# RON: Choosing Resiliency

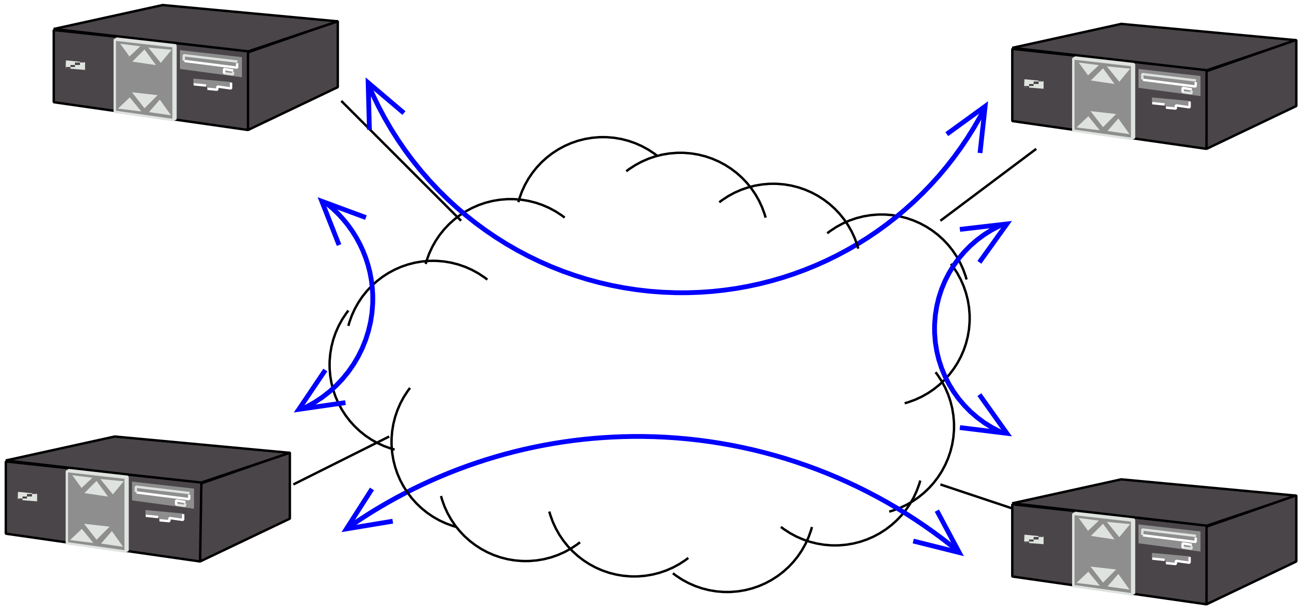
*David G. Andersen, Hari Balakrishnan, M. Frans Kaashoek*
*Robert Morris, Alex Snoeren*

**MIT Laboratory for Computer Science**

October 2002

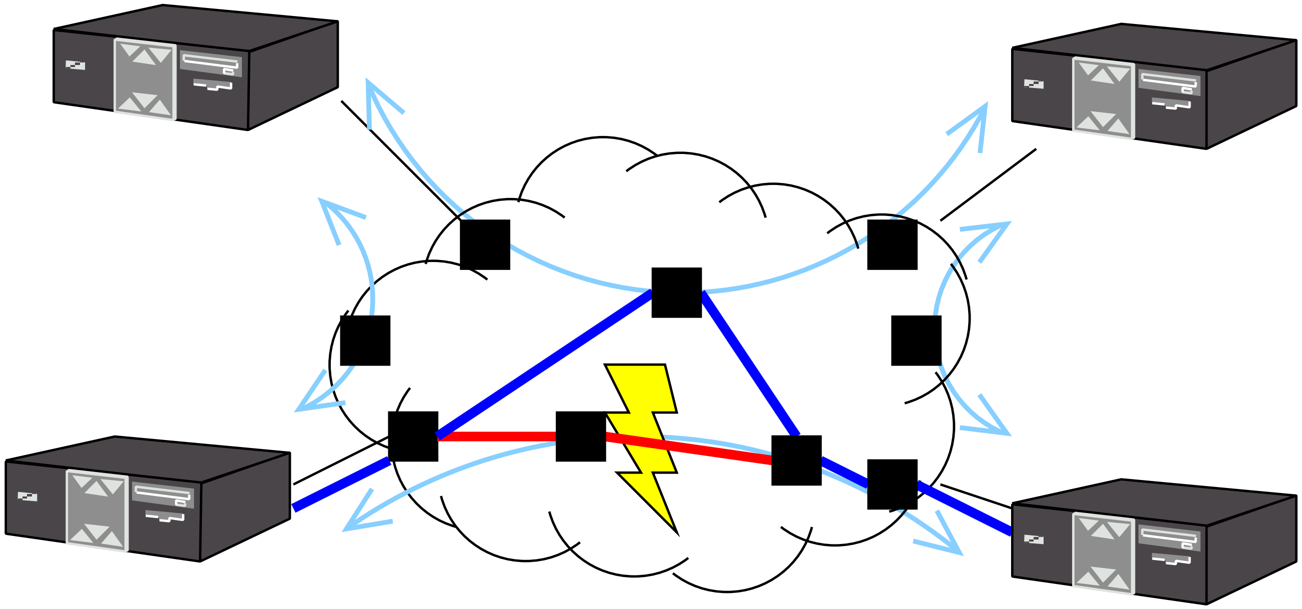
`http://nms.lcs.mit.edu/ron/`

# The Internet Abstraction



- Any-to-any communication

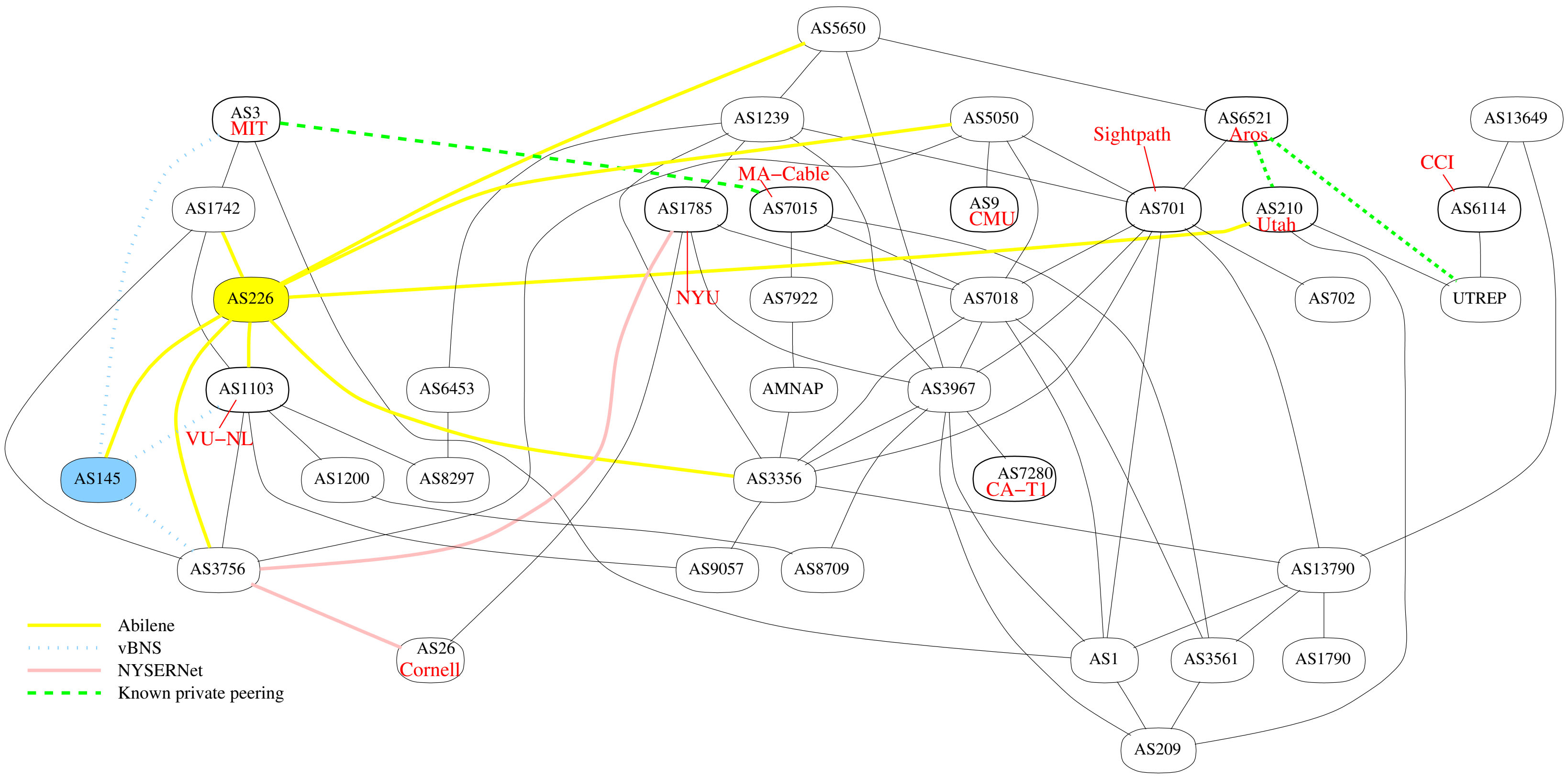
# The Internet Abstraction



- Any-to-any communication transparently routing around failures

# The Internet *has* Redundancy

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



# How Robust is Internet Routing?

✔ Scales well

✘ Suffers slow outage detection and recovery

Internet backbone routing also cannot:

- Detect badly performing paths
- Efficiently leverage redundant paths
- Multi-home small customers
- Express sophisticated routing policy / metrics

➔ We'd like to fix these shortcomings

# Goal

Improve communication availability, at a layer where we can affect the network: Overlay communities.

- Collaboration and conferencing
- Virtual Private Networks (VPNs)
- 5 friends who want better service...
- ...Or a new kind of ISP?

Interest in improving communication between *any* members of the community

# Overlays

- Old idea in networks

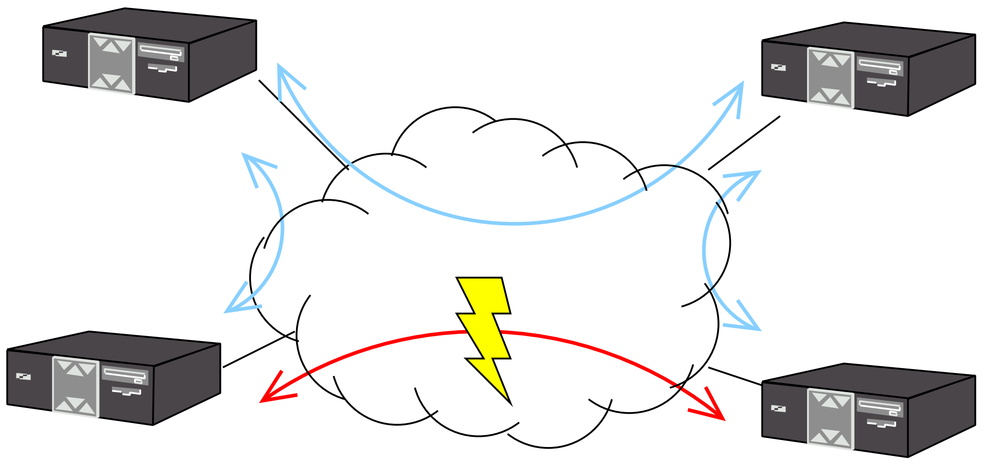
✔ Easily deployed

✔ Lets Internet focus on scalability

✔ Keep functionality between *active* peers

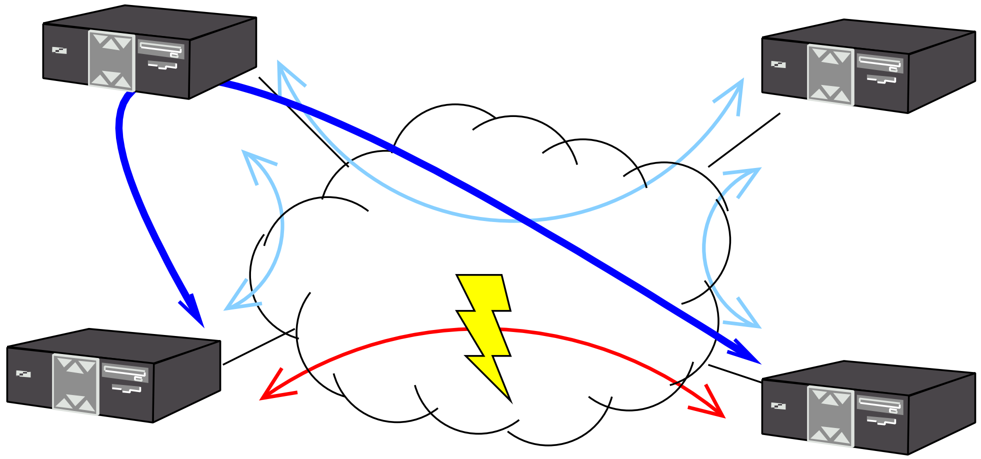
✔ Lets us choose resiliency mechanisms

# RON: Routing around Internet Failures



The Internet takes a while to re-route

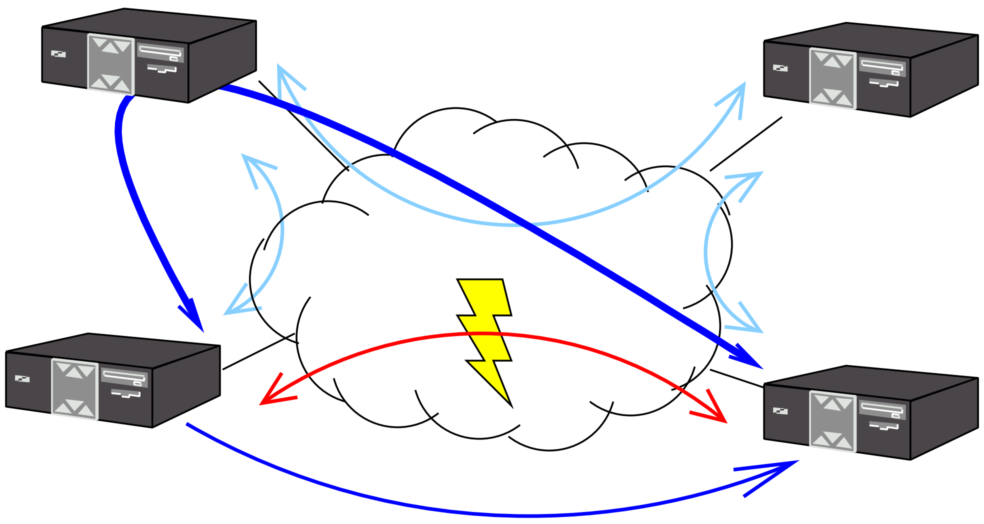
# RON: Best Path Routing



The Internet takes a while to re-route

... Cooperating hosts in different routing domains can do better by re-routing through a peer node

# RON: Redundant Multipath Routing

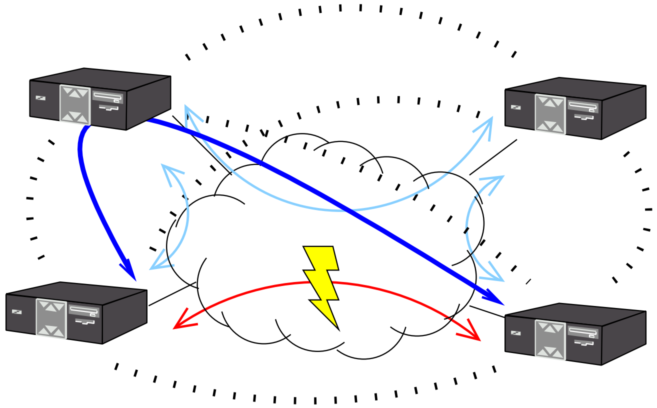


The Internet takes a while to re-route

...So proactively defend against loss

by using multiple routes

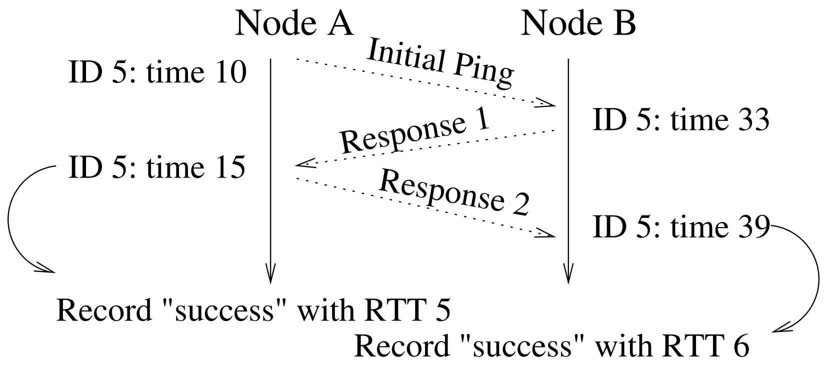
# Best Path Routing



Probes and Routing

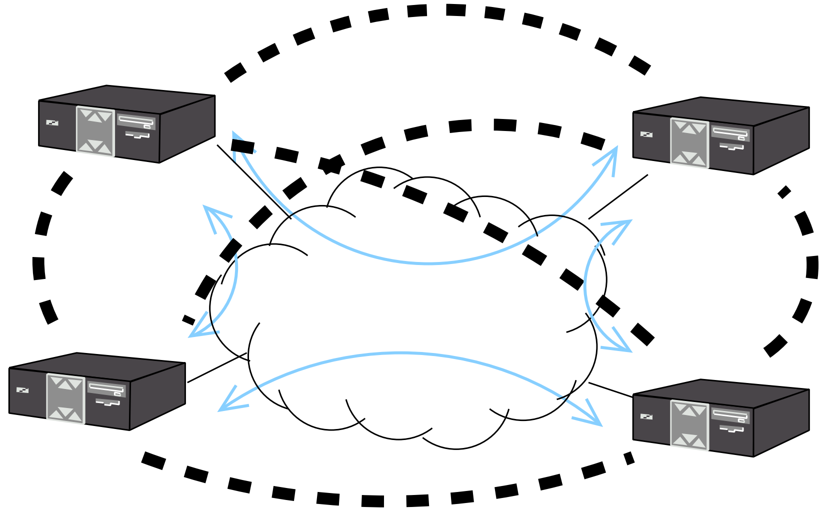
- Frequently measure *all* inter-node paths
- Exchange routing information
- Route along app-specific best path consistent with routing policy

# Probing and Outage Detection



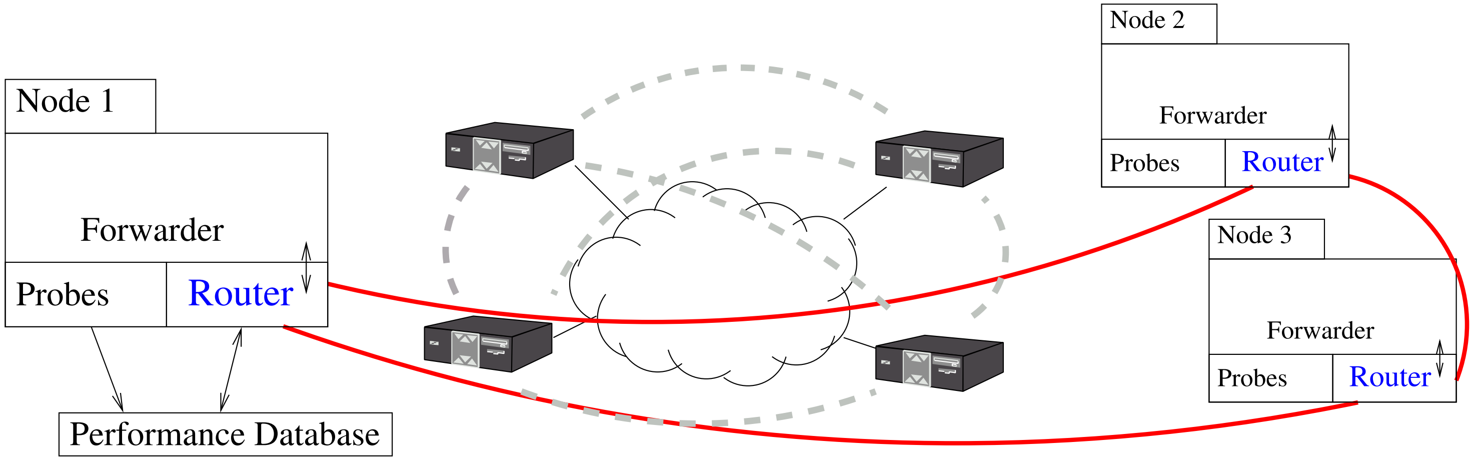

- Probe every random(14) seconds
- 3 packets, both sides get RTT and reachability
- If “lost probe,” send next immediately
  Timeout based on RTT and RTT variance
- If $N$ lost probes, notify outage

# Architecture: Probing



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

# Architecture: Routing Protocol




- Probe between nodes, determine path qualities
- Store probe results in performance database
- ➔ Link-state routing protocol between nodes
  Disseminates info using the overlay

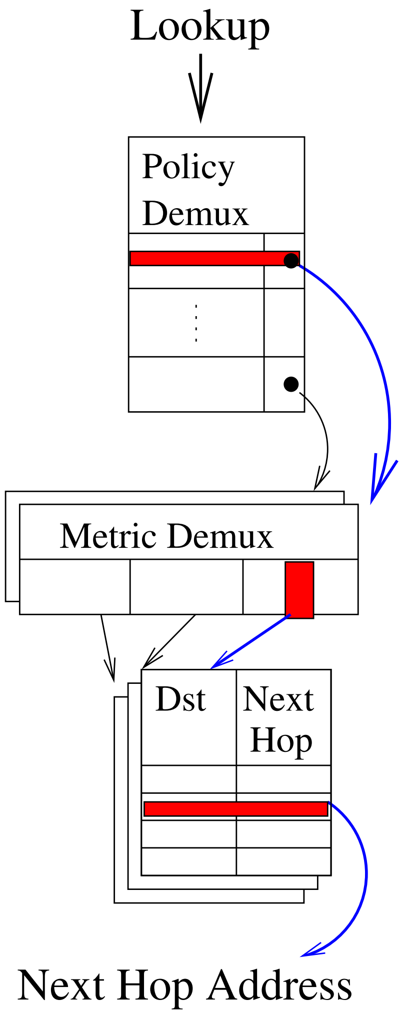
# Routing: Building Forwarding Tables

Policy routing

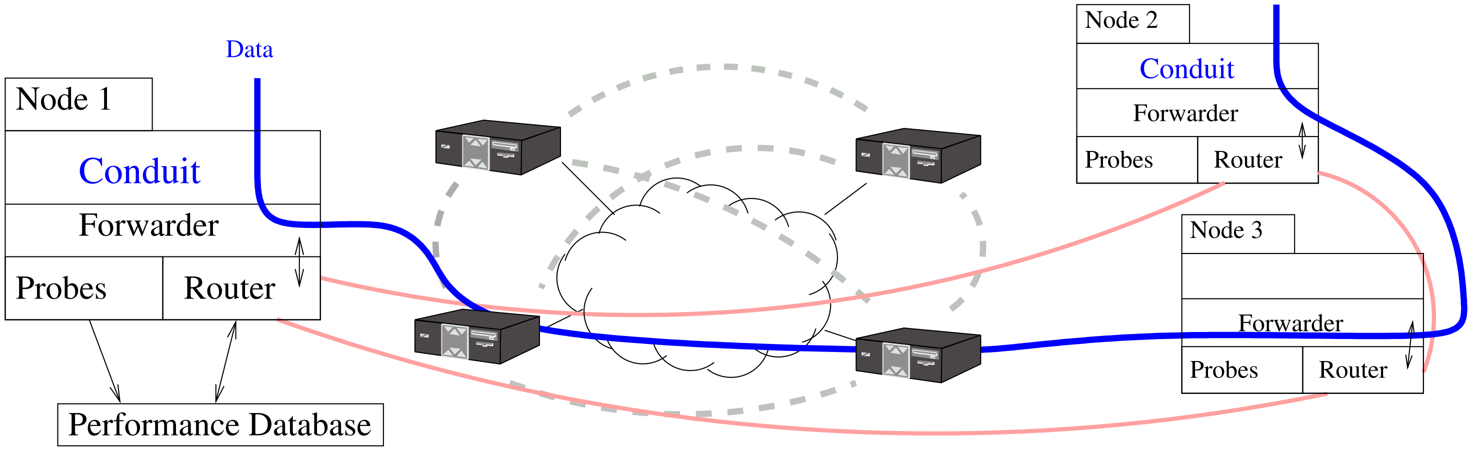
- Classify by policy
- Generate table per policy
- E.g. Internet2 Clique

Metric optimization

- App tags packets
  (e.g. "low latency")
- Generate one table per metric



# Architecture




- Probe between nodes, determine path qualities
- Link-state routing protocol between nodes
- Data handled by application-specific conduit (UDP)

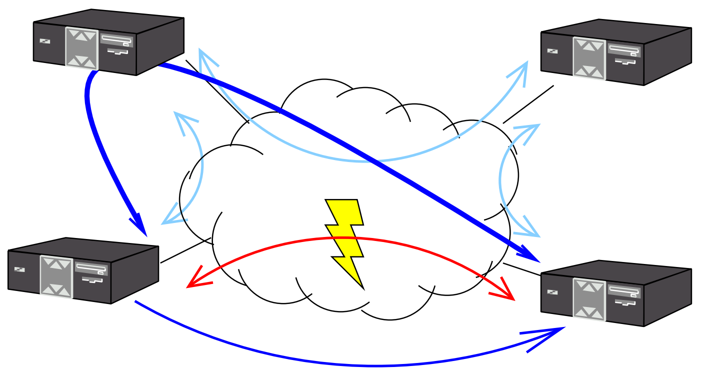
➔ Probing: Knowledge about network paths

➔ Forwarding: Control which path packets take

# 2-Redundant Multipath Routing

Packet duplication: simple FEC.
Choice of paths:

- Direct + Random
  (efficient)
- Random + Random
  (interesting)
- Use probe data
  (possibly better)



# Two Mechanisms

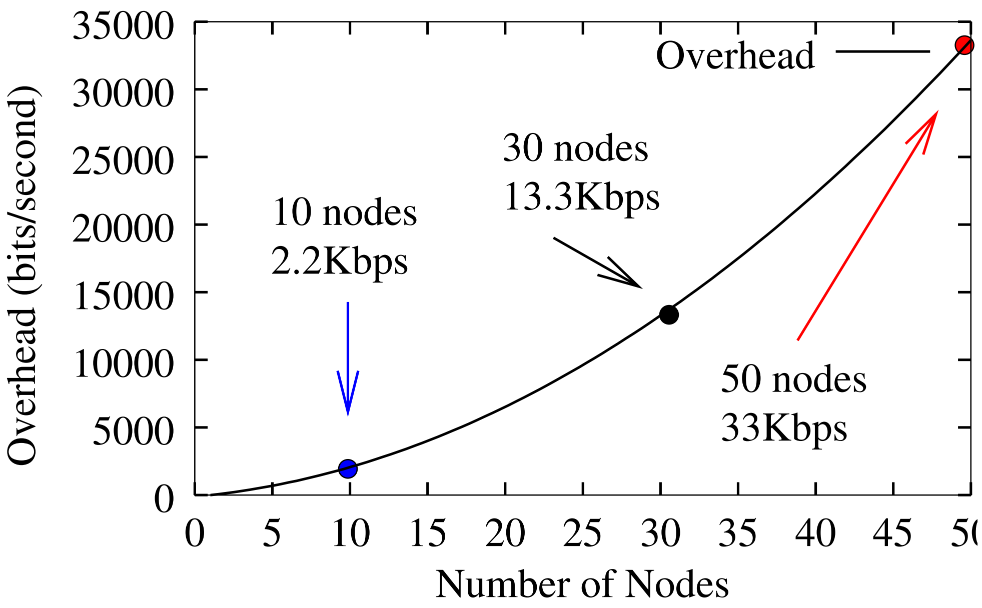
Best path vs. 2-Redundant. When to use which?

- Number of nodes scaling
- Responsiveness tradeoff
- Traffic volume

# Best Path Scaling

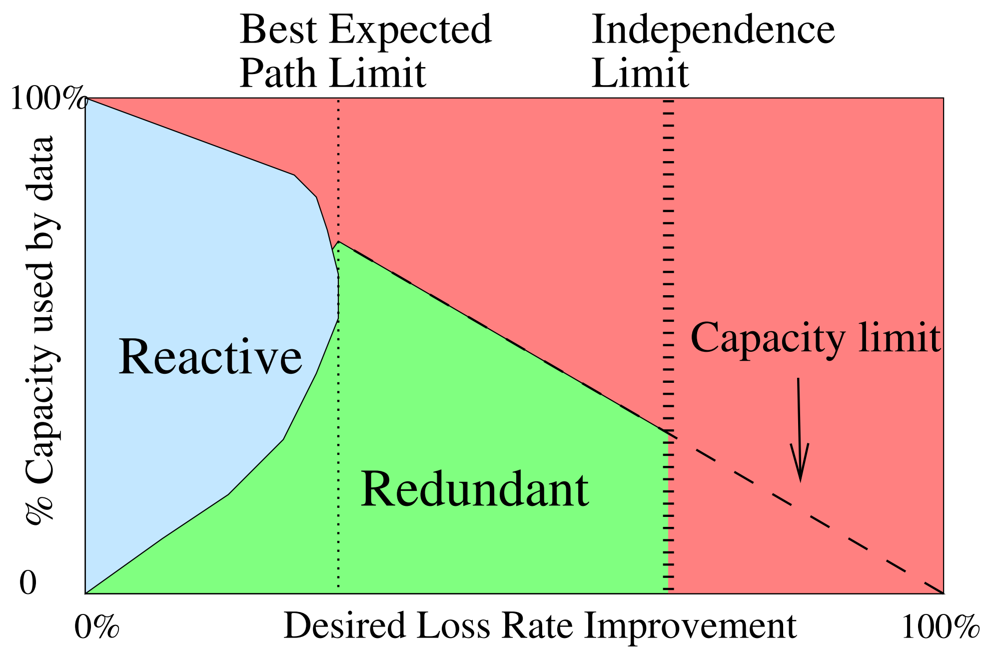
Routing and probing add packets:

Responsiveness vs. overhead vs. size




- 50 nodes pushes it, but is enough for many apps. 2-Redundant scales higher.

# Reactive vs. Redundant Routing




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

## Many Evaluation Questions

- Does the RON approach work?
  - How fast do we detect and avoid bad paths?
  - How many Internet outages are avoidable?
  - How does RON affect latency/throughput?
- How does best-path routing compare to redundant routing?

# Evaluation

Four datasets from Internet deployment

- $RON_1$: 12 nodes, 64 hours, Mar 2001
- $RON_2$: 16 nodes, 85 hours, May 2001
- $RON_{wide}$: 17 nodes, 5 days, Jul 2002
- $RON_{narrow}$: 17 nodes, 3 days, Jul 2002

US, Europe, Asia testbed of $\sim$ 20 nodes

- Variety of network types and bandwidths
- $N^2$ path scaling effect

# Evaluation Methodology

- Loss & latency. Each node repeats:
  1. Pick random node $j$
  2. Pick a probe type $(direct, loss,$ $direct + random, latency + loss)$ round-robin. Send to $j$
  3. Delay for random interval
- $RON_{wide}$ explored more probe types in less detail. $RON_1$ and $RON_2$ lacked mesh.

# Major Results

- ✔ Probe-based outage detection effective
  - RON takes ~10s to route around failure
    Compared to BGP's several minutes
  - Many Internet outages are avoidable
  - RON improves latency / loss / throughput
- ✔ Redundant routing equally or more effective
  - Avoids same outages
  - Reduces "baseline" loss rate more.

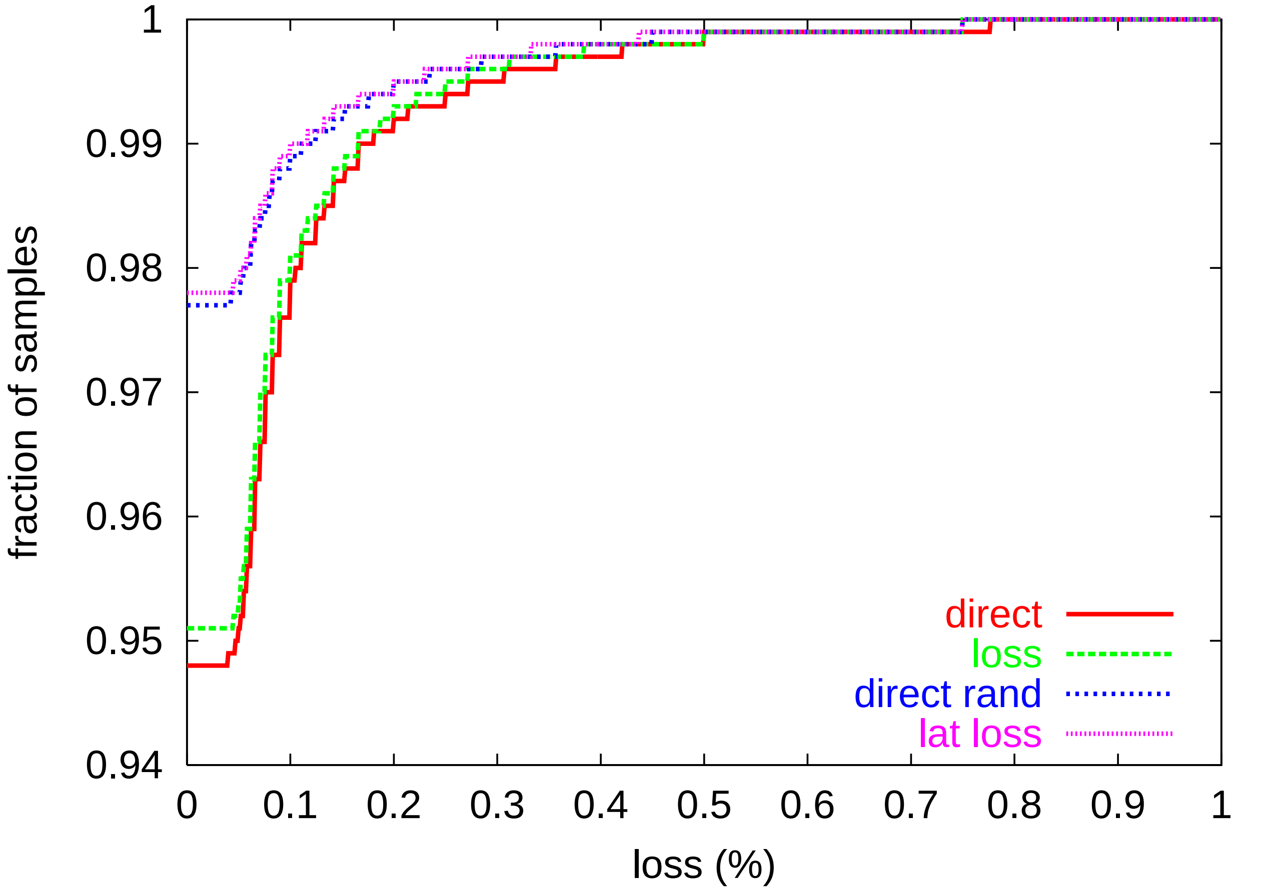
# $RON_1$ vs Internet 30 minute loss rates

| Internet Loss Rate | (0,10) | [10,20) | [20,30) | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| [90,100] | 12 | | | | | | | | | |
| [80,90) | 2 | | | | | | | | | |
| | 1 | | | | | | | | | |
| | 3 | 1 | | | | | | | | |
| | 1 | | | | | | | | | |
| | 3 | | | | | | | | | |
| | 8 | 1 | | | | | | | | |
| [20,30) | 87 | 8 | 4 | | | | | | | |
| [10,20) | 362 | 32 | 12 | | | | | | | |
| (0,10) | 2188 | 44 | 3 | | | | | | | |

RON loss rate

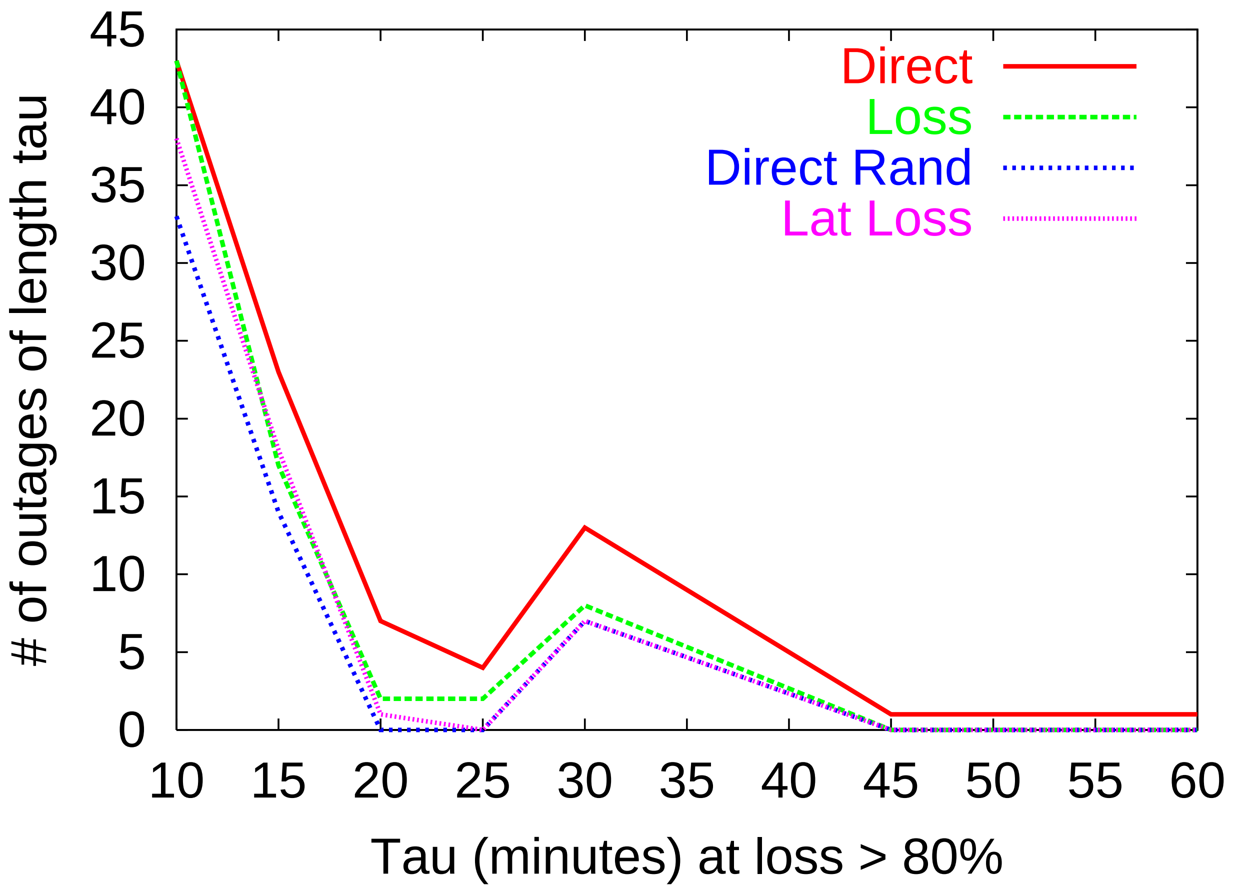
- 6,825 “path hours” (13,650 samples)

# $RON_{narrow}$ 10 minute loss rates




Low loss vs. high loss improvement

# $RON_{narrow}$ **Major** $> 80\%$ **Outages**



# Future Work

- Fundamentals
  - Internet scalability / resilience trade-off
- Scaling
  - How big? What tactics?
  - Interacting RONs? Stability?

# Conclusions

- ✔ Control over resiliency allows mechanism to match application needs.Best Path and Redundant each good for different traffic mix.
- ✔ Overlays attractive spot for resiliency: development, fewer nodes, simple substrate
- ➜ RON libraries are good platform for development, research

Lots of interesting work remains!

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