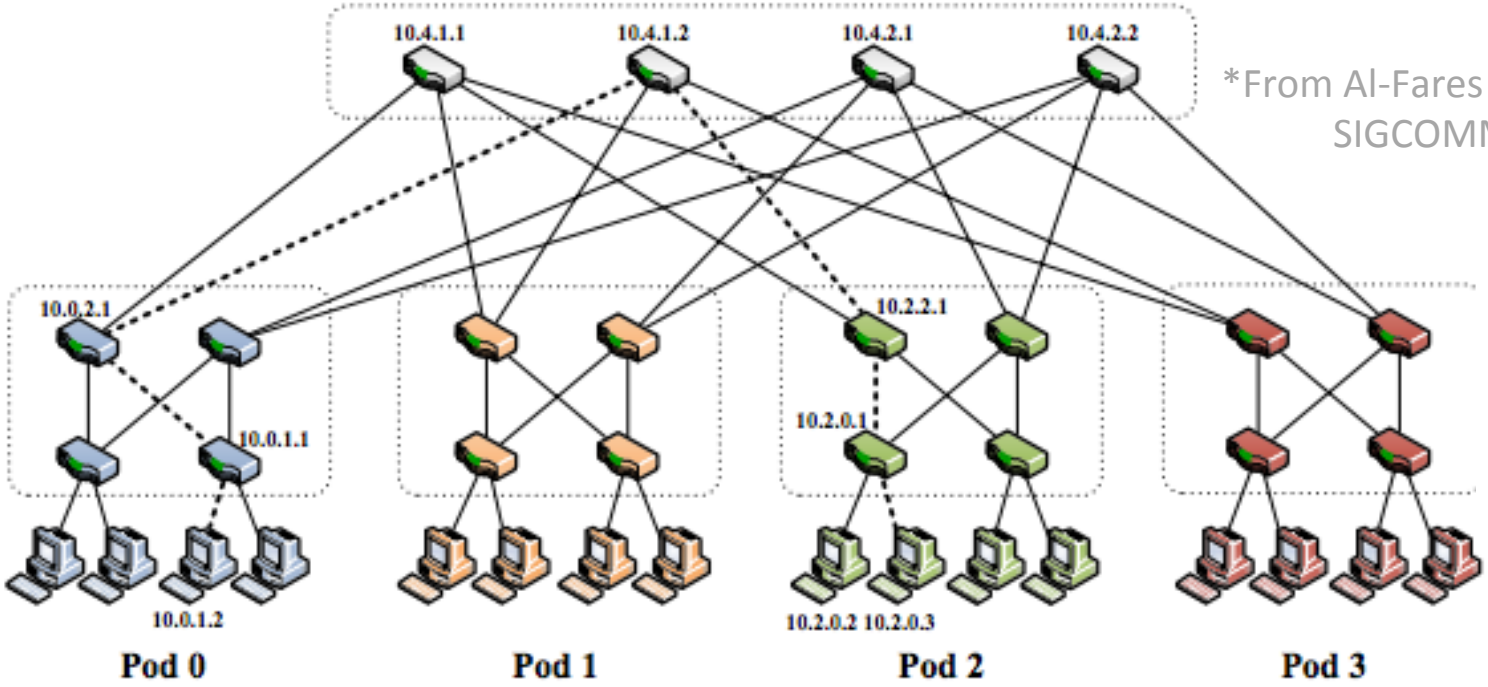
F10: A Fault-Tolerant Engineered Network

Vincent Liu, Daniel Halperin,
Arvind Krishnamurthy, Thomas Anderson
University of Washington

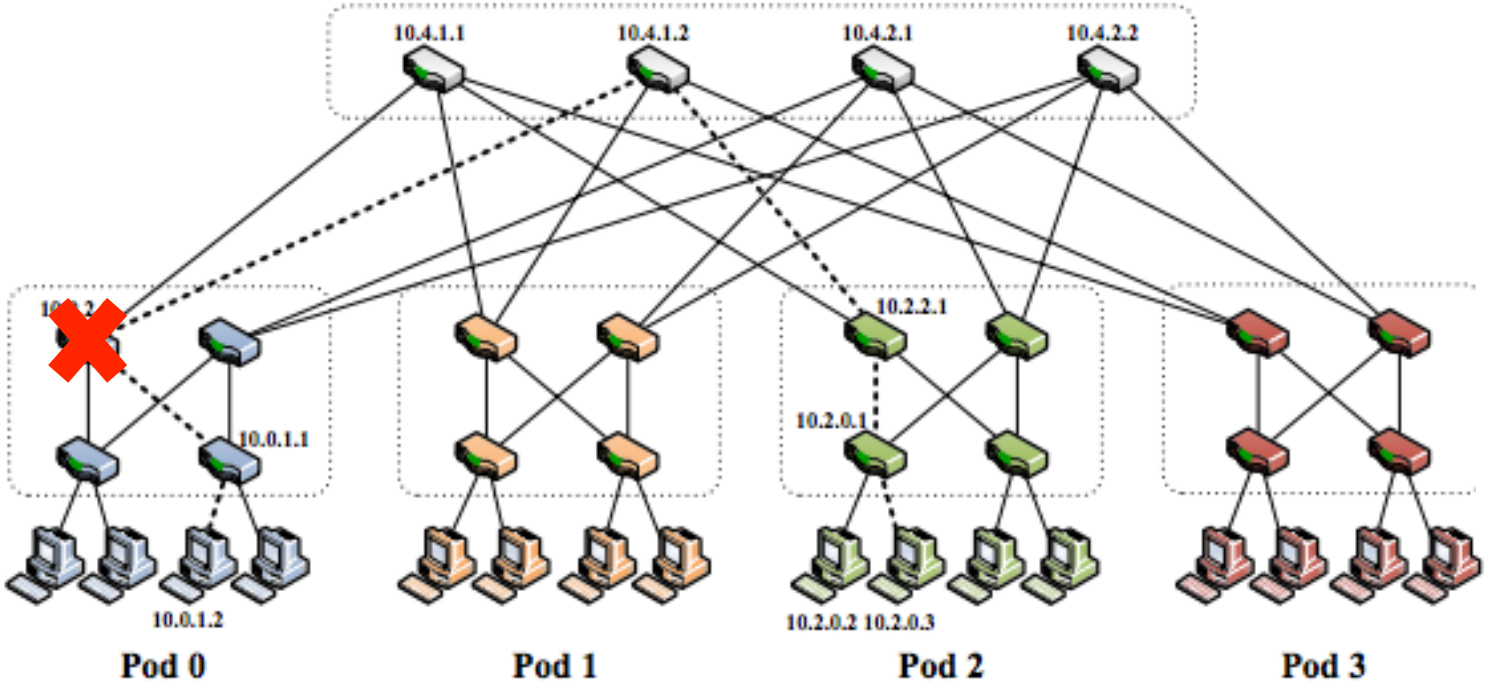
Today's Data Centers



*From Al-Fares et al. SIGCOMM '08

- Today's data centers are built using multi-rooted trees
- Commodity switches for cost, bisection bandwidth, and resilience to failures

FatTree Example: PortLand



- Heartbeats to detect failures
- Centralized controller installs updated routes
- Exploits path redundancy

Unsolved Issues with FatTrees

- **Slow Detection**
  - Commodity switches fail often
  - Not always sure they failed (gray/partial failures)
- **Slow Recovery**
  - Failure recovery is not local
  - Topology does not support local reroutes
- **Suboptimal Flow Assignment**
  - Failures result in an unbalanced tree
  - Loses load balancing properties

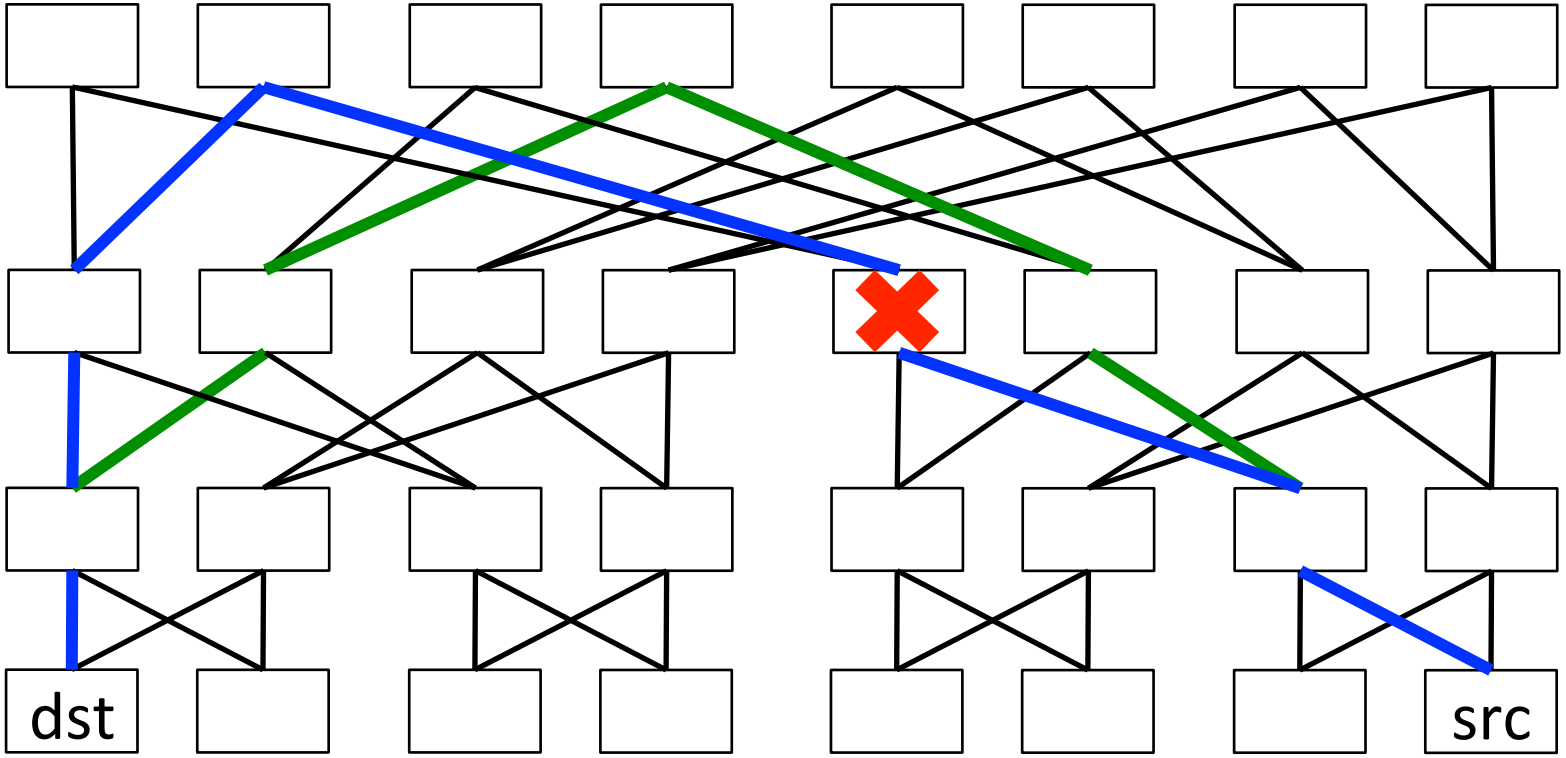
F10

- Co-design of topology, routing protocols and failure detector
  - Novel topology that enables local, fast recovery
  - Cascading protocols for optimal recovery
  - Fine-grained failure detector for fast detection
- Same # of switches/links as FatTrees

Outline

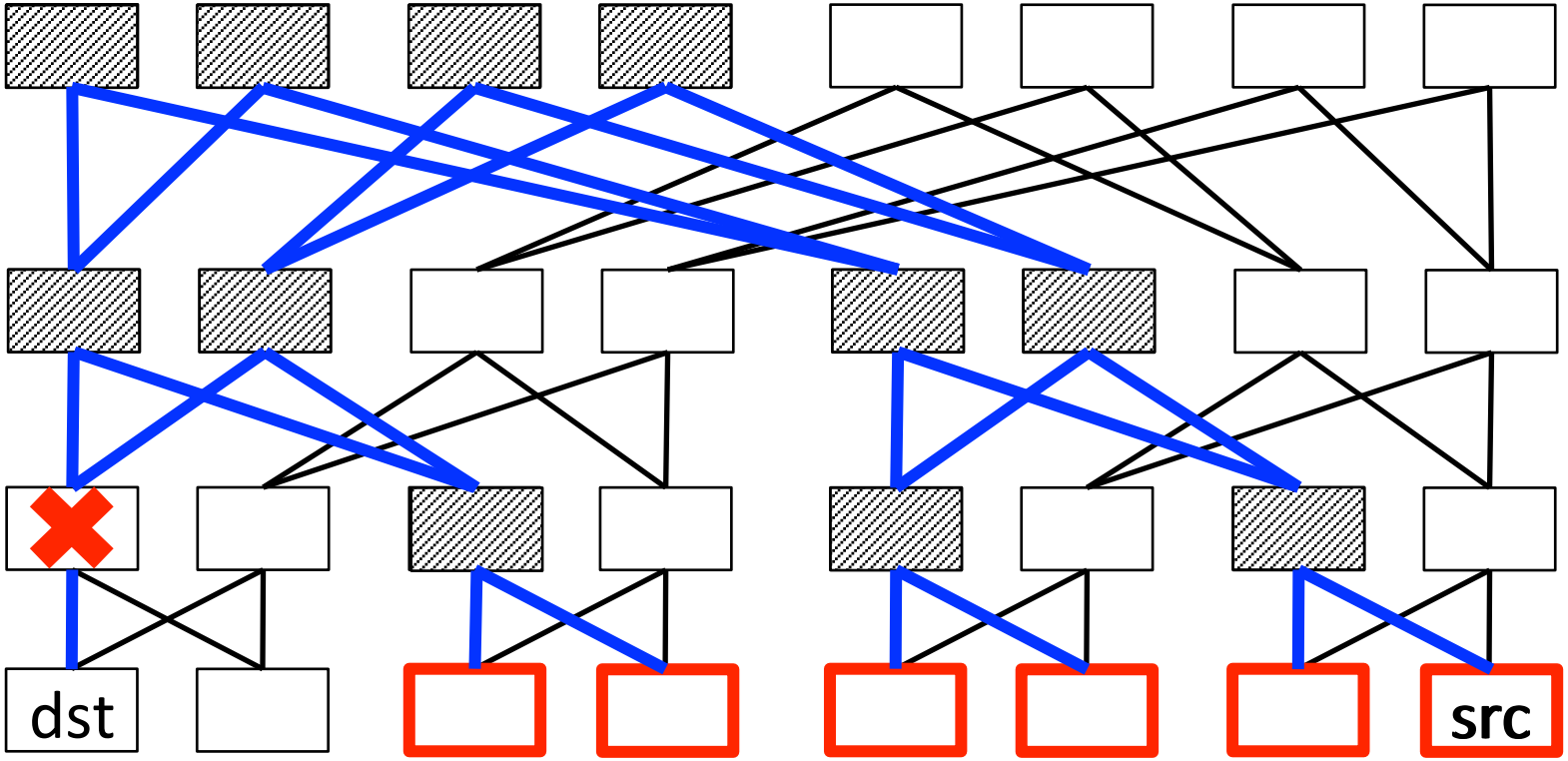
- Motivation & Approach
- Topology: AB FatTree
- Cascaded Failover Protocols
- Failure Detection
- Evaluation
- Conclusion

Why is FatTree Recovery Slow?



- Lots of redundancy on the upward path
- Immediately restore connectivity at the point of failure

Why is FatTree Recovery Slow?



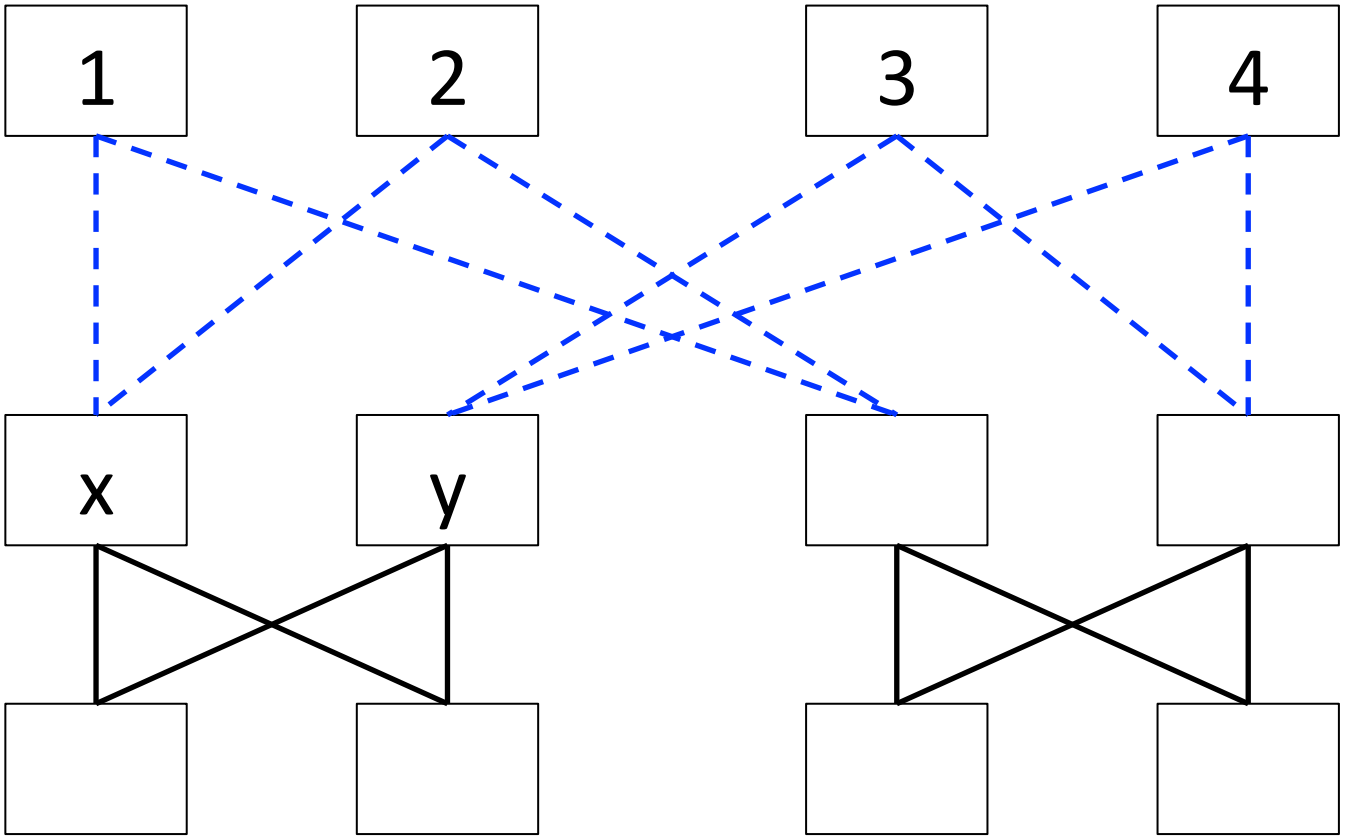
- No redundancy on the way down
- Alternatives are many hops away

No direct path

Has alternate path

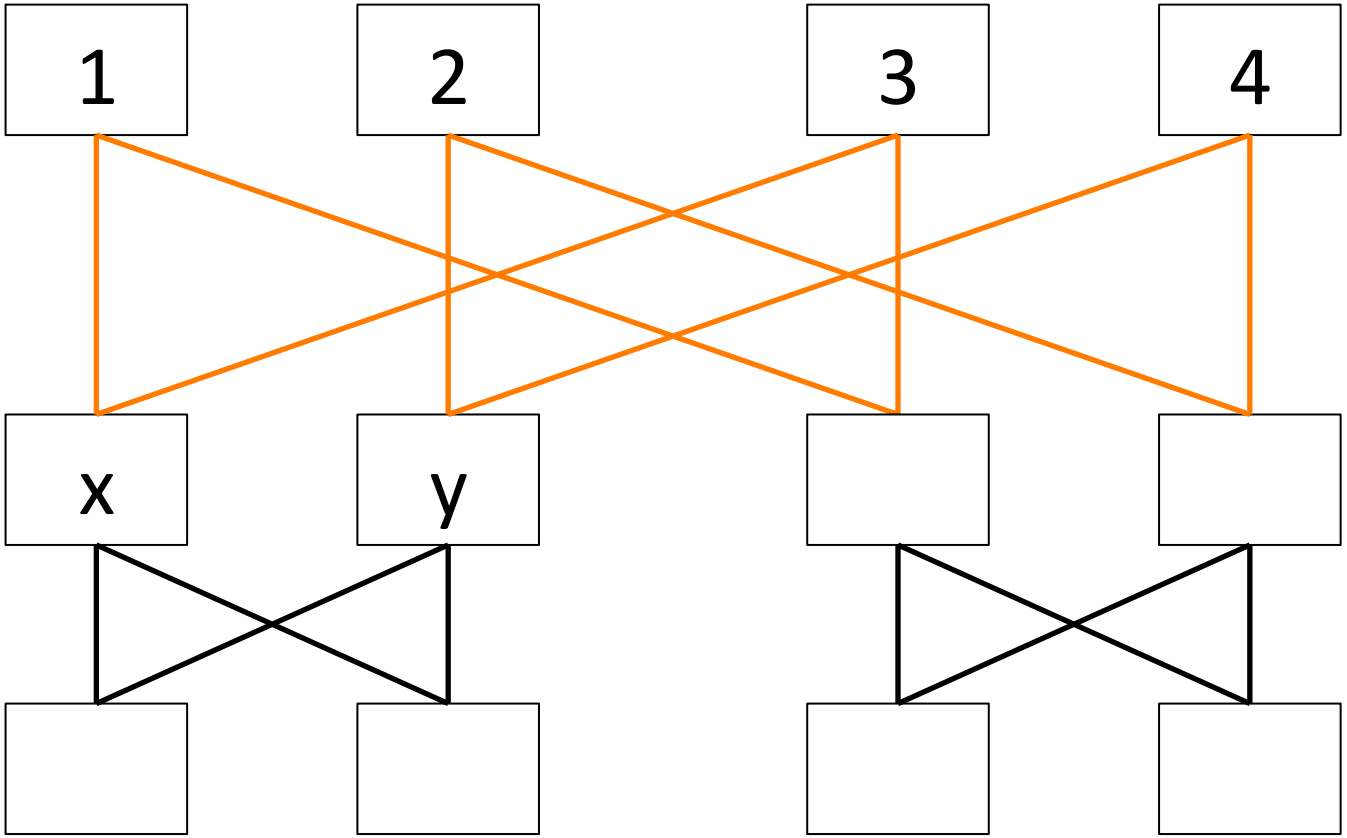
Type A Subtree

Consecutive Parents

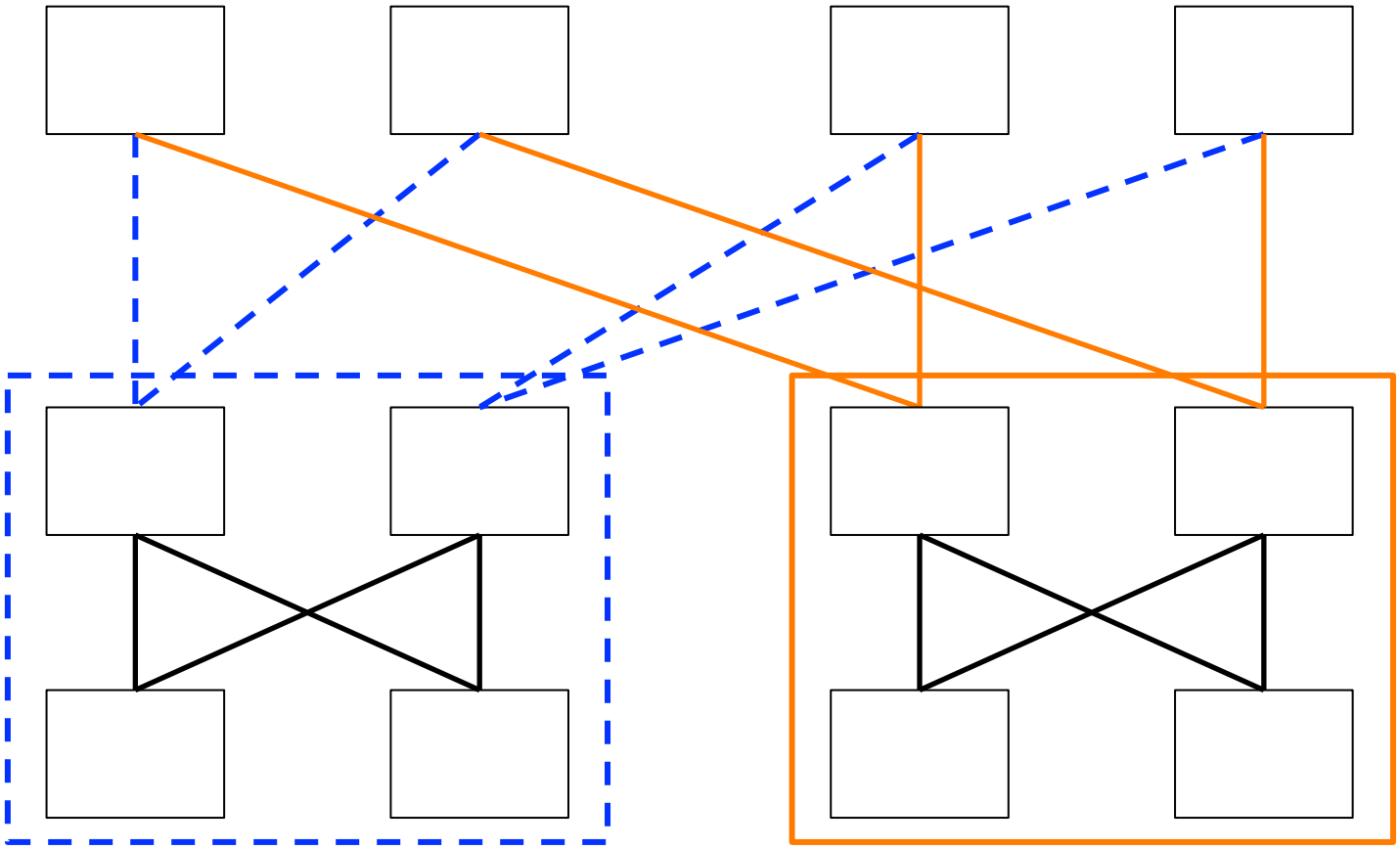


Type B Subtree

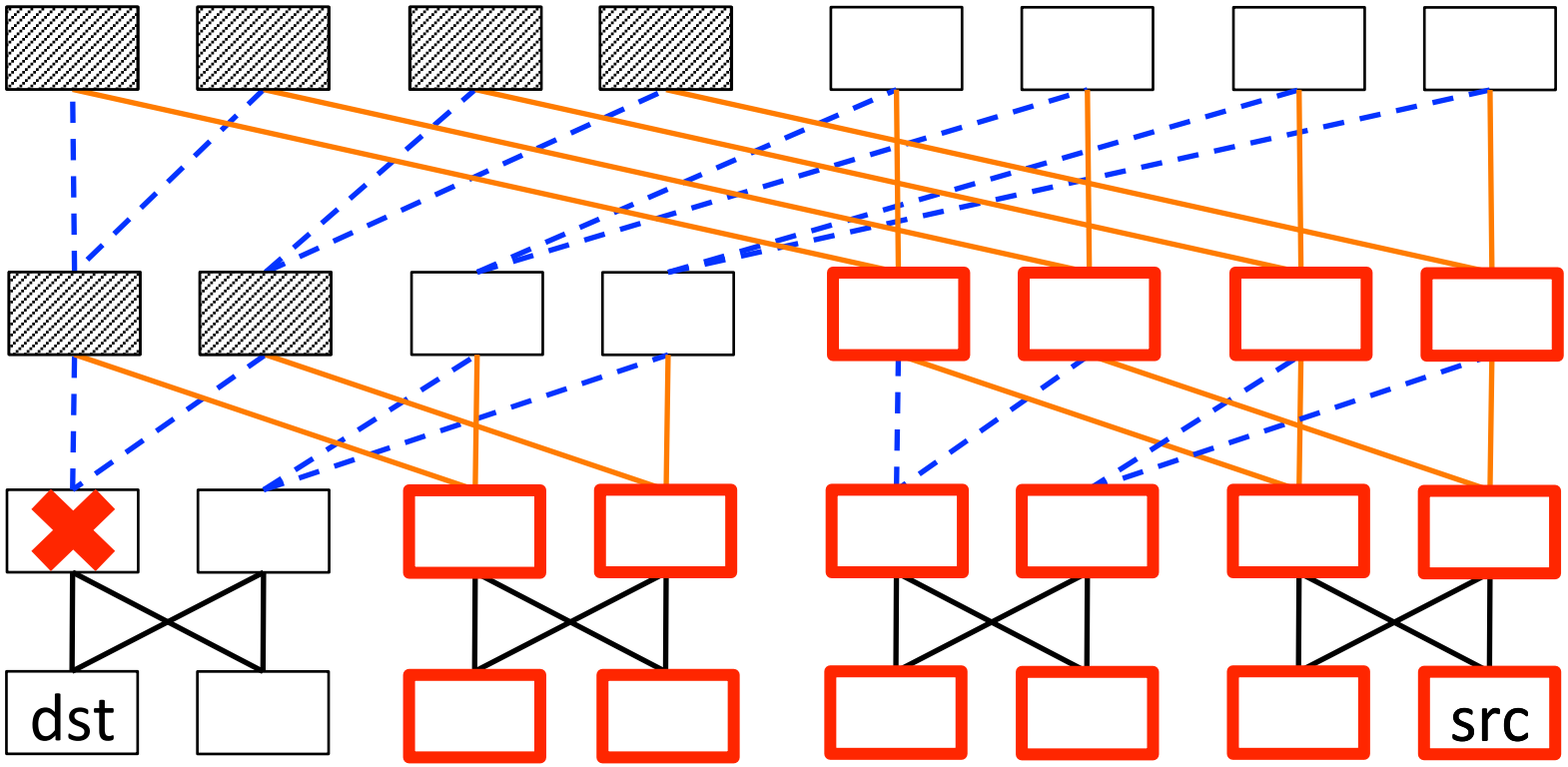
Strided Parents



AB FatTree



Alternatives in AB FatTrees



- More nodes have alternative, direct paths
- One hop away from node with an alternative

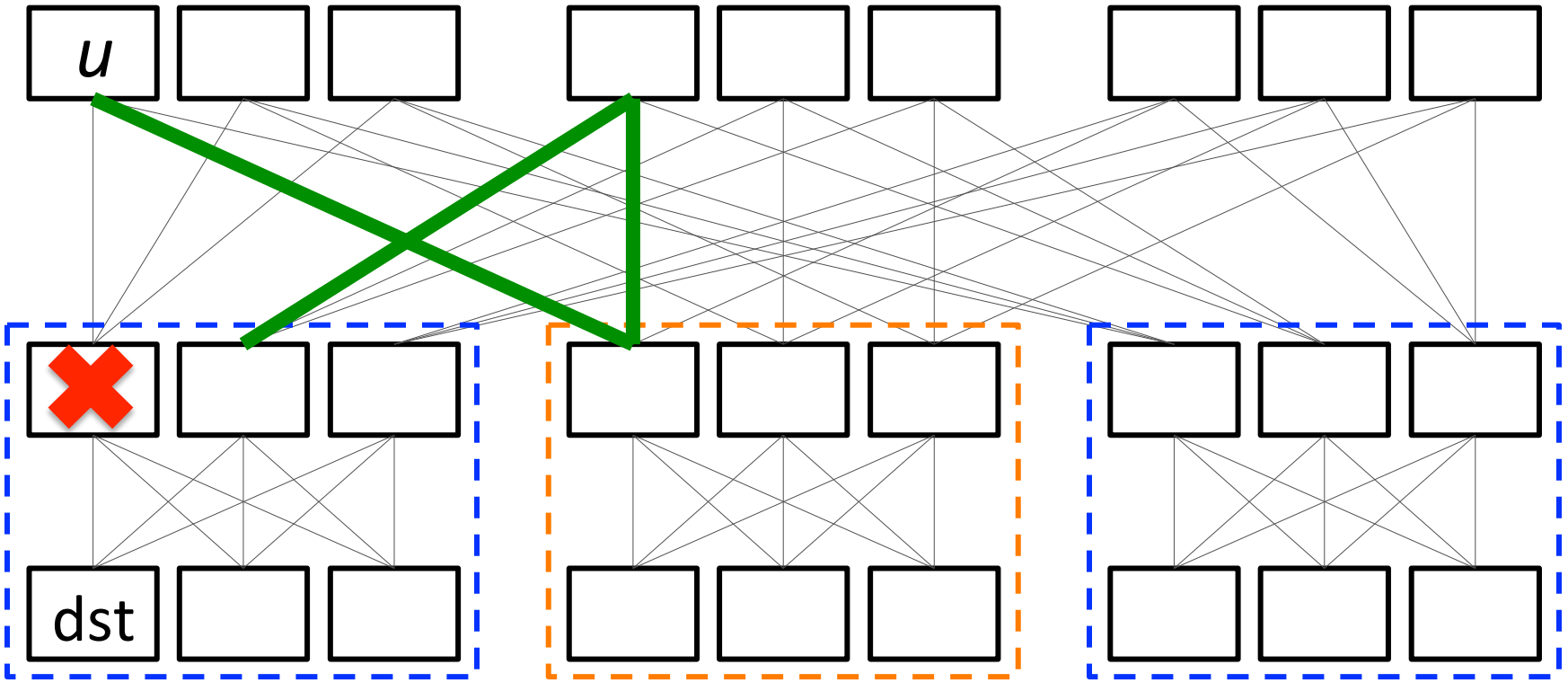
No direct path

Has alternate path

Cascaded Failover Protocols

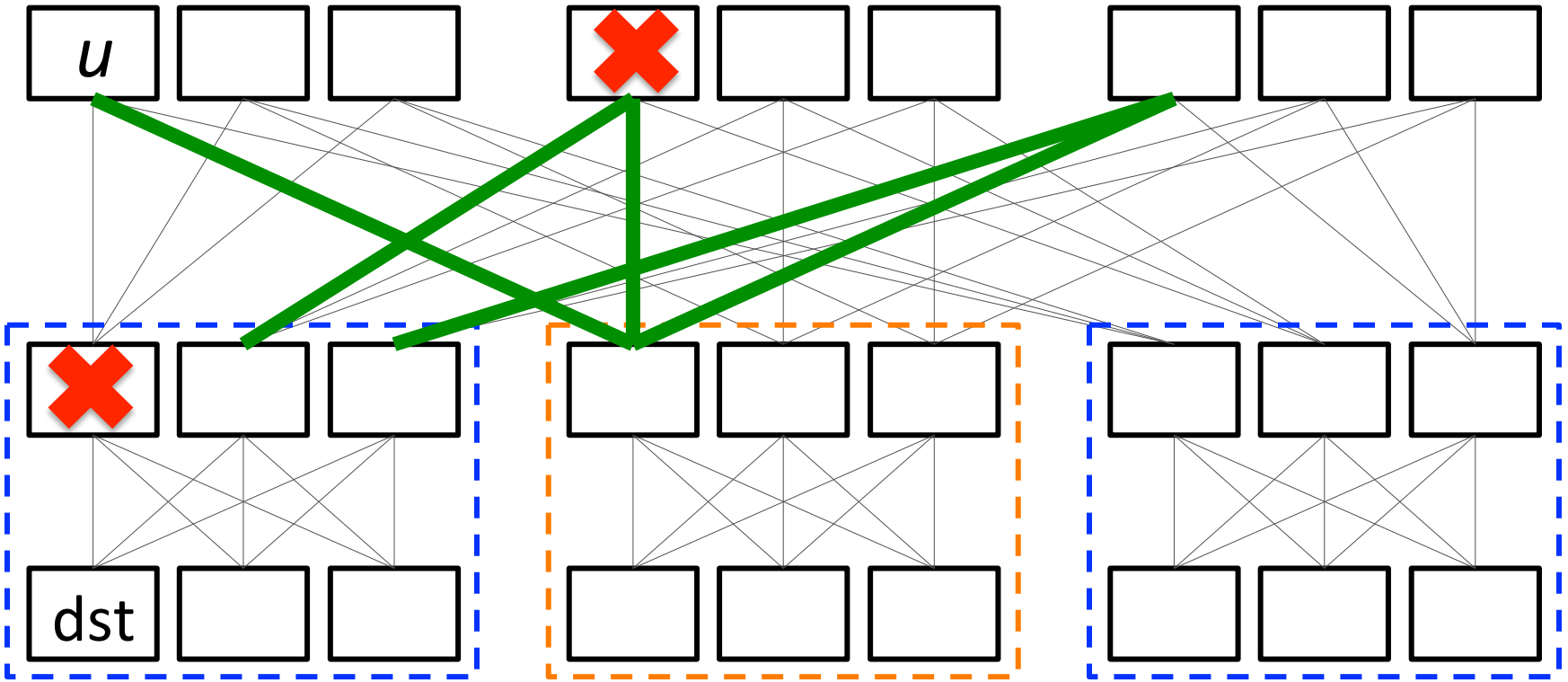
- A local rerouting mechanism
  - Immediate restoration
- A pushback notification scheme
  - Restore direct paths
- An epoch-based centralized scheduler
  - globally re-optimizes traffic

Local Rerouting



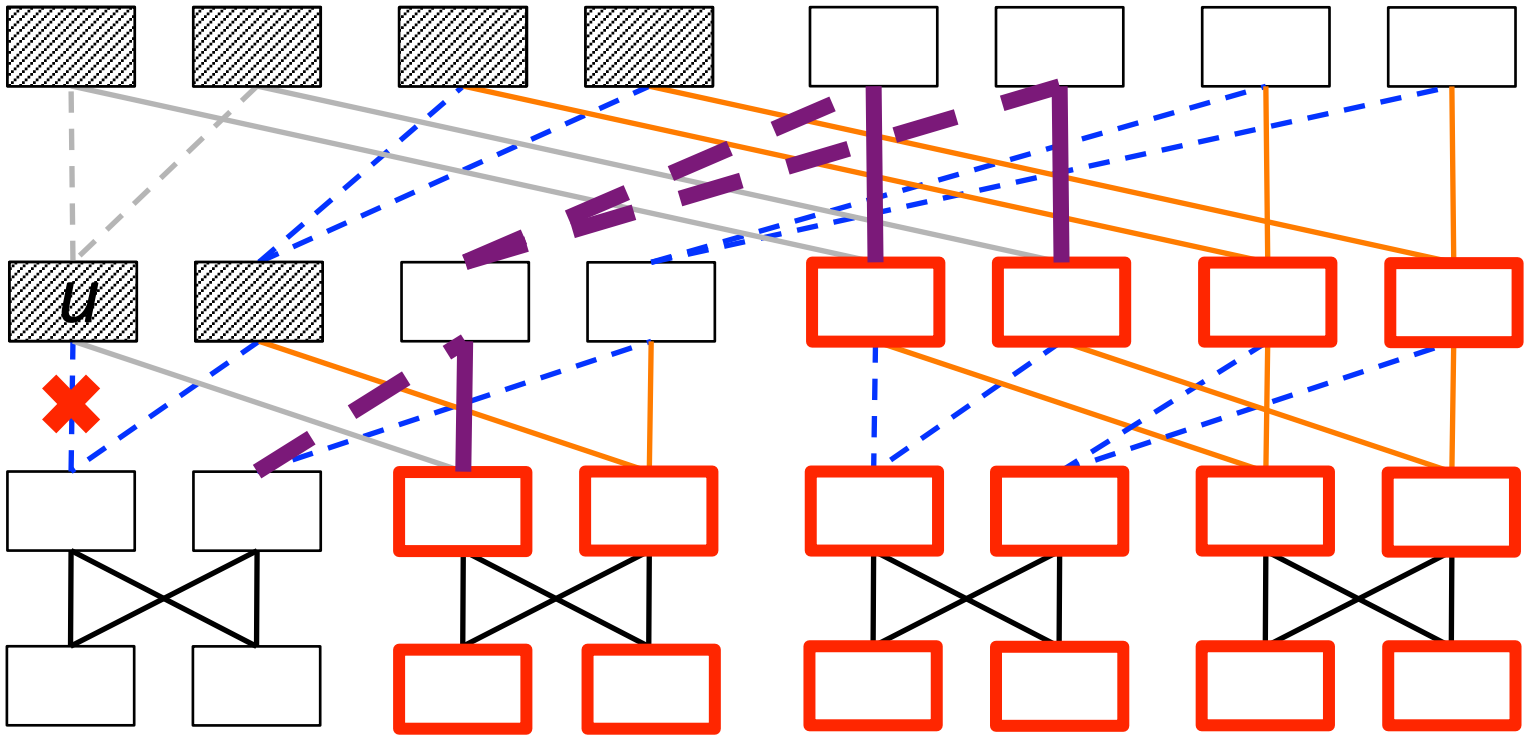
- Route to a sibling in an opposite-type subtree
- Immediate, local rerouting around the failure

Local Rerouting – Multiple Failures



- Resilient to multiple failures, refer to paper
- Increased load and path dilation

Pushback Notification



- Detecting switch broadcasts notification
- Restores direct paths, but not finished yet

No direct path

Has alternate path

Centralized Scheduler

- Related to existing work (Hedera, MicroTE)
- Gather traffic matrices
- Place **long-lived** flows based on their size
- Place **shorter** flows with weighted ECMP

Outline

- Motivation & Approach
- Topology: AB FatTree
- Cascaded Failover Protocols
- Failure Detection
- Evaluation
- Conclusion

Why are Today's Detectors Slow?

- Based on loss of multiple heartbeats
  - Detector is separated from failure
- Slow because:
  - Congestion
  - Gray failures
  - Don't want to waste too many resources

F10 Failure Detector

- Look at the link itself
  - Send traffic to physical neighbors when idle
  - Monitor incoming bit transitions and packets
  - Stop sending and reroute the very next packet
- Can be fast because rerouting is cheap

Outline

- Motivation & Approach
- Topology: AB FatTree
- Cascaded Failover Protocols
- Failure Detection
- Evaluation
- Conclusion

Evaluation

1. Can F10 reroute quickly?

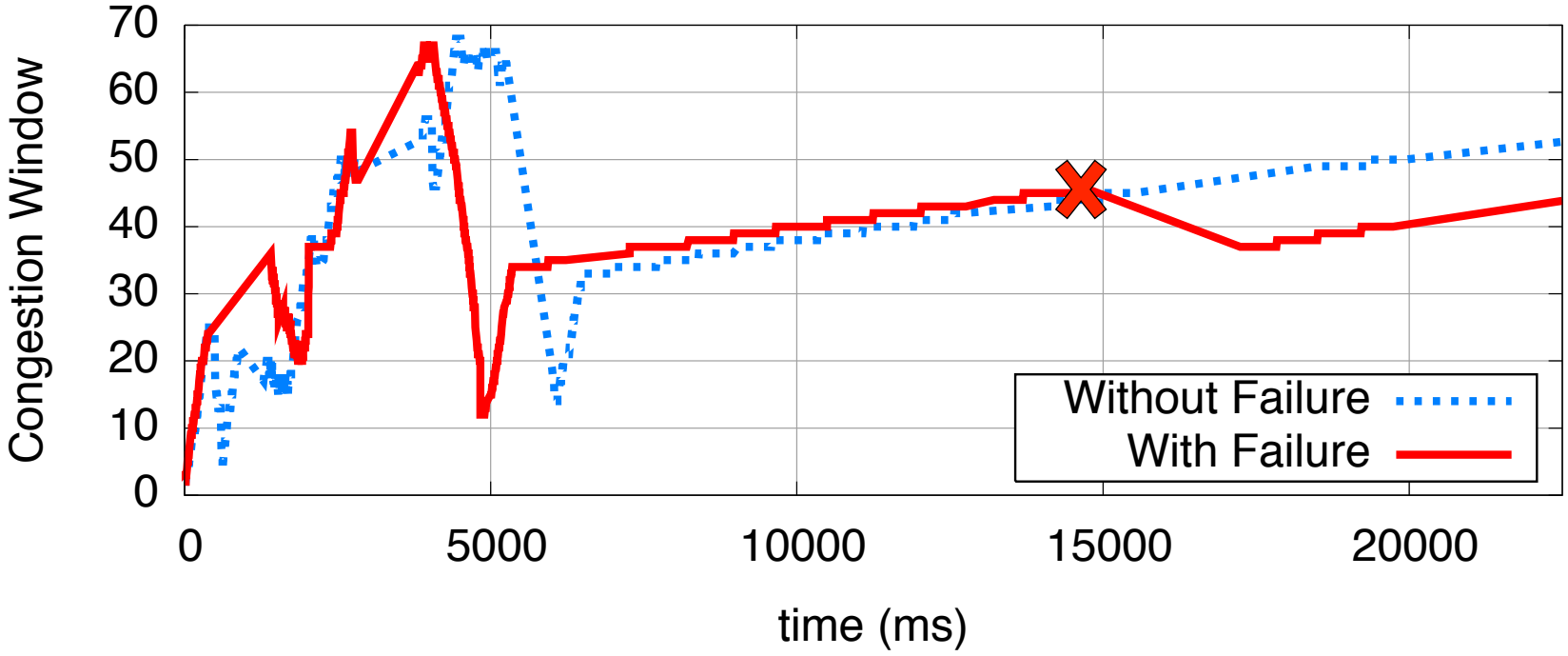
2. Can F10 avoid congestion loss that results from failures?

3. How much does this effect application performance?

Methodology

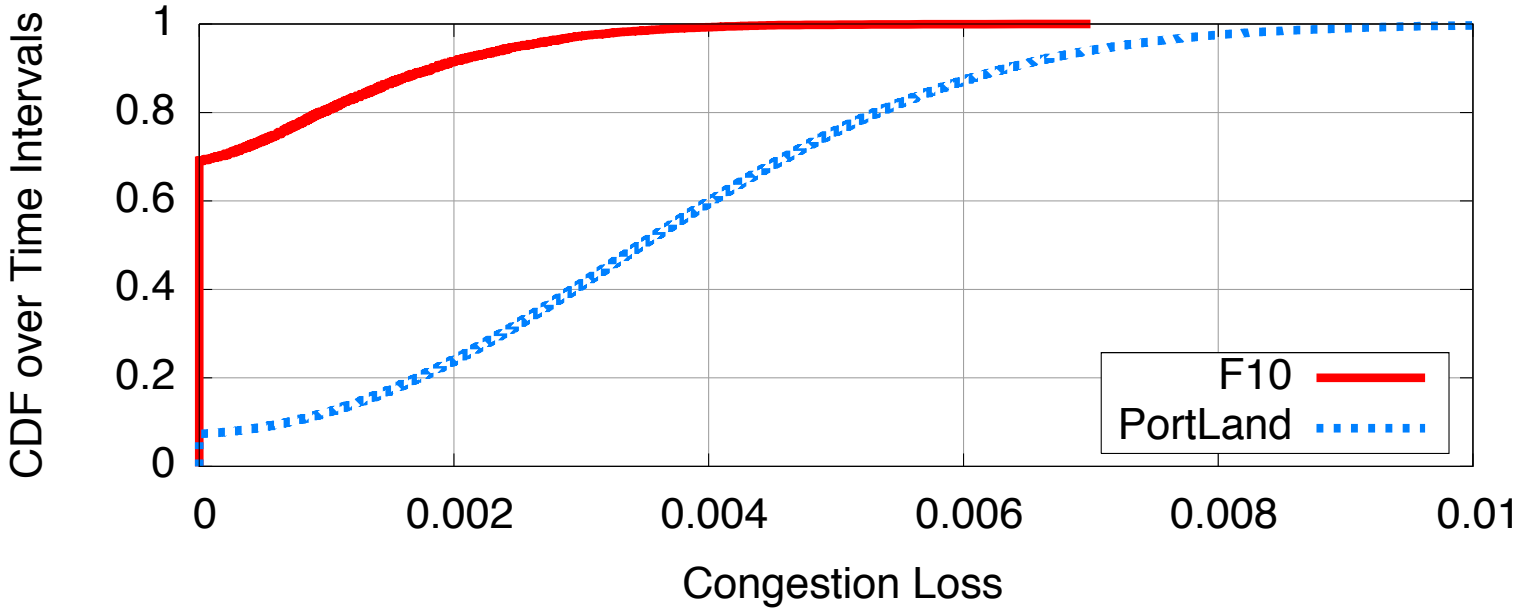
- Testbed
  - Emulab w/ Click implementation
  - Used smaller packets to account for slower speed
- Packet-level simulator
  - 24-port 10GbE switches, 3 levels
  - Traffic model from Benson et al. IMC 2010
  - Failure model from Gill et al. SIGCOMM 2011
  - Validated using testbed

F10 Can Reroute Quickly



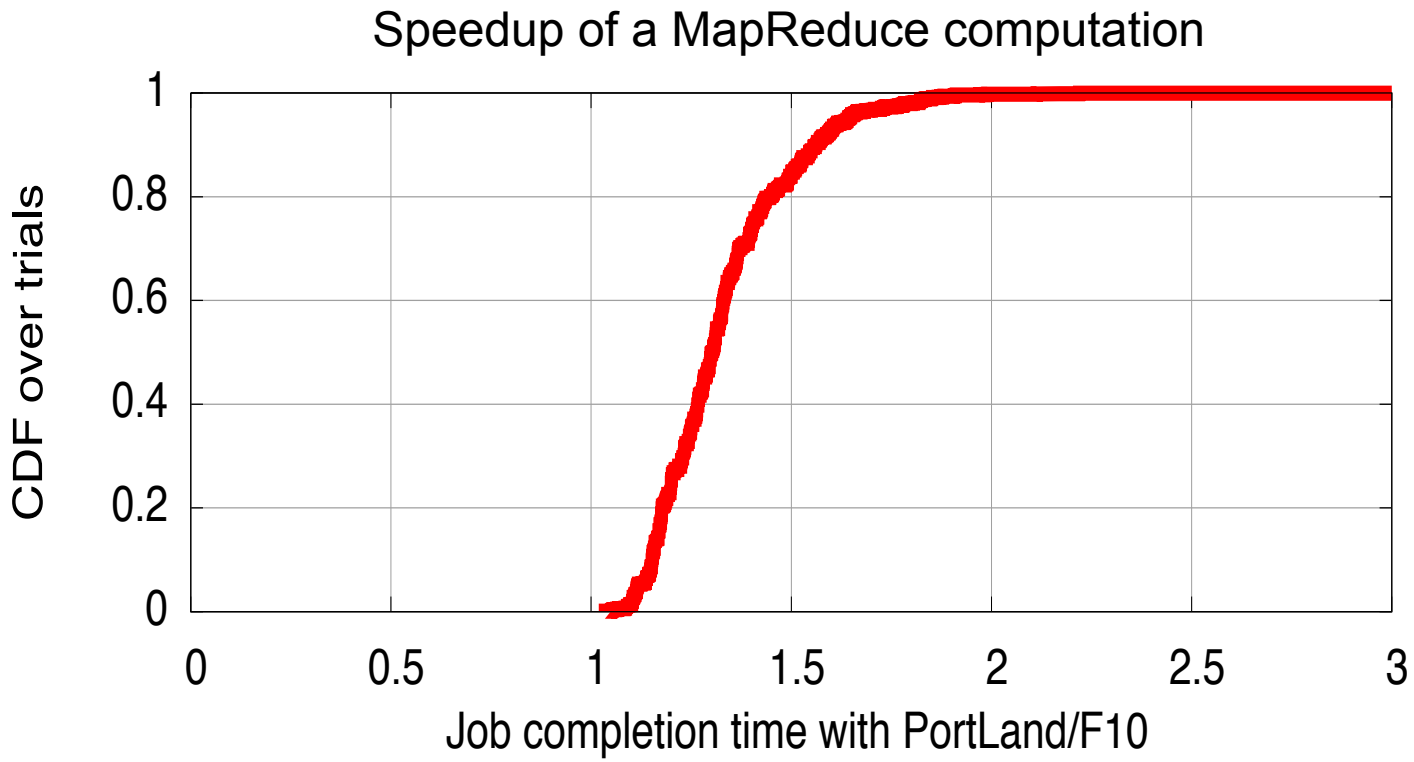
- F10 can recover from failures in under a millisecond
- Much less time than a TCP timeout

F10 Can Avoid Congestion Loss



PortLand has 7.6x the congestion loss of F10 under realistic traffic and failure conditions

F10 Improves App Performance



Median speedup is 1.3x

Conclusion

- F10 is a co-design of topology, routing protocols, and failure detector:
  - AB FatTrees to allow local recovery and increase path diversity
  - Pushback and global re-optimization restore congestion-free operation
- Significant benefit to application performance on typical workloads and failure conditions

- Thanks!