Exploiting Route Redundancy via Structured Peer to Peer Overlays

Ben Y. Zhao, Ling Huang, Jeremy Stribling, Anthony D. Joseph, and John D. Kubiatowicz

University of California, Berkeley

ICNP 2003

Challenges Facing Network Applications

- Network connectivity is not reliable
  - Disconnections frequent in the wide-area Internet
  - IP-level repair is slow
    - Wide-area: BGP ≈ 3 mins
    - Local-area: IS-IS ≈ 5 seconds
- Next generation network applications
  - Mostly wide-area
  - Streaming media, VoIP, B2B transactions
  - Low tolerance of delay, jitter and faults
  - Our work: transparent resilient routing infrastructure that adapts to faults in not seconds, but milliseconds

Talk Overview

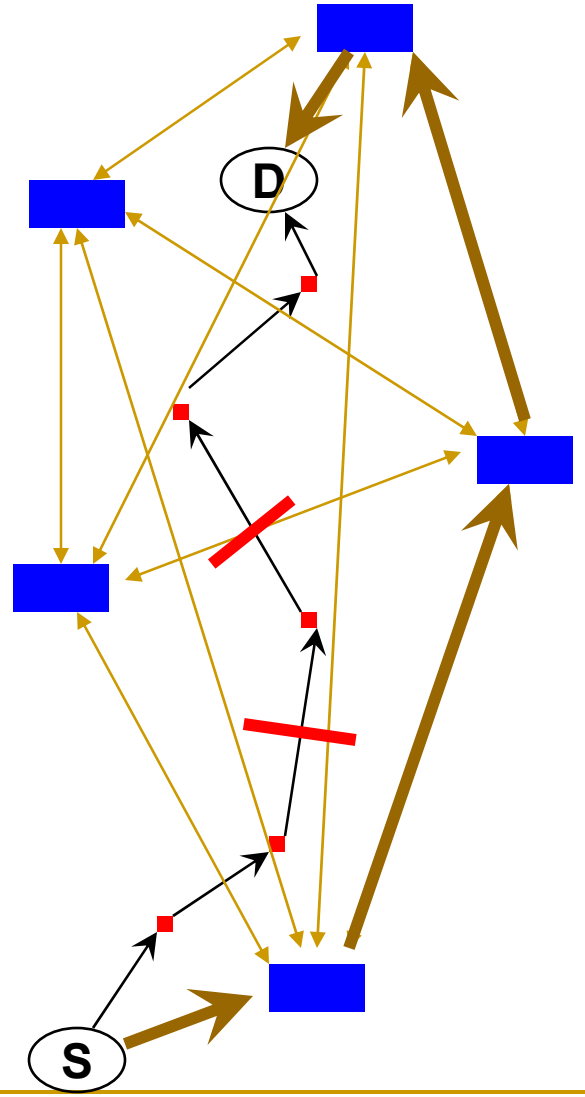
- Motivation
- Why structured routing
- Structured Peer to Peer overlays
- Mechanisms and policy
- Evaluation
- Summary

Routing in "Mesh-like" Networks

- Previous work has shown reasons for long convergence [Labovitz00, Labovitz01]
- *MinRouteAdver* timer
  - Necessary to aggregate updates from all neighbors
    - Commonly set to 30 seconds
  - Contributes to lower bound of BGP convergence time
- Internet becoming more mesh-like [Kaat99,labovitz99]
  - Worsens BGP convergence behavior
- Question
  - Can convergence be faster in context of structured routing?

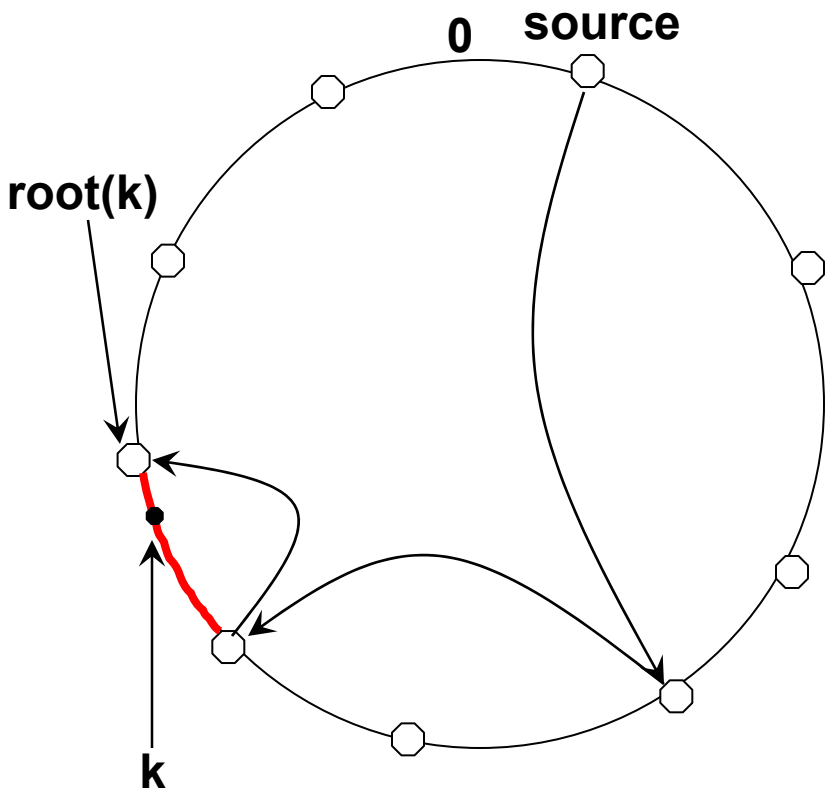
Resilient Overlay Networks (MIT)

- Fully connected mesh
- Allows each node full knowledge of network
  - Fast, independent calculation of routes
  - Nodes can construct any path, maximum flexibility
- Cost of flexibility
  - Protocol needs to choose the "right" route/nodes
  - Per node *O(n)* state
    - Monitors n - 1 paths
  - *O(n²)* total path monitoring is expensive



Leveraging Structured Peer-to-Peer Overlays

- Key based routing (IPTPS 03)
  - Large sparse ID space *N* (160 bits: 0 – 2^{160})
  - Nodes in overlay network have nodeIDs $\in N$
  - Given some key $k \in N$, overlay deterministically maps *k* to its ***root*** node (live node in the network)
  - route message to *root* (*k*)

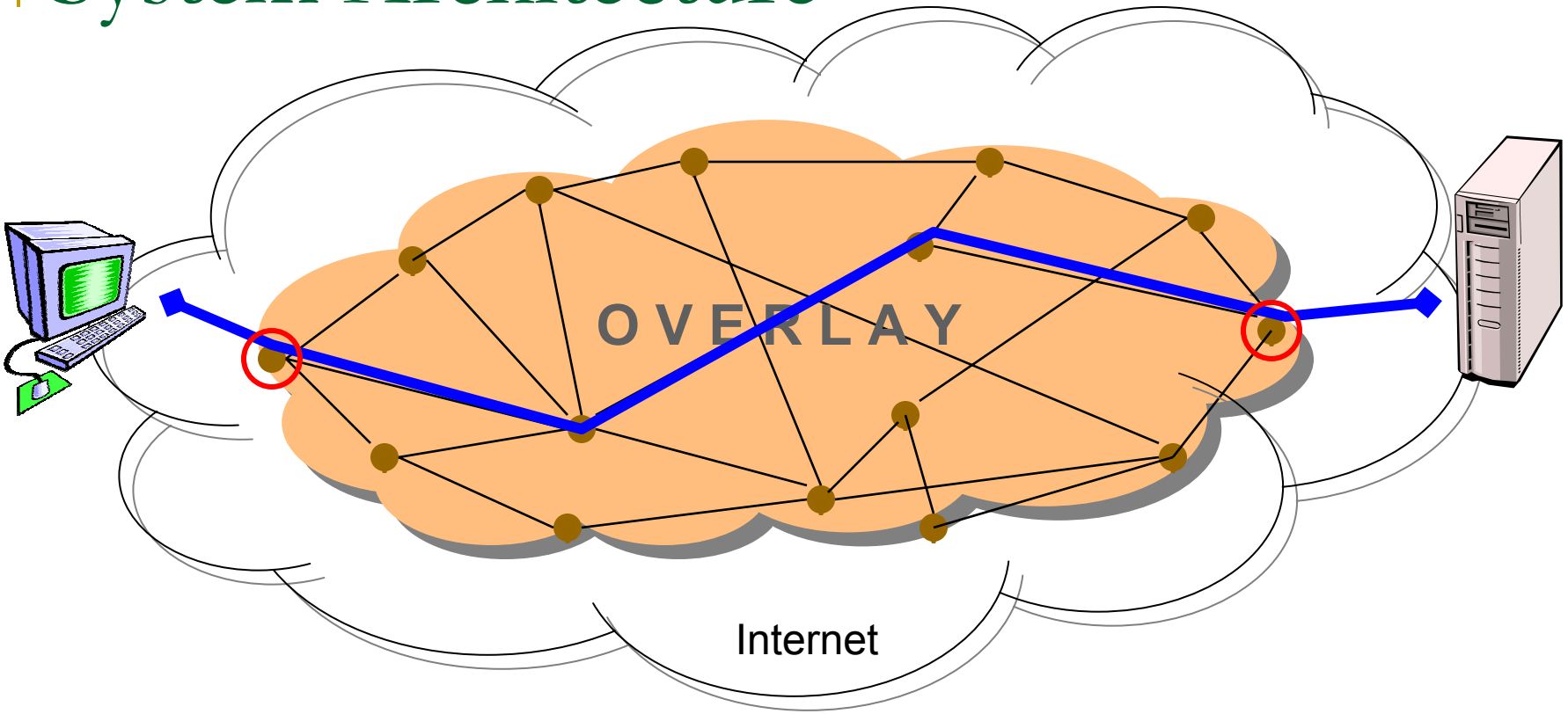


- Distributed Hashtables (DHT) is interface on KBR
  - Key is leveraging underlying routing mesh

Proximity Neighbor Selection

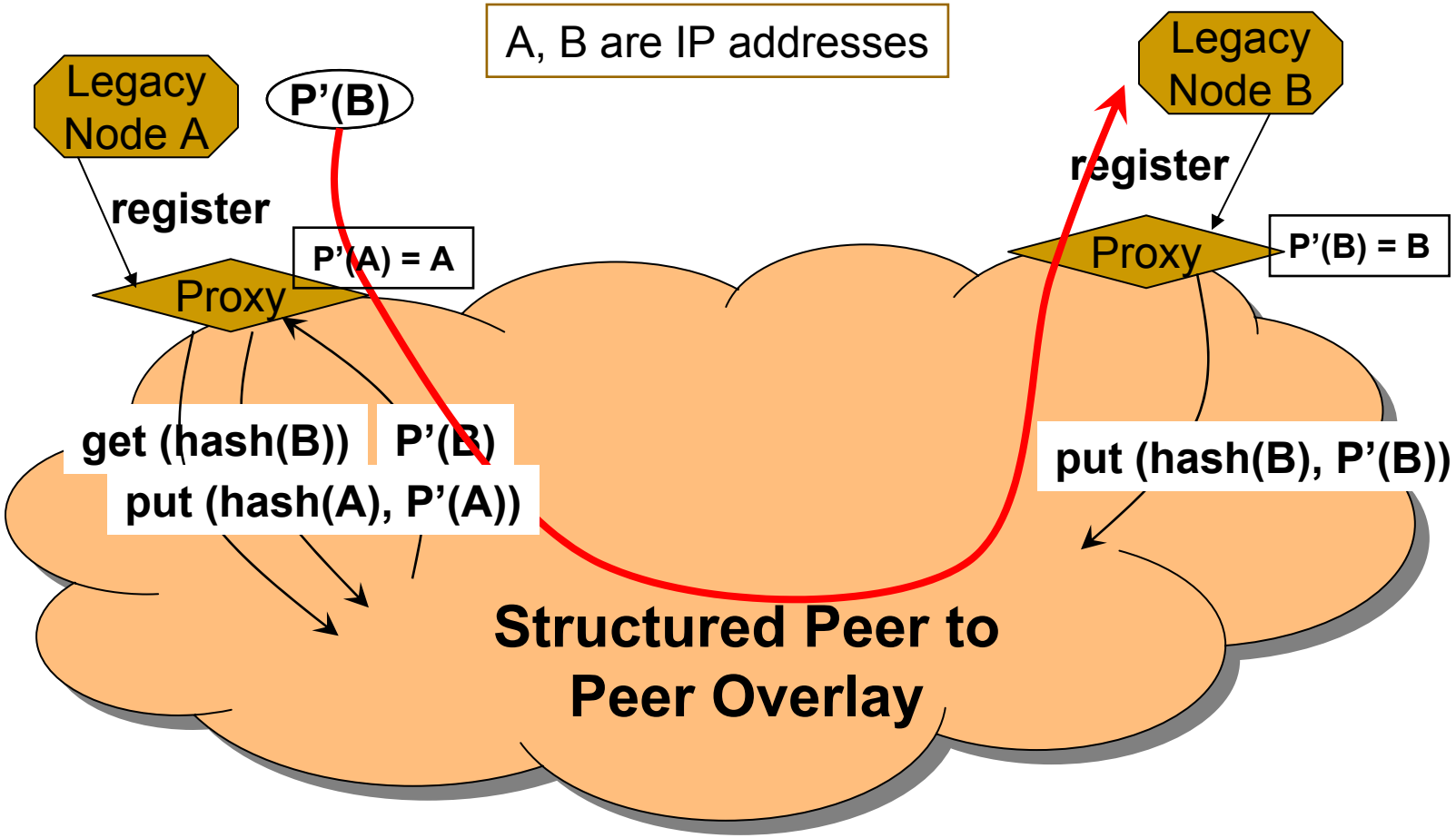
- PNS = network aware overlay construction
  - Within routing constraints, choose neighbors closest in network distance (latency)
  - Generally reduces # of IP hops
- Important for routing
  - Reduce latency
  - Reduce susceptibility to faults
    - Less IP links = smaller chance of link/router failure
  - Reduce overall network bandwidth utilization
- We use Tapestry to demonstrate our design
  - P2P protocol with PNS overlay construction
  - Topology-unaware P2P protocols will likely perform worse

System Architecture



- Locate nearby overlay proxy
- Establish overlay path to destination host
- Overlay traffic routes traffic resiliently

Traffic Tunneling



- Store mapping from end host IP to its proxy's overlay ID
- Similar to approach in *Internet Indirection Infrastructure (I3)*

Tradeoffs of Tunneling via P2P

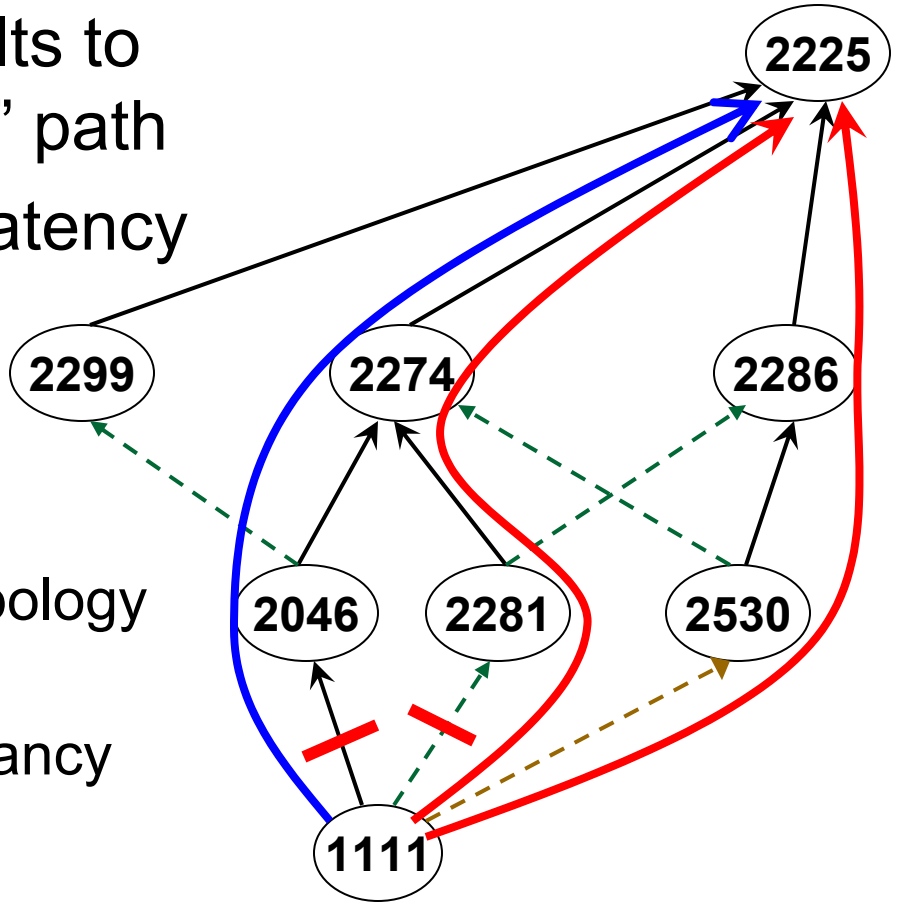
- Less neighbor paths to monitor per node: *O(log(n))*
  - Large reduction in probing bandwidth: *O(n)* → *O(log(n))*
  - Increase probing frequency
  - Faster fault detection with low bandwidth consumption
- Actively maintain path redundancy
  - Manageable for "small" # of paths
  - Redirect traffic immediately when a failure is detected
  - Eliminate on-the-fly calculation of new routes
  - Restore redundancy when a path fails
- End result
  - Fast fault detection + precomputed paths = increased responsiveness to faults
- Cons
  - Overlay imposes routing stretch (more IP hops), generally < 2

Some Details

- Efficient fault detection
  - Use soft-state to periodically probe *log(n)* neighbor paths
  - "Small" number of routes → reduced bandwidth
  - Exponentially weighted moving average in link quality estimation
    - Avoid route flapping due to short term loss artifacts
    - Loss rate $L_n = (1 - \alpha) \cdot L_{n-1} + \alpha \cdot p$
    - p = instantaneous loss rate, α = hysteresis factor
- Maintaining backup paths
  - Each hop has flexible routing constraint
    - Create and store backup routes at node insertion
  - Restore redundancy via "intelligent" gossip after failures
  - Simple policies to choose among redundant paths

First Reachable Link Selection (FRLS)

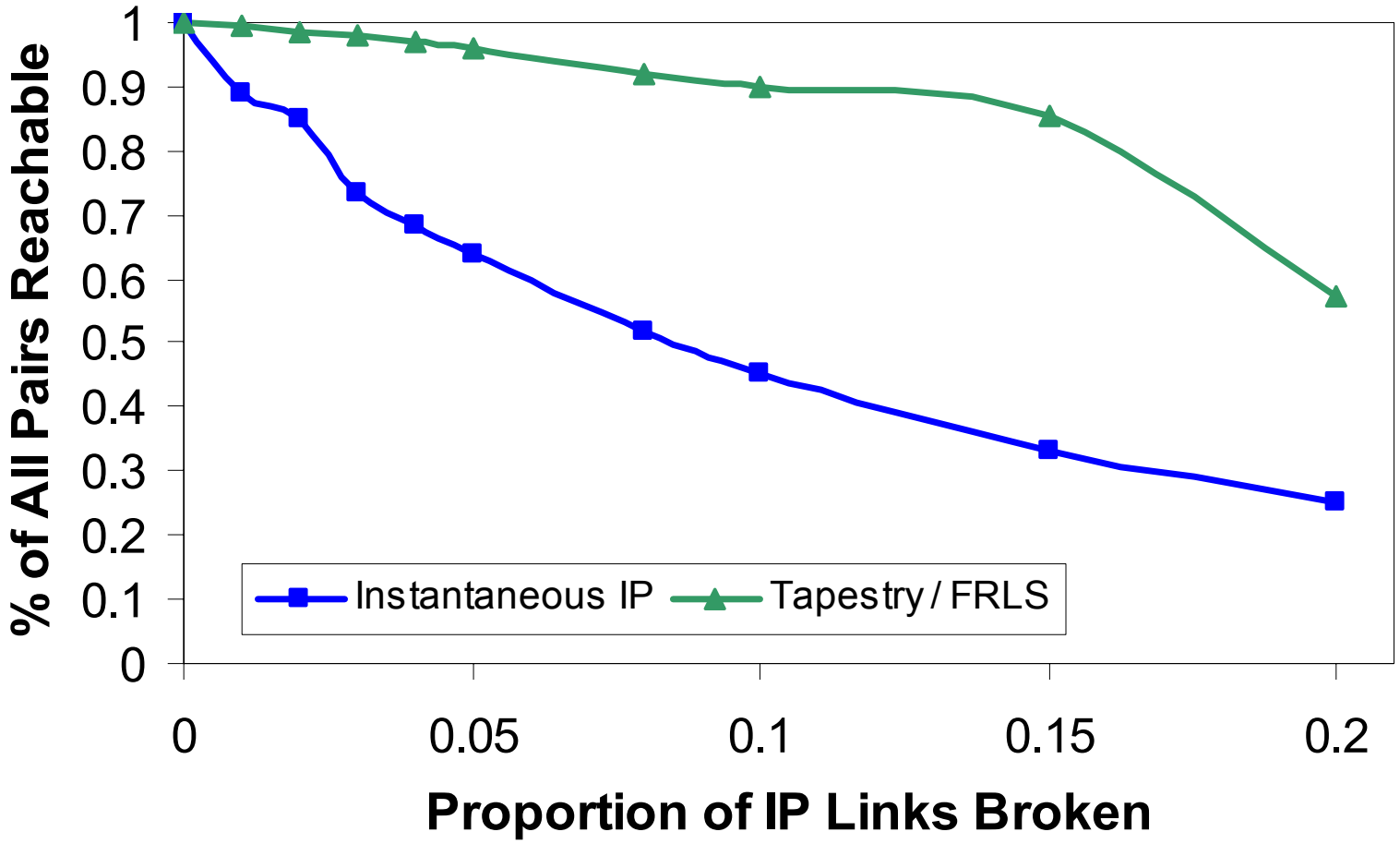
- Use estimated loss results to choose shortest “usable” path
- Sort next hop paths by latency
- Use shortest path with minimal quality > T
- Correlated failures
  - Reduce with intelligent topology construction
  - Key is to leverage redundancy available



Evaluation

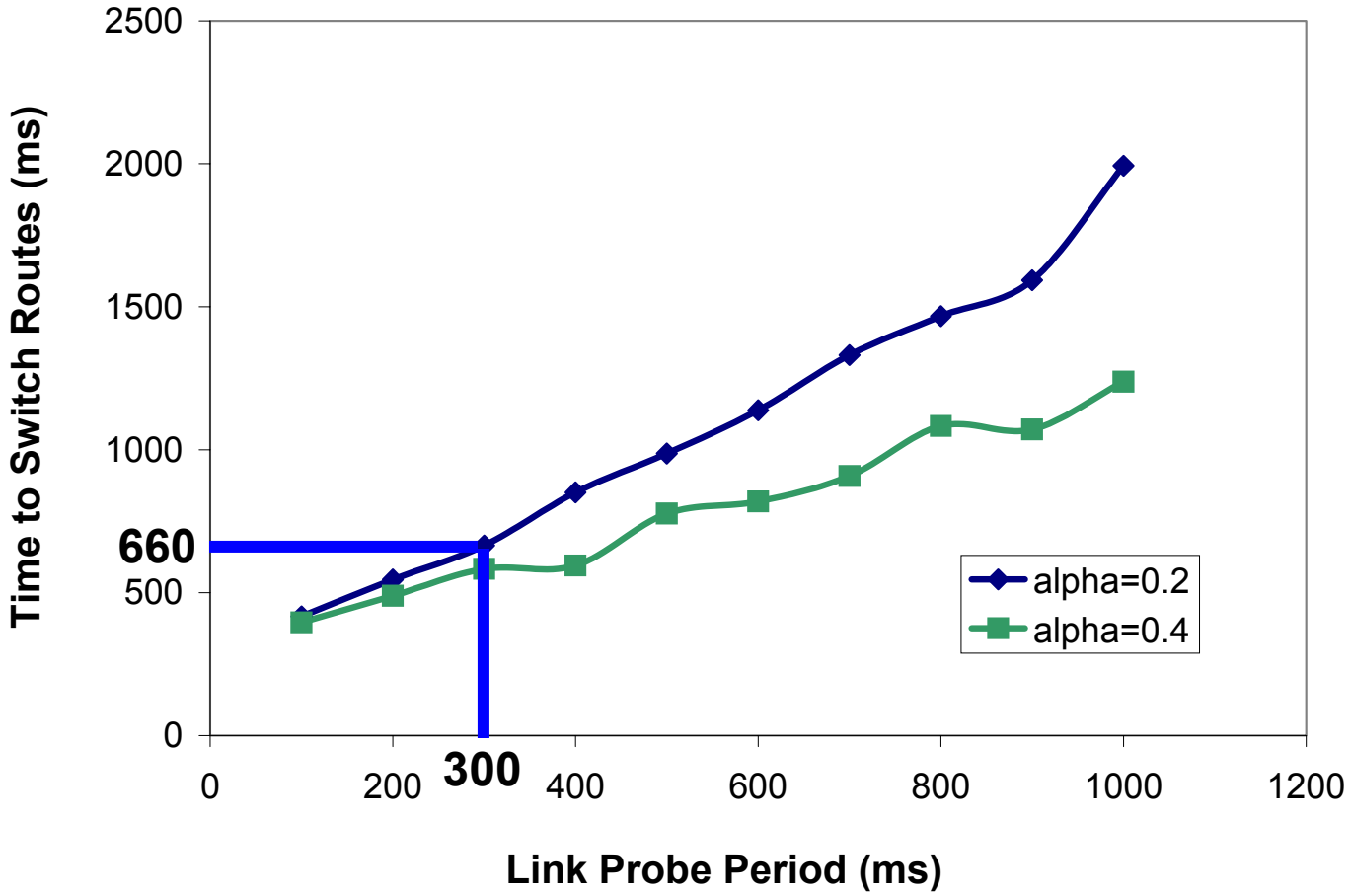
- Metrics for evaluation
  - How much routing resiliency can we exploit?
  - How fast can we adapt to faults?
  - What is the overhead of routing around a failure?
    - Proportional increase in end to end latency
    - Proportional increase in end to end bandwidth used
- Experimental platforms
  - Event-based simulations on transit stub topologies
    - Data collected over different 5000-node topologies
  - PlanetLab measurements
    - Microbenchmarks on responsiveness
    - Bandwidth measurements from 200+ node overlays
    - Multiple virtual nodes run per physical machine

Exploiting Route Redundancy (Sim)



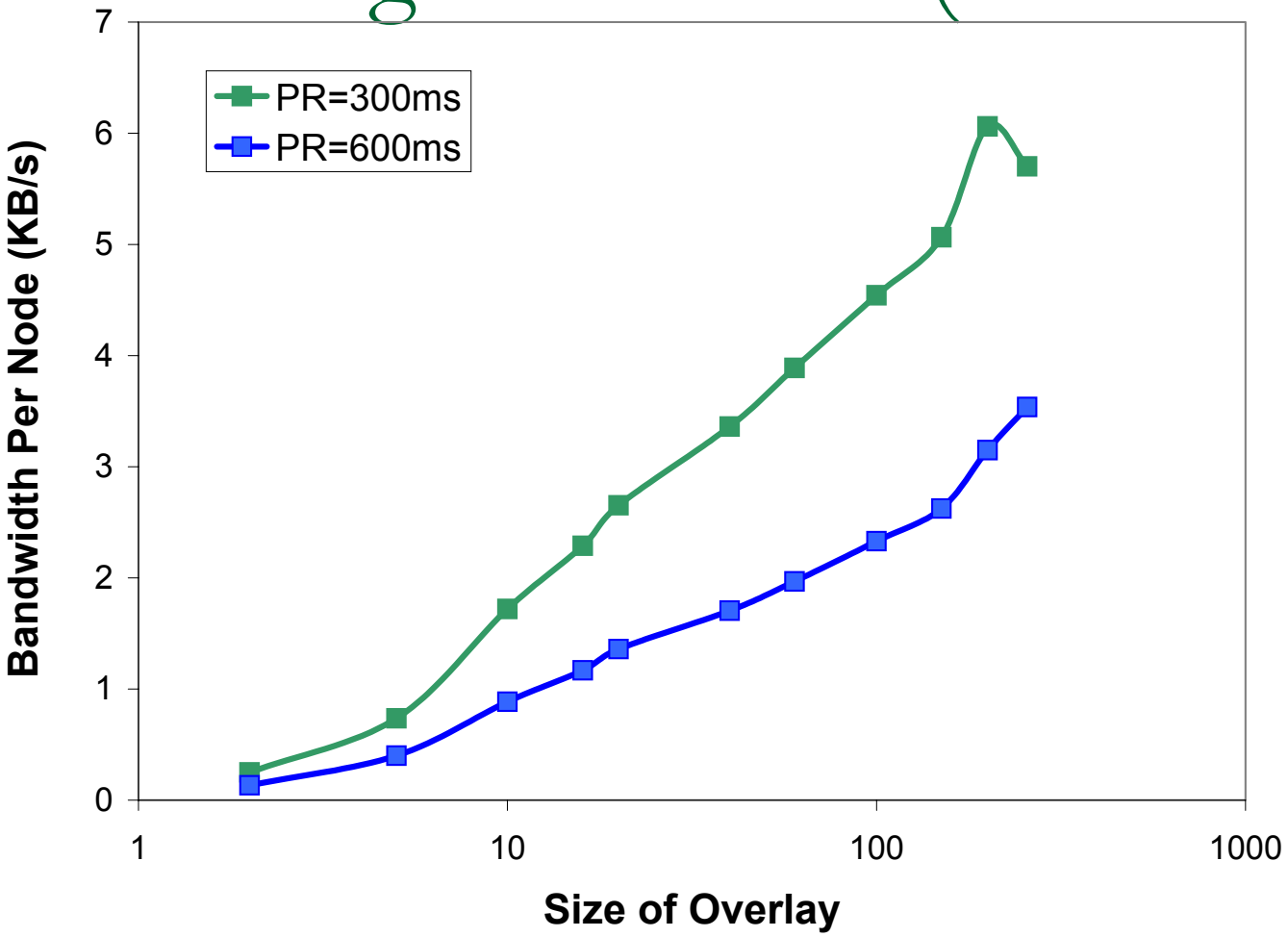
- Simulation of Tapestry, 2 backup paths per routing entry
- Transit-stub topology shown, results from TIER and AS graphs similar

Responsiveness to Faults (PlanetLab)



- Response time increases linearly with probe period
- Minimum link quality threshold T = 70%, 20 runs per data point

Link Probing Bandwidth (Planetlab)



- Medium sized routing overlays incur low probing bandwidth
- Bandwidth increases logarithmically with overlay size

Related Work

- Redirection overlays
  - Detour (IEEE Micro 99)
  - Resilient Overlay Networks (SOSP 01)
  - Internet Indirection Infrastructure (SIGCOMM 02)
  - Secure Overlay Services (SIGCOMM 02)
- Topology estimation techniques
  - Adaptive probing (IPTPS 03)
  - Peer-based shared estimation (Zhuang 03)
  - Internet tomography (Chen 03)
  - Routing underlay (SIGCOMM 03)
- Structured peer-to-peer overlays
  - Tapestry, Pastry, Chord, CAN, Kademlia, Skipnet, Viceroy, Symphony, Koorde, Bamboo, X-Ring…

Conclusion

- Benefits of structure outweigh costs
  - Structured routing lowers path maintenance costs
    - Allows “caching” of backup paths for quick failover
  - Can no longer construct arbitrary paths
    - Structured routing with low redundancy gets very close to ideal in connectivity
    - Incur low routing stretch
- Fast enough for highly interactive applications
  - 300ms beacon period → response time < 700ms
  - On overlay networks of 300 nodes, b/w cost is 7KB/s
- Future work
  - Deploying a public routing and proxy service on PlanetLab
  - Examine impact of
    - Network aware topology construction
    - Loss sensitive probing techniques

Questions…

- Related websites:
  - Tapestry

  http://www.cs.berkeley.edu/~ravenben/tapestry
  - Pastry

  http://research.microsoft.com/~antr/pastry
  - Chord

  http://lcs.mit.edu/chord
- Acknowledgements
  - Thanks to Dennis Geels and Sean Rhea for their work on the BMark benchmark suite