Best-Path vs. Multi-Path Overlay Routing

David G. Andersen (MIT)

Alex C. Snoeren (UCSD)

Hari Balakrishnan (MIT)

October 2003

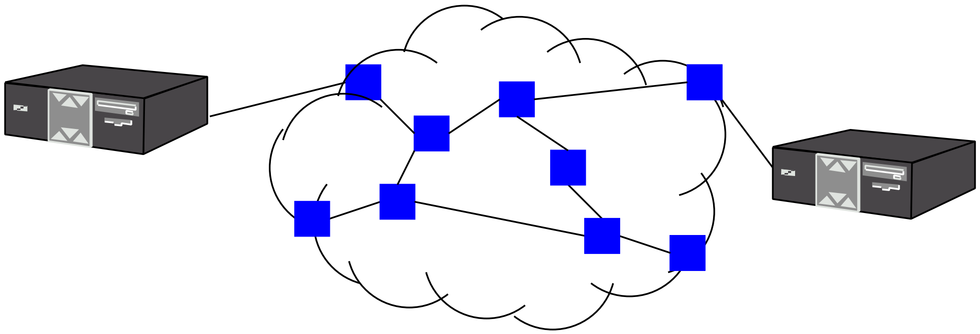
`http://nms.lcs.mit.edu/ron/`

Overview

Best-path vs. redundant overlay routing

- What tactics work best to
  - Reduce loss?
  - Reduce latency?
  - Avoid outages?
- In what circumstances do they perform best?
- Implications for new strategies

Context: Reliability via Path Diversity



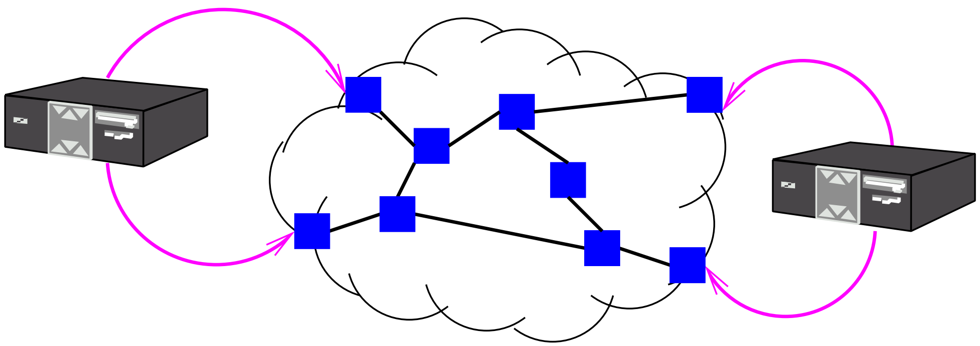
- Backup links provide alternatives

➔ Mechanisms for obtaining diversity
(existing diversity)

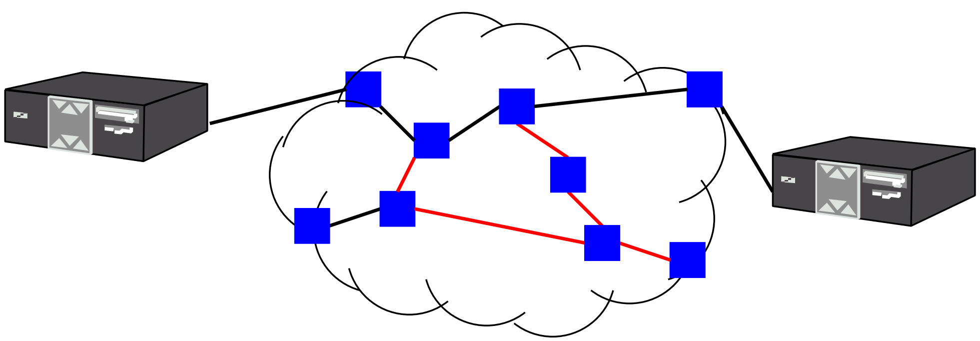
➔ Mechanisms for using diversity
(overlay techniques)

Obtaining Diversity

Engineered diversity:

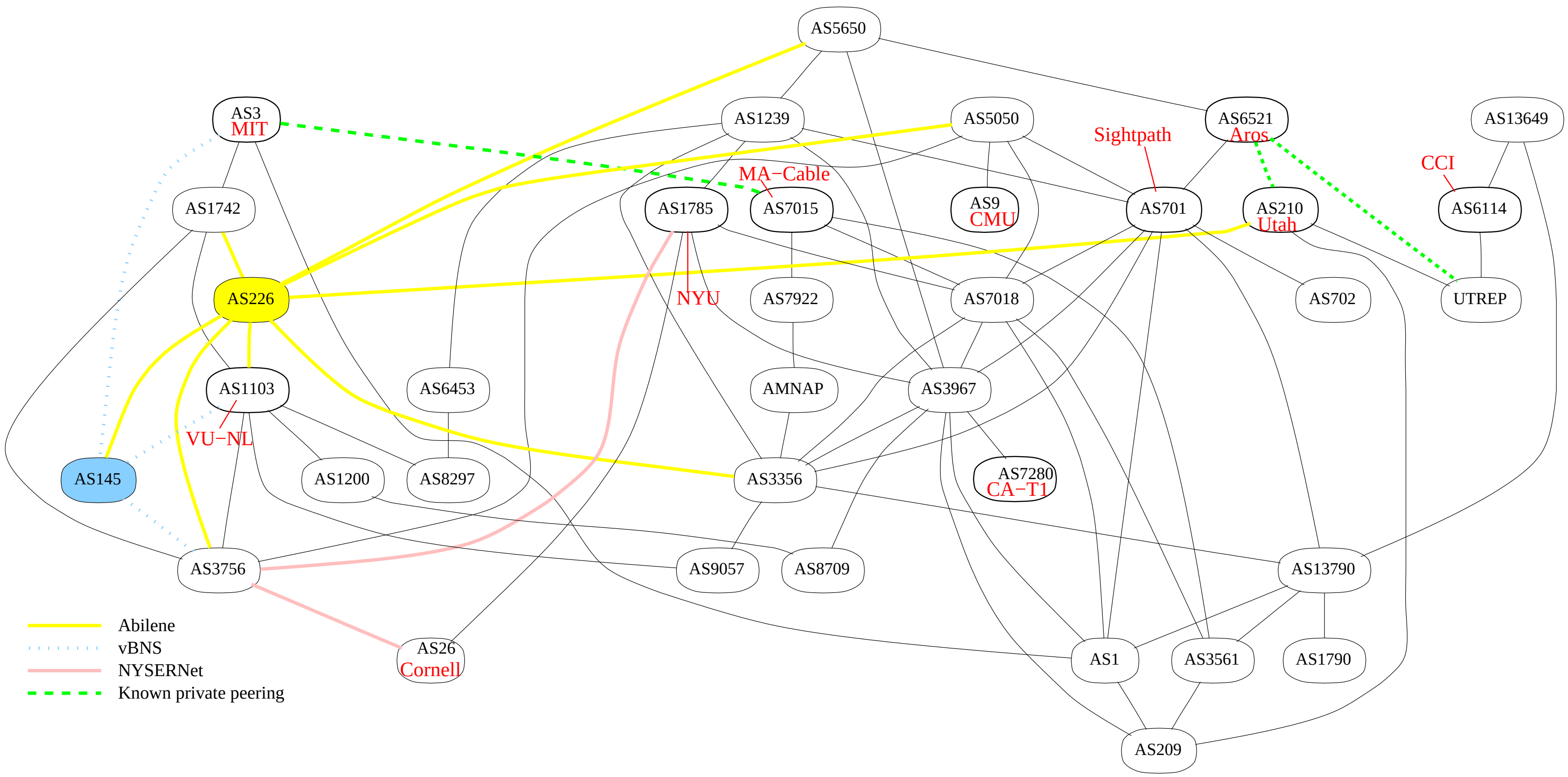


Exploiting existing diversity:

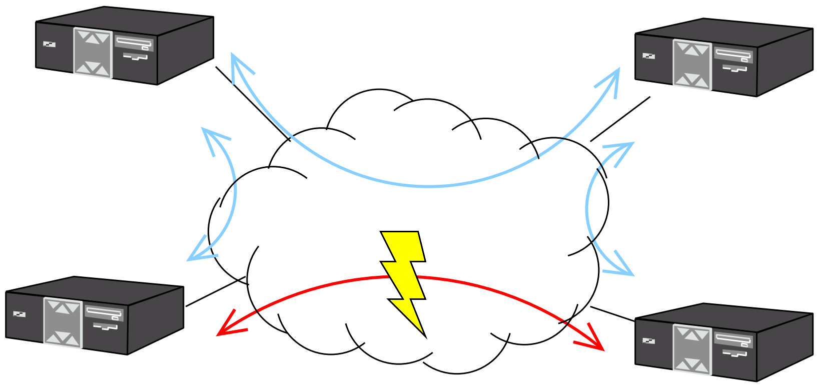


Existing AS-level Redundancy

- Traceroute between 12 hosts, showing Autonomous Systems (AS's)



Exploiting Diversity via overlays

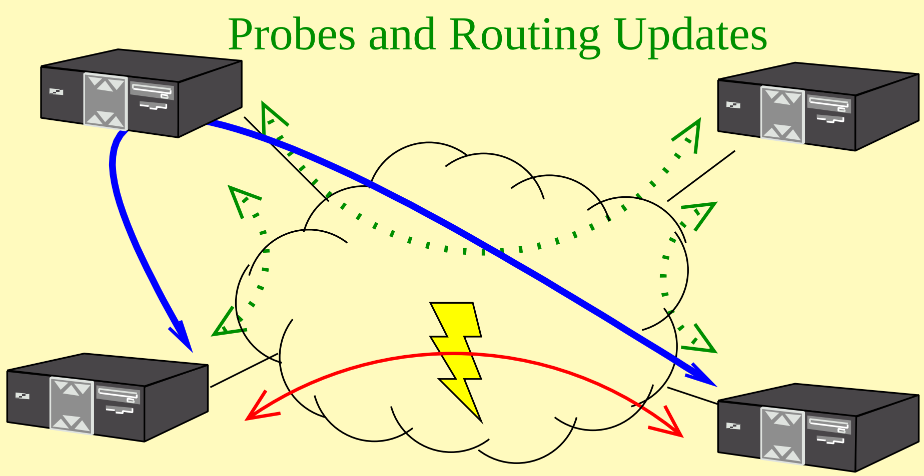


- Send packets through cooperating peers
- End-hosts only, no network support

Exploiting Diversity via Overlays

Reactive Routing

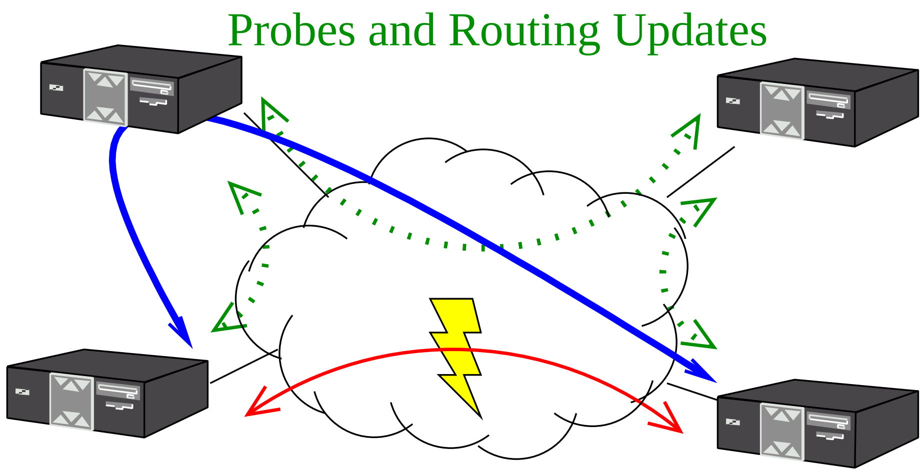
- Probe paths
- Route via best
- RON (SOSP'01) Detour



Exploiting Diversity via Overlays

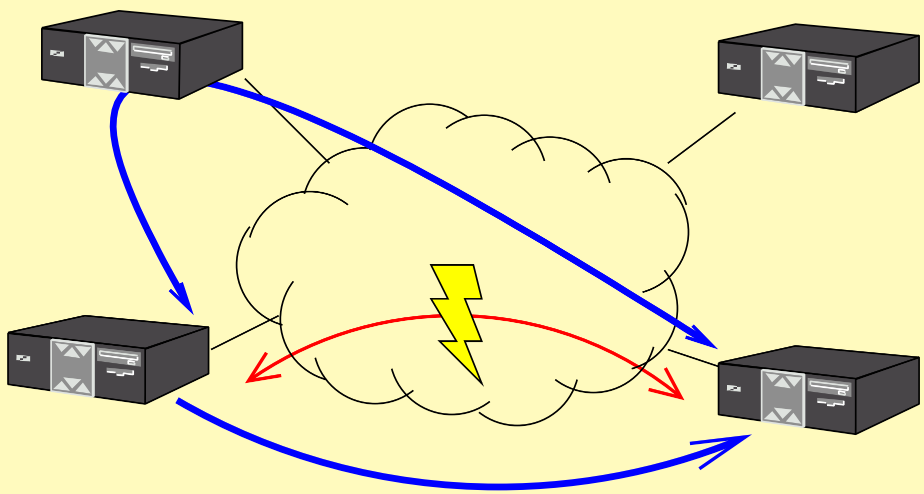
Reactive Routing

- Probe paths
- Route via best

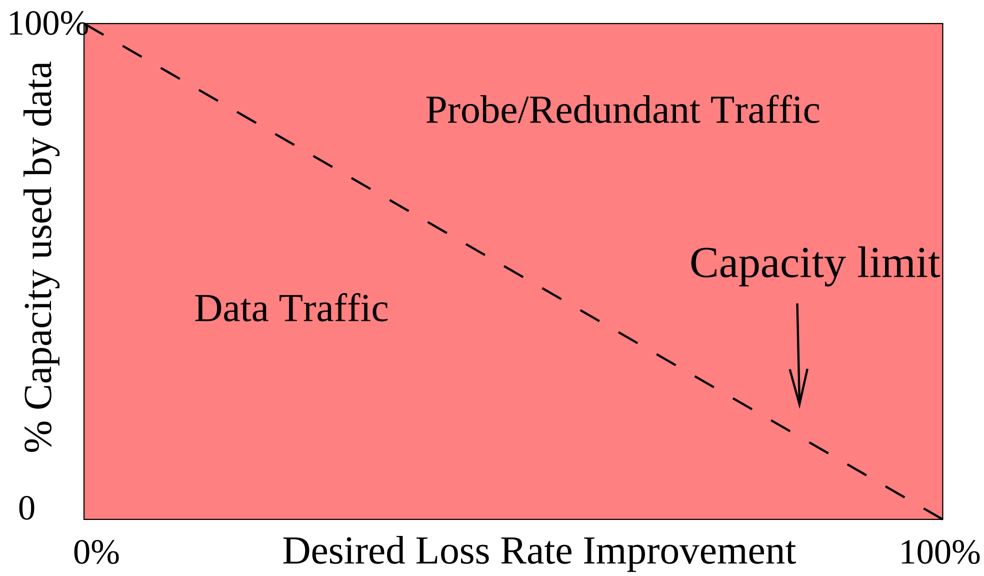


Redundant Routing

- Parallel paths
- No probing
- Mesh routing (SOSP'01)

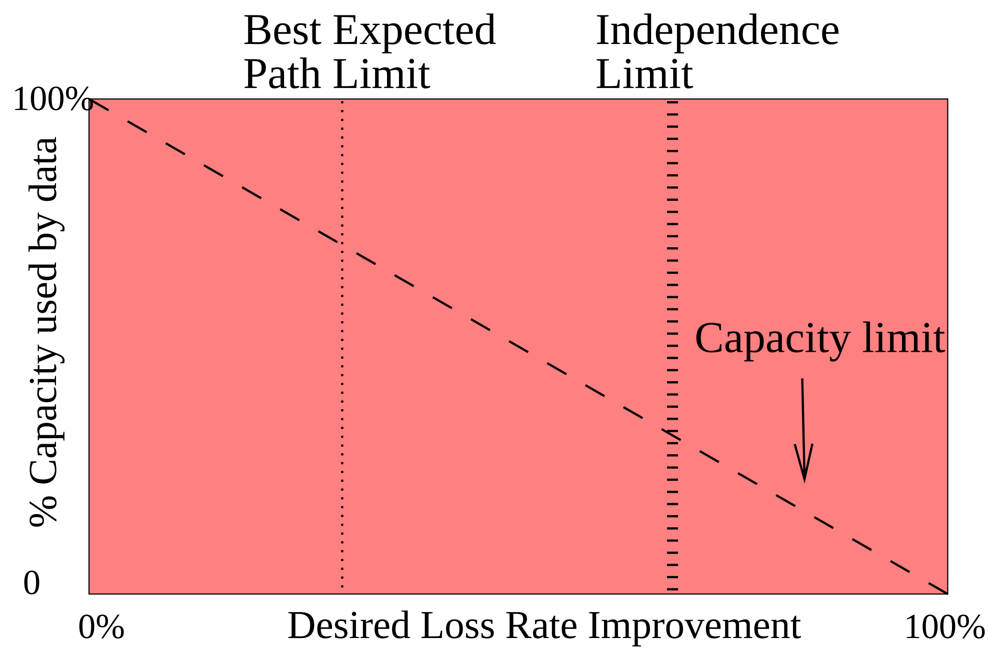


Reactive vs. Redundant Routing



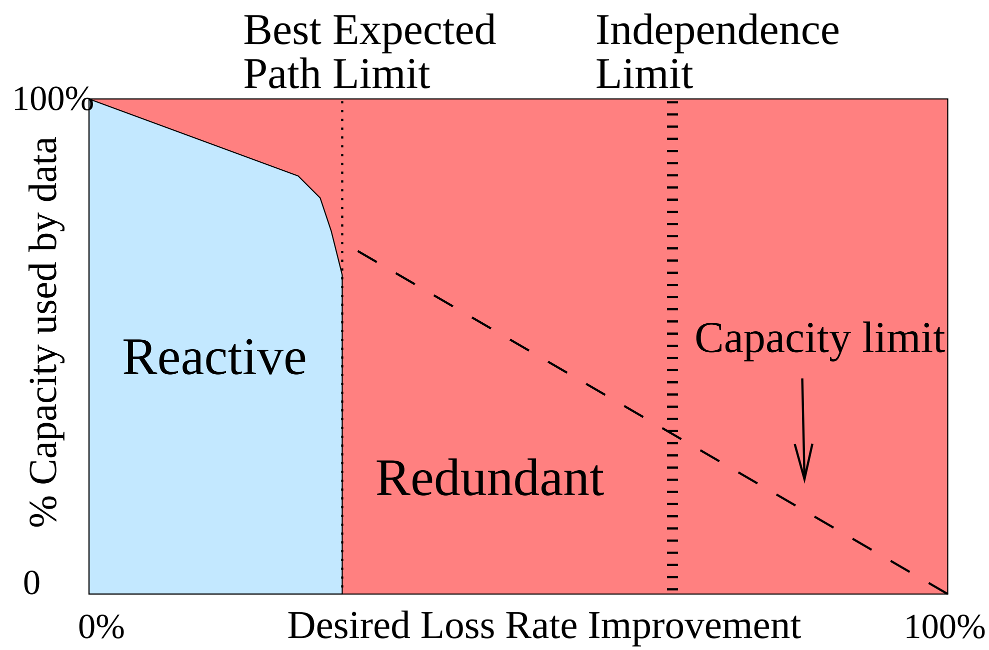
- Capacity limits probing *and* redundancy

Reactive vs. Redundant Routing



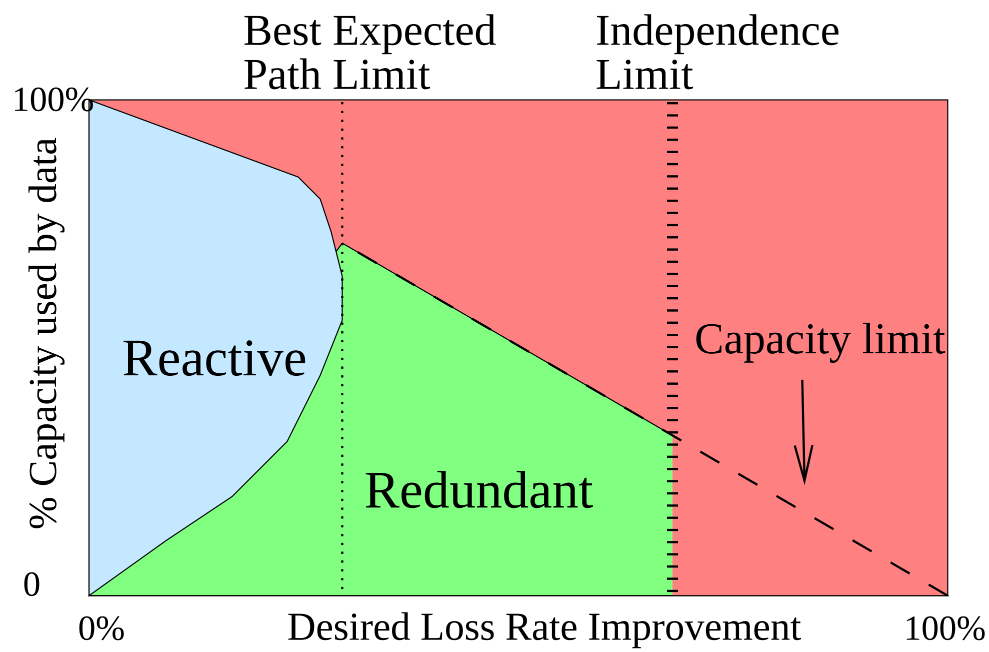
- Reactive limit: best path performance
- Redundant limit: Path independence

Reactive vs. Redundant Routing



- Reactive limit: best path performance
- Redundant limit: Path independence

Reactive vs. Redundant Routing



- Reactive limit: best path performance
- Redundant limit: Path independence
- Overhead scaling: throughput vs. nodes

8 Routing Methods

Direct	Single packet, direct path
Direct Direct	2 packets, direct, no spacing
DD 10ms	2 packets, direct, 10ms spacing
DD 20ms	2 packets, direct, 20ms spacing

8 Routing Methods

Direct	Single packet, direct path
Direct Direct	2 packets, direct, no spacing
DD 10ms	2 packets, direct, 10ms spacing
DD 20ms	2 packets, direct, 20ms spacing
Lat	Reactive routing, min latency
Loss	Reactive routing, min loss

8 Routing Methods

Direct	Single packet, direct path
Direct Direct	2 packets, direct, no spacing
DD 10ms	2 packets, direct, 10ms spacing
DD 20ms	2 packets, direct, 20ms spacing
Lat	Reactive routing, min latency
Loss	Reactive routing, min loss
Direct Rand	2pkts, Redundant routing, simplest

8 Routing Methods

Direct	Single packet, direct path
Direct Direct	2 packets, direct, no spacing
DD 10ms	2 packets, direct, 10ms spacing
DD 20ms	2 packets, direct, 20ms spacing
Lat	Reactive routing, min latency
Loss	Reactive routing, min loss
Direct Rand	2pkts, Redundant routing, simplest
Lat Loss	2pkts, Reactive + Redundant (Falls back to random)

Probing on Internet Testbed

Each node repeats:

1. Pick random node j
2. Pick one of the 8 routing types
   $(direct, loss, lat, etc.)$
   in round-robin order. Send to j.
3. Delay for random interval [0.6s - 1.2s]

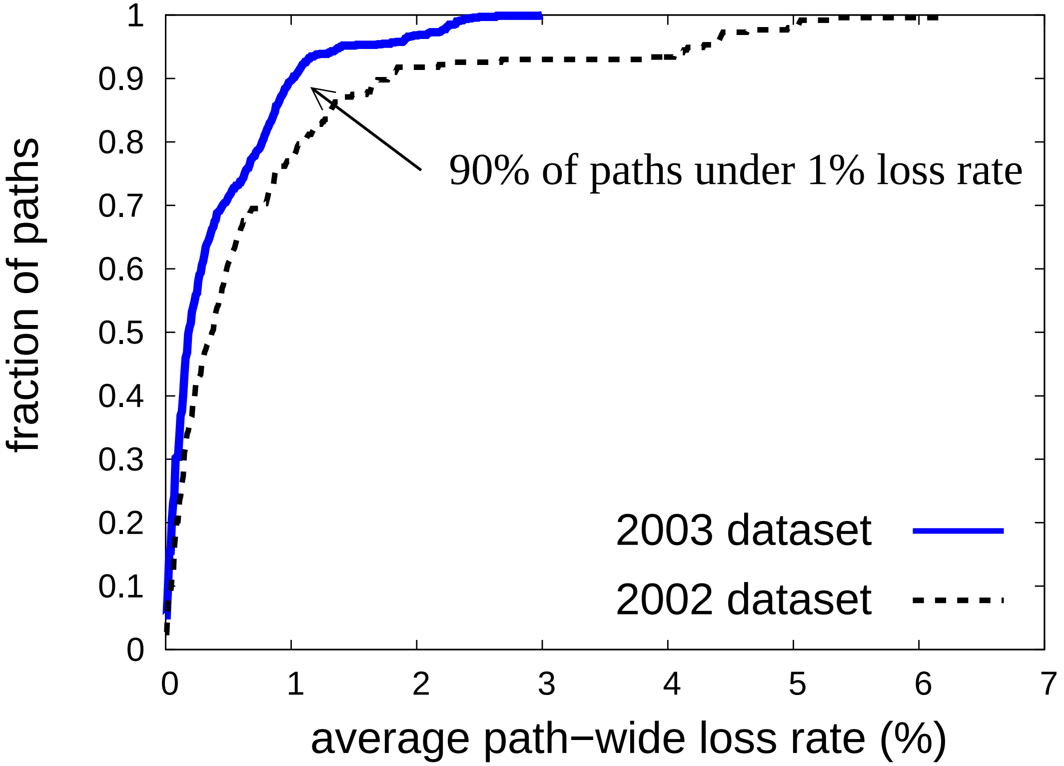
Probes are one-way, recorded at sender & receiver.

Datasets From Internet Deployment

Dataset	Nodes	Time	Measurements
RON_{wide}	17	5 days	4.7M
RON_{narrow}	17	3 days	2.8M
RON_{2003}	30	14 days	32.6M

- ✔ Variety of network types and bandwidths
  5 int'l, 3 Cable/DSL, 7 universities...
- ✔ N^2 path scaling $\sim$ 900 paths

One-way Loss Rates Are Low



- Overall loss 0.42% in 2003

- Includes quiescent periods
- Outages still (painfully) apparent

Duplication Reduces Overall Loss

Type	Loss %
direct	0.42
direct direct	0.30
dd 10ms	0.27
dd 20ms	0.27

Duplication Reduces Overall Loss

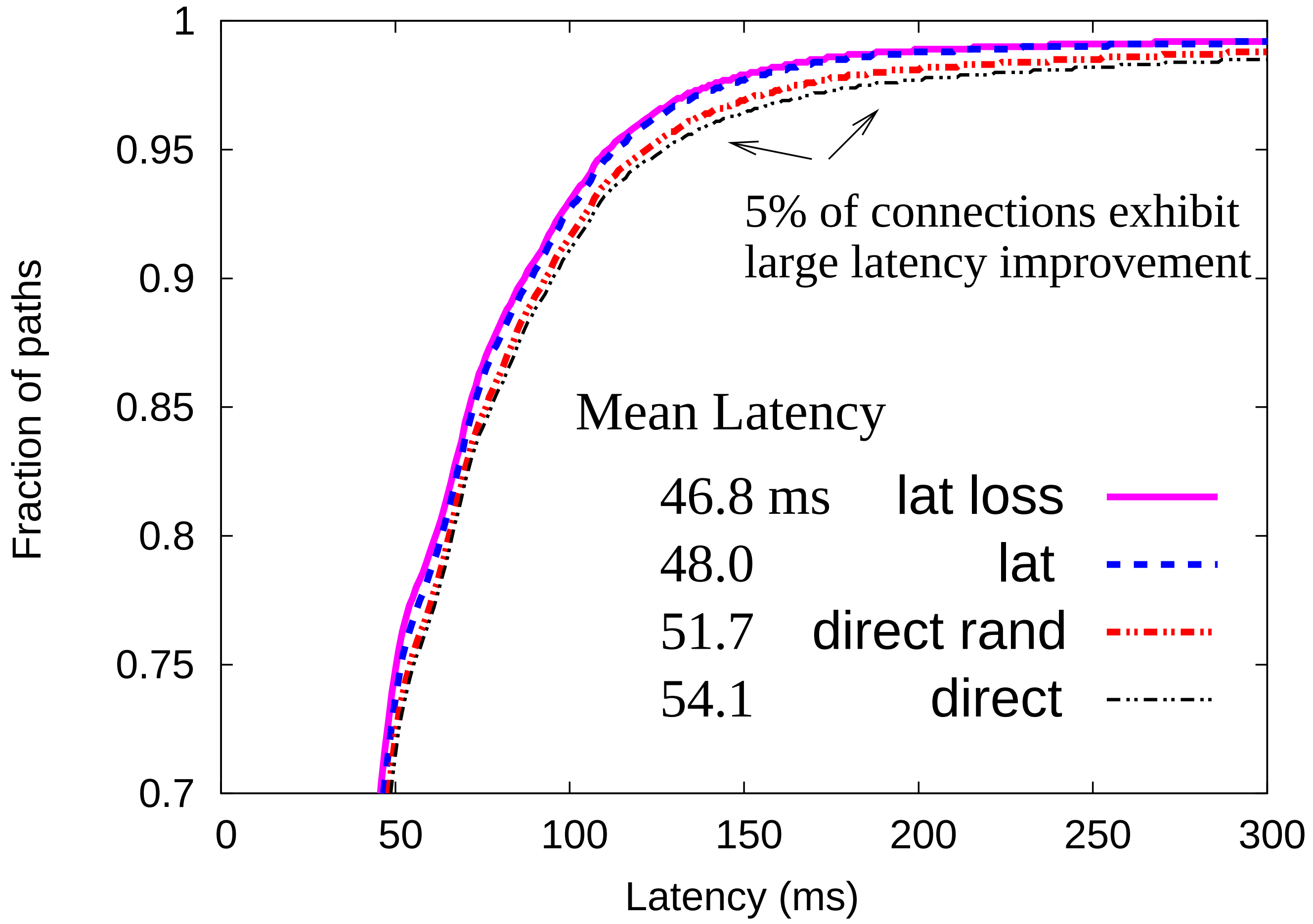
Type	Loss %
direct	0.42
direct direct	0.30
dd 10ms	0.27
dd 20ms	0.27
Lat	0.43
Loss	0.33
Direct Rand	0.26
Lat Loss	0.23

Loss Probabilities Sanity Check

- 0.42% loss $<<$ [Paxson 94,95] (2.8%, 5%).
- Unloaded paths vs. loaded by TCP transfer
- Conditional loss probabilities are similar

Study	$P(\text{lose P2}\mid\text{lost P1})$
Paxson TCP	$\sim$ 50%
Bolot 8ms spacing	60%
RON_{2003} no spacing	72%
RON_{2003} 20ms	65%
RON_{2003} direct rand	62%

Latency Improvements



Unlike loss, most latency from specific bad paths

High Loss Periods (1 hr, normalized)

Type	$> 0\%$	
direct	1 (8817)	
direct direct	0.59	
dd 20ms	0.43	
Lat	1.2	
Loss	0.80	$\leftarrow$ Worse than naive duplication
Direct Rand	0.44	for low loss situations
Lat Loss	0.38	

High Loss Periods (1 hr, normalized)

Type	$> 0\%$	$> 30\%$	
direct	1 (8817)	1 (630)	
direct direct	0.59	0.93	
dd 20ms	0.43	0.91	
Lat	1.2	0.96	
Loss	0.80	0.91	$\leftarrow$ on par
Direct Rand	0.44	0.92	
Lat Loss	0.38	0.89	

High Loss Periods (1 hr, normalized)

Type	$> 0\%$	$> 30\%$	$> 60\%$
direct	1 (8817)	1 (630)	1 (255)
direct direct	0.59	0.93	0.98
dd 20ms	0.43	0.91	0.98
Lat	1.2	0.96	0.91
Loss	0.80	0.91	0.86 ★
Direct Rand	0.44	0.92	0.92 ★
Lat Loss	0.38	0.89	0.84 ★

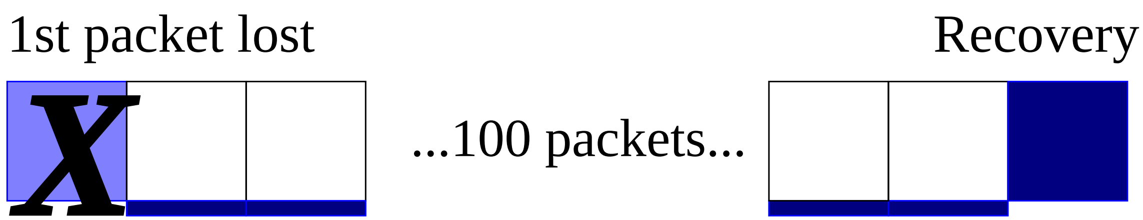
Measurement Summary

- ✔ Redundant beats reactive for low loss
  - “Meshing” beats controls during outages
- ✔ Reactive finds specific good paths
  - Latency improvements
  - Low loss paths
- ✘ No overlay technique near independent paths
  - Hypothesis: Access link failures
  - More severe outages harder to correct

Why Not FEC?

Redundant assumption: Fast recovery, low rate

0.42% loss rate $\longrightarrow$ need little redundancy



Failure losses bursty ($\geq$ 0.5 conditional loss)

- ✘ Spread FEC over *even more* packets

➔ Latency-critical traffic: 2-redundant mesh

Conclusions

- Loss rate for low-rate traffic low (0.42%)
- Conditional loss probability high (0.72)
  even for random mesh (0.62)
- 40-60% of loss avoidable

✔ Reundant: Avoiding low loss rates

✔ Reactive: Avoiding high loss, latency

➔ Low loss suggests selective approach . . .

Future Work

Strategies for avoiding losses and outages:

- Selective redundancy: Protecting SYNs, etc. (shameless plug: Currently implementing)
- Selective probing: Activate on first loss

Measurements:

- Engineered network redundancy impact? (testing now, looking for multihomed sites)

`http://nms.lcs.mit.edu/ron/`

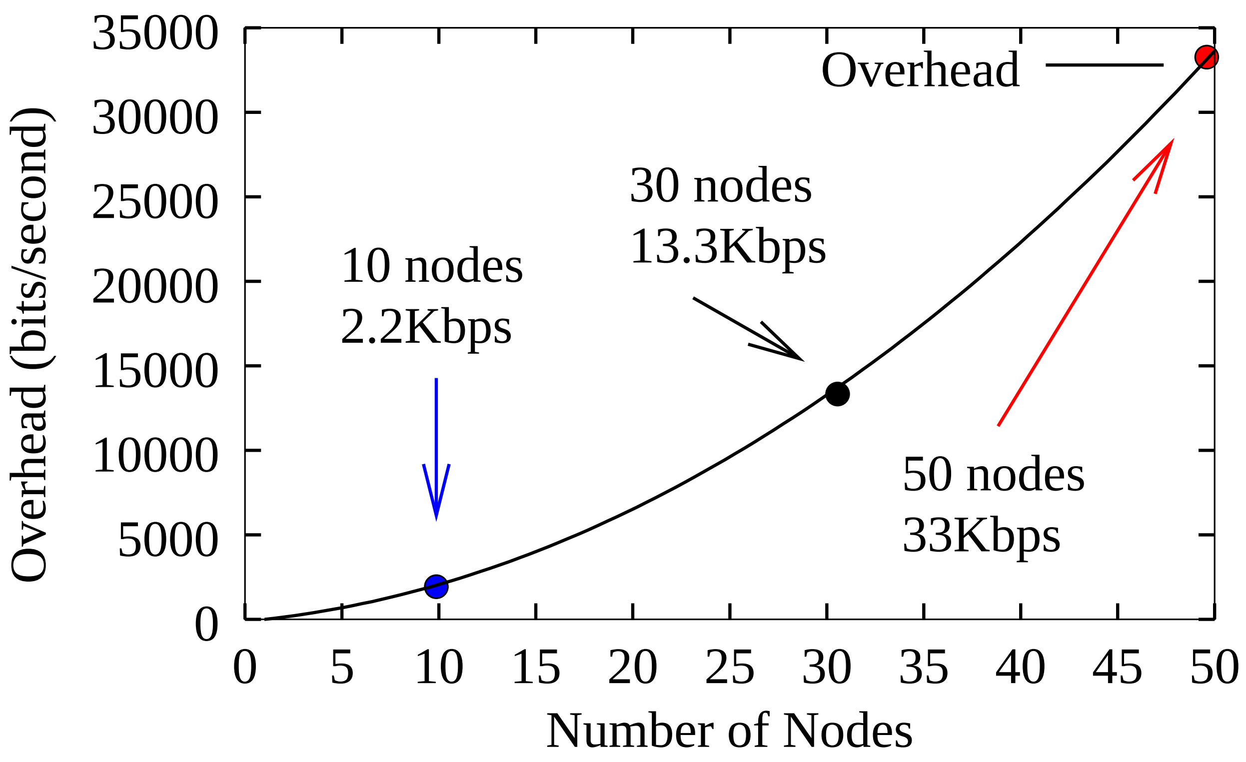
Scaling

- Reactive: Scales with # nodes
- Redundant: Scales with traffic volume

Best Path Scaling

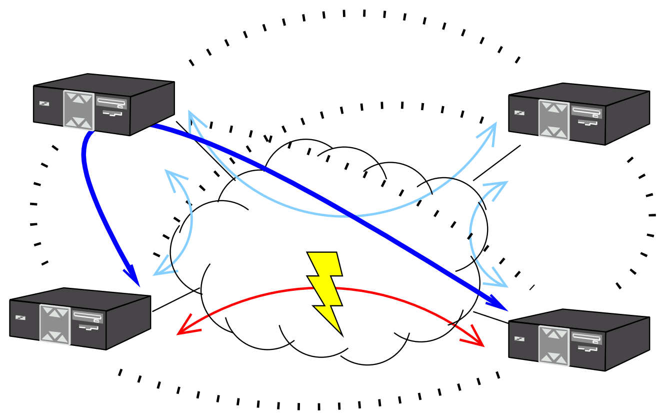
Routing and probing add packets:

Responsiveness vs. overhead vs. size



- 50 nodes near limit, enough for many apps.

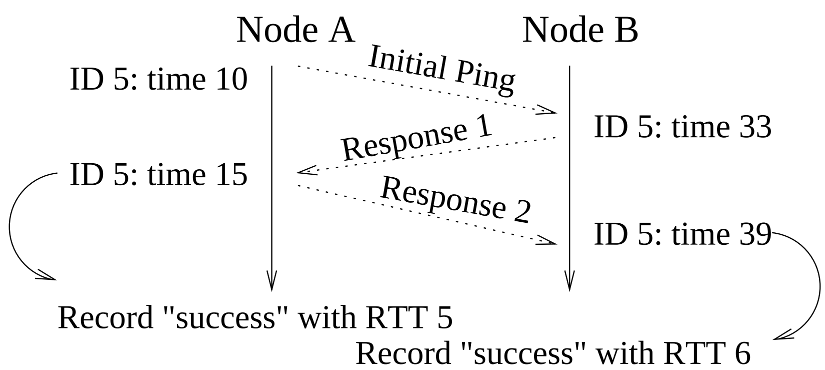
Best Path Routing



Probes and Routing

- Frequently measure *all* inter-node paths
- Exchange routing information
- Route along app-specific best path consistent with routing policy

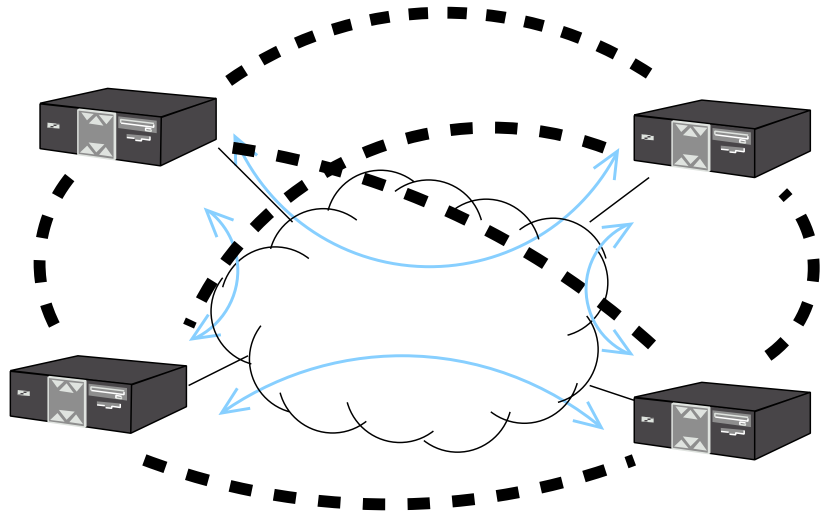
Probing and Outage Detection



- Probe every random(14) seconds
- 3 packets, both sides get RTT and reachability
- If "lost probe," send next immediately

  Timeout based on RTT and RTT variance
- If N lost probes, notify outage

Architecture: Probing



- Probe between nodes, determine path qualities
  - $O\left(N^2\right)$ probe traffic with active probes
  - Passive measurements