How I Learned to Stop Worrying and Love to Spoof

Ethan Katz-Bassett, Harsha V. Madhyastha, Arvind Krishnamurthy, Thomas Anderson

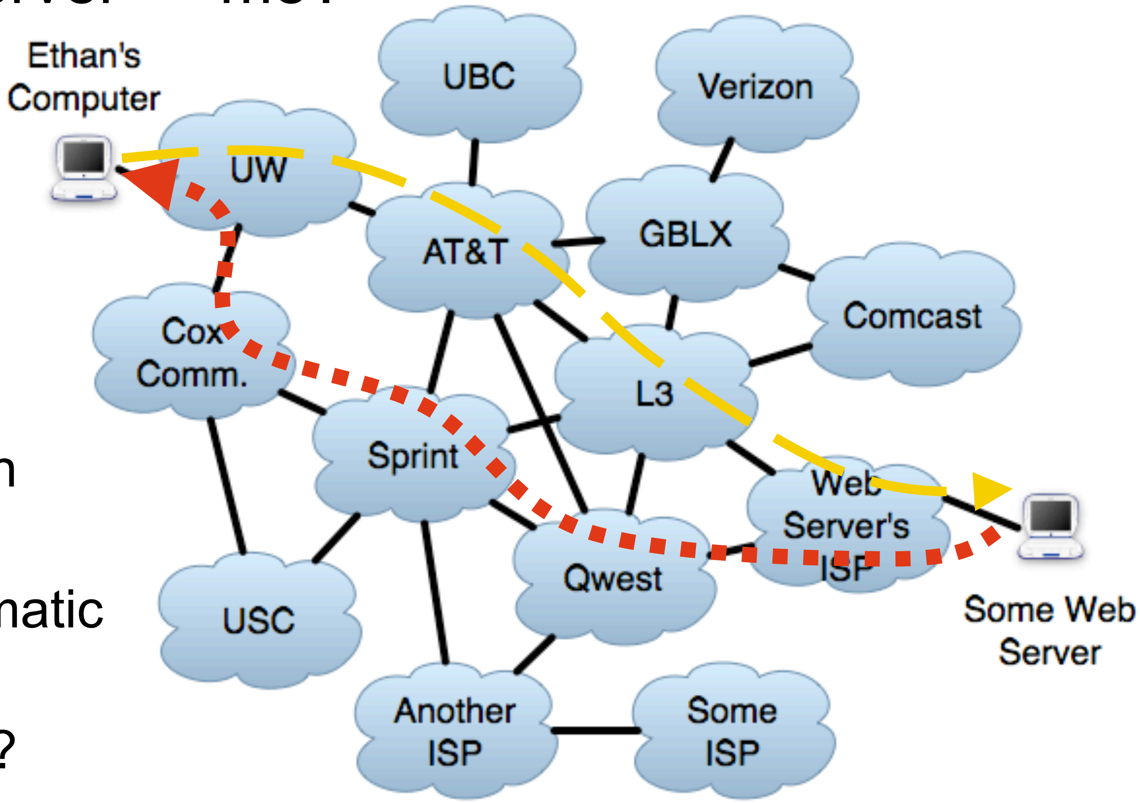
AIMS 2009, January 2009

This work partially supported by Cisco, Google, NSF

Probing One Direction of a Path

How to probe path server ⇒ me?

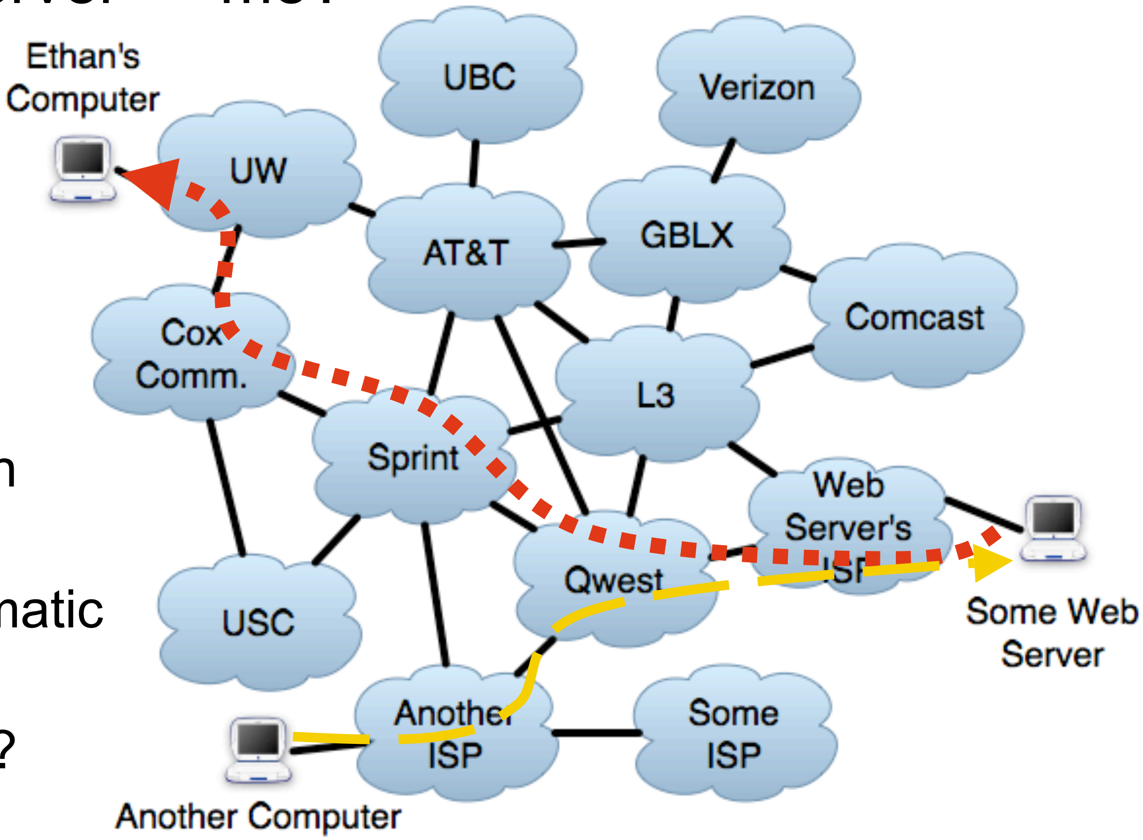
- Probe from server
  - What if we don't control it?
- Round-trip probe both directions
  - What if forward path is broken?
  - Or contains problematic ASes/ routers?
  - Or lacks properties?
  - Or we want to differentiate forward from reverse?



Probing One Direction of a Path

How to probe path server ⇒ me?

- Probe from server
  - What if we don't control it?
- Round-trip probe both directions
  - What if forward path is broken?
  - Or contains problematic ASes/ routers?
  - Or lacks properties?
  - Or we want to differentiate forward from reverse?
- Spoof as me from another vantage point



Spoofing as another vantage point

- We use restricted version that is perfectly safe
  - Only spoof as nodes we control
    - Like a “reply to” address
    - Send from a vantage point to another, through destination
  - Millions of spoofed probes sent to 100s of thousands of IPs, no complaints
- Lets us approximate:
  - Having control of destinations
  - One-hop loose source routing

Outline

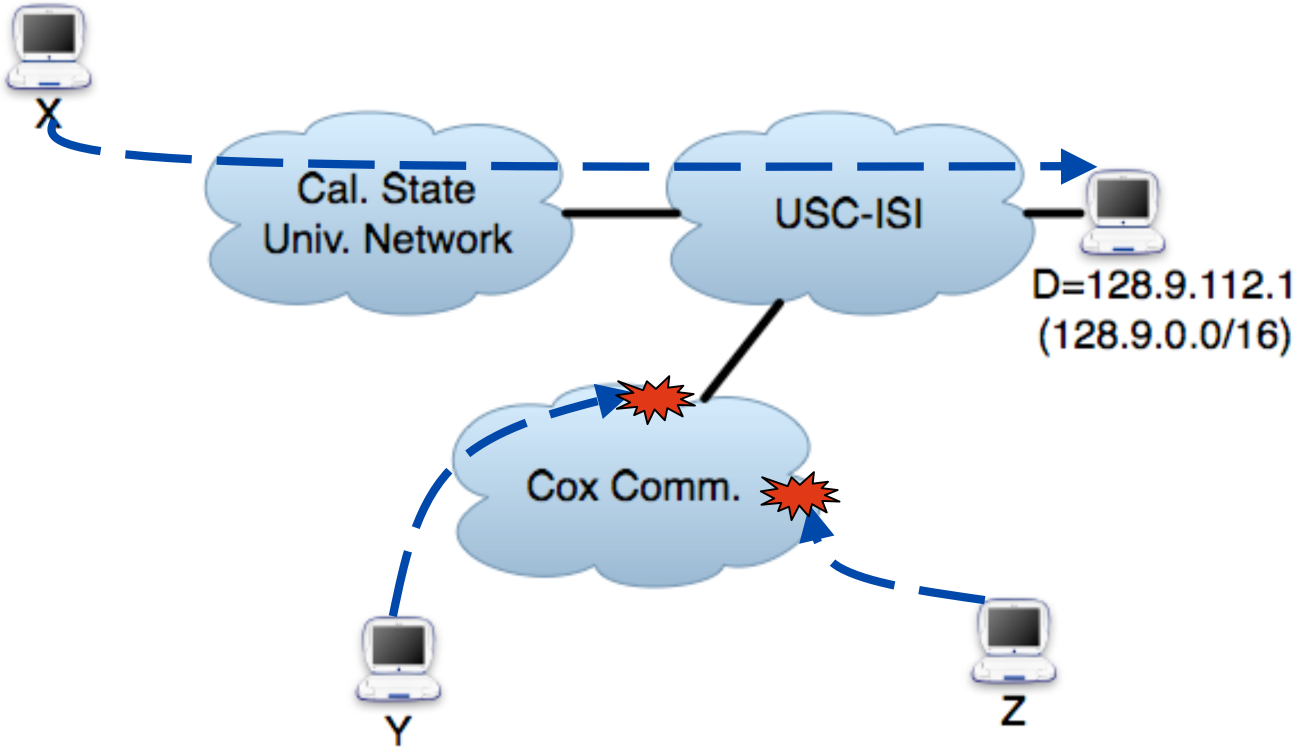
- *Spoofing lets us probe on direction of path*
- Examples of spoofing to probe one direction
  - Isolate direction of failure
  - Reverse traceroute
    - Application: One-way latency
- Discussion of spoofing
  - Operators and ISPs
  - Testbeds and how to spoof without complaints

Example 1: Isolate direction of failure

```
traceroute to 18.0.0.1 (18.0.0.1), 64 hops max, 40 byte packets
 1   128.208.3.102   0.710 ms 0.291 ms 0.275 ms
 2   205.175.108.21 0.489 ms 0.648 ms 0.273 ms
…
 9   216.24.186.33   74.425 ms  73.705 ms  73.820 ms
10   216.24.184.102 73.218 ms  73.274 ms  73.228 ms
11   * * *
12   * * *
13   * * *
```

- With traceroute, forward and reverse path failures look the same

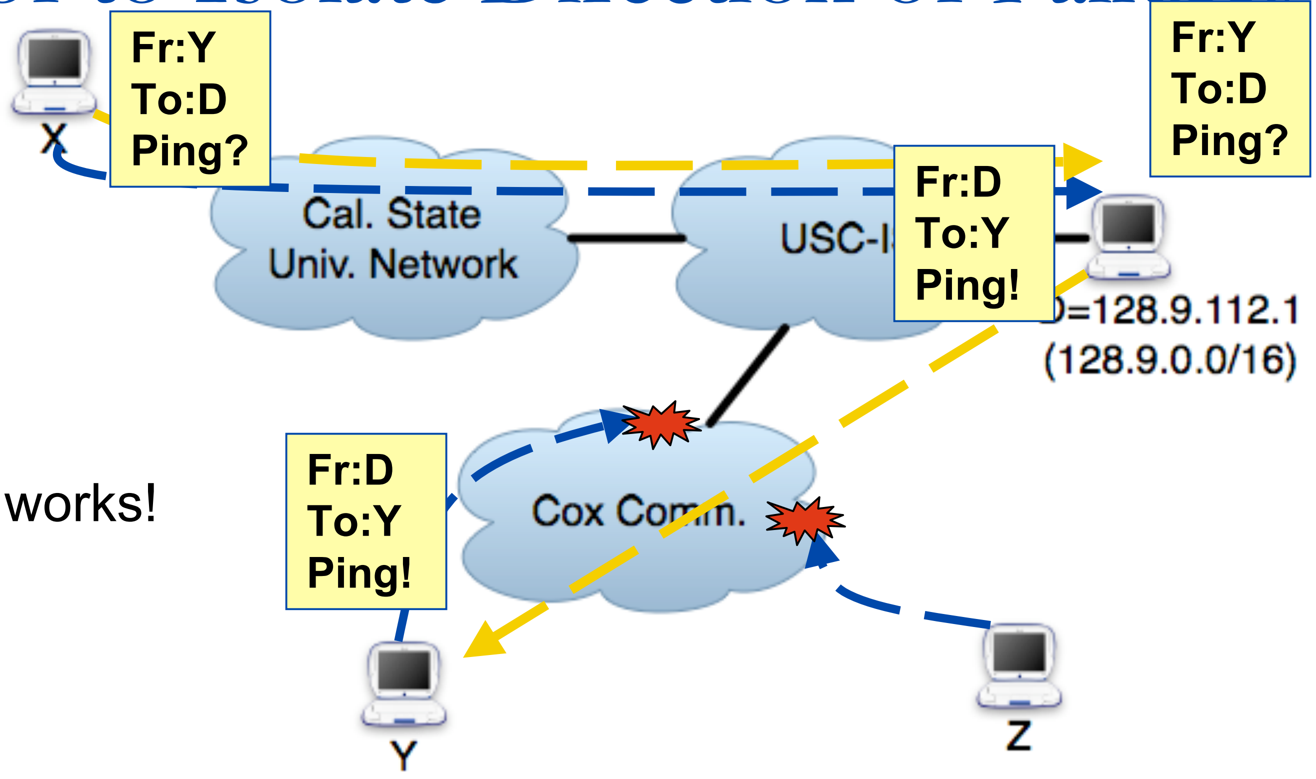
Spoof to Isolate Direction of Failures



Example seen by **Hubble** on October 8, 2007

1. Determine location of failure
   a) Failed traceroutes suggest problem with Cox
      … but could actually be on (asymmetric?) reverse path

Spoof to Isolate Direction of Failures

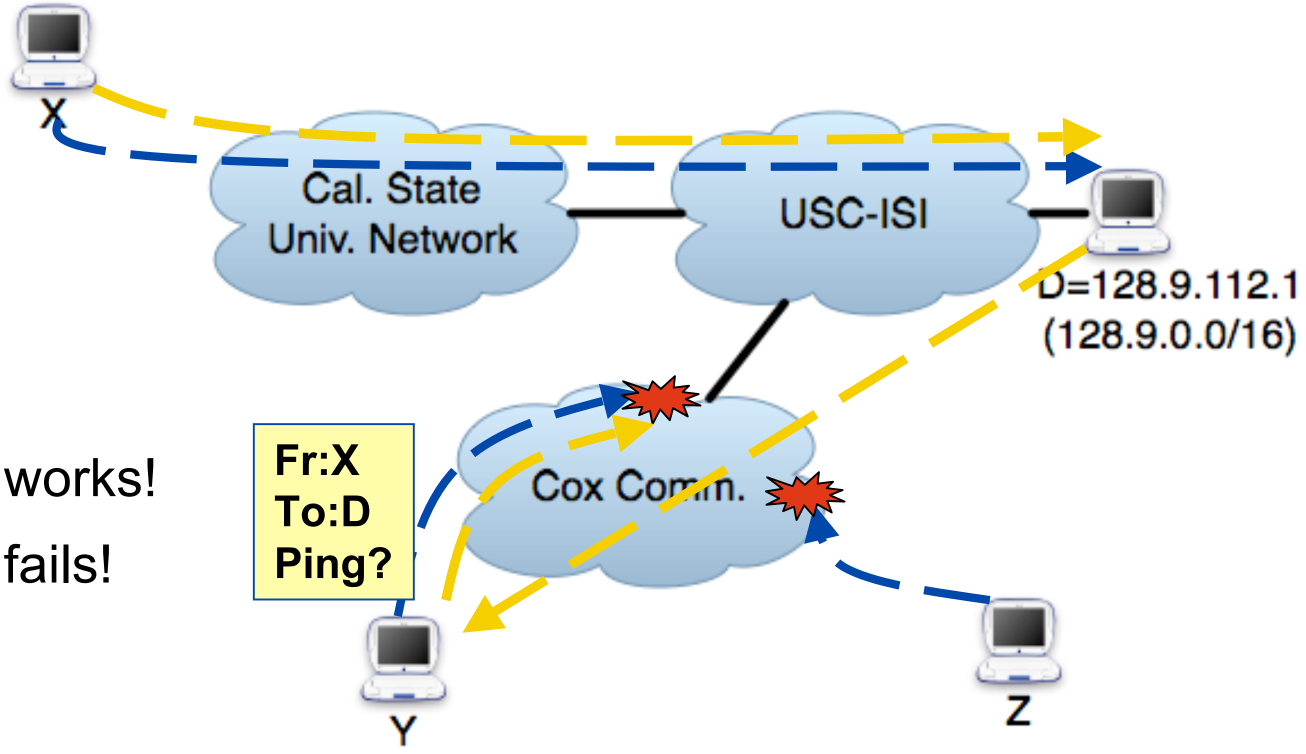


D to Y works!

Example seen by **Hubble** on October 8, 2007

1. Determine location of failure
   a) Failed traceroutes suggest problem with Cox
      … but could actually be on (asymmetric?) reverse path
   b) Spoofed pings isolate problem to one direction

Spoof to Isolate Direction of Failures



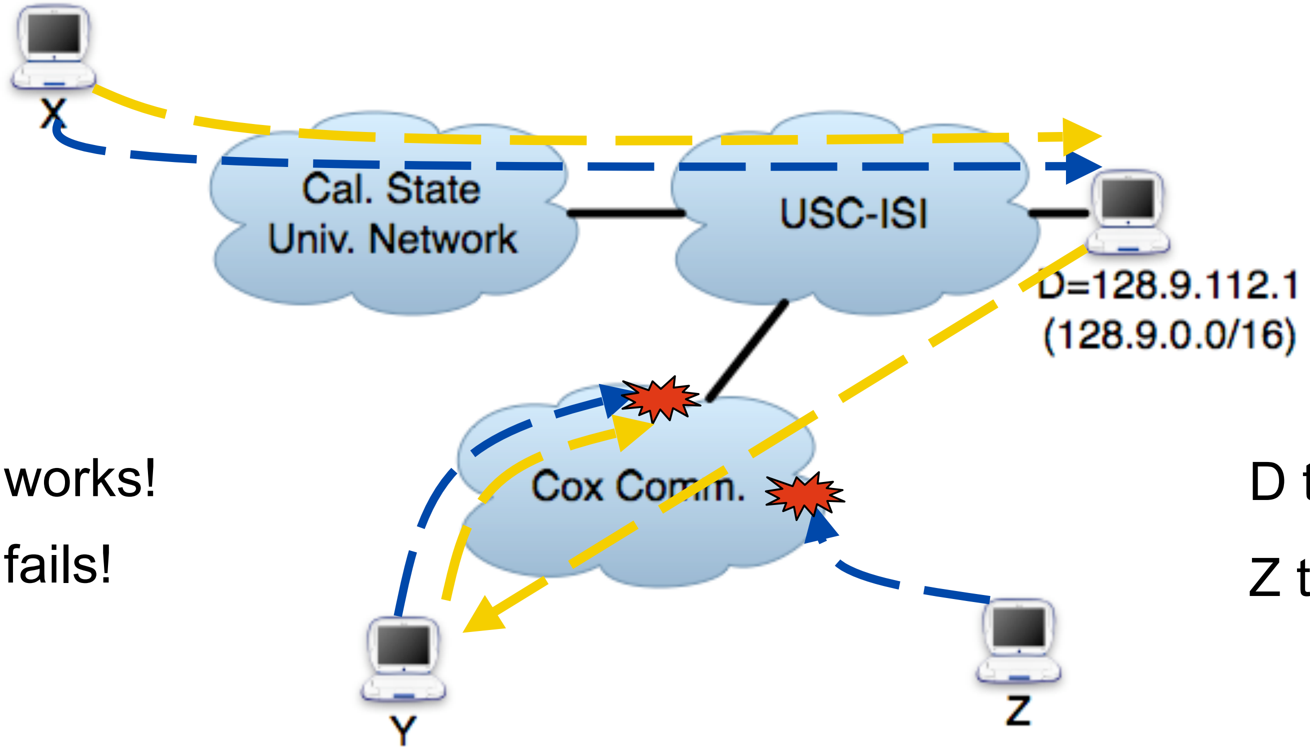
D to Y works!

Y to D fails!

Example seen by **Hubble** on October 8, 2007

1. Determine location of failure
   a) Failed traceroutes suggest problem with Cox
      … but could actually be on (asymmetric?) reverse path
   b) Spoofed pings isolate problem to one direction

Spoof to Isolate Direction of Failures



D to Y works!

Y to D fails!

D to Z works!

Z to D fails!

Example seen by **Hubble** on October 8, 2007

1. Determine location of failure
   a) Failed traceroutes suggest problem with Cox
      … but could actually be on (asymmetric?) reverse path
   b) Spoofed pings isolate problem to one direction

How often can we isolate direction?

Results from 3 week study with **Hubble**

- 68% of black holes are partial
- Isolate failure direction in 68% of these cases

Hundreds of problems involve multi-homing

- Like COX example, one provider works, another not successfully forwarding traffic
- 6% of classified problems

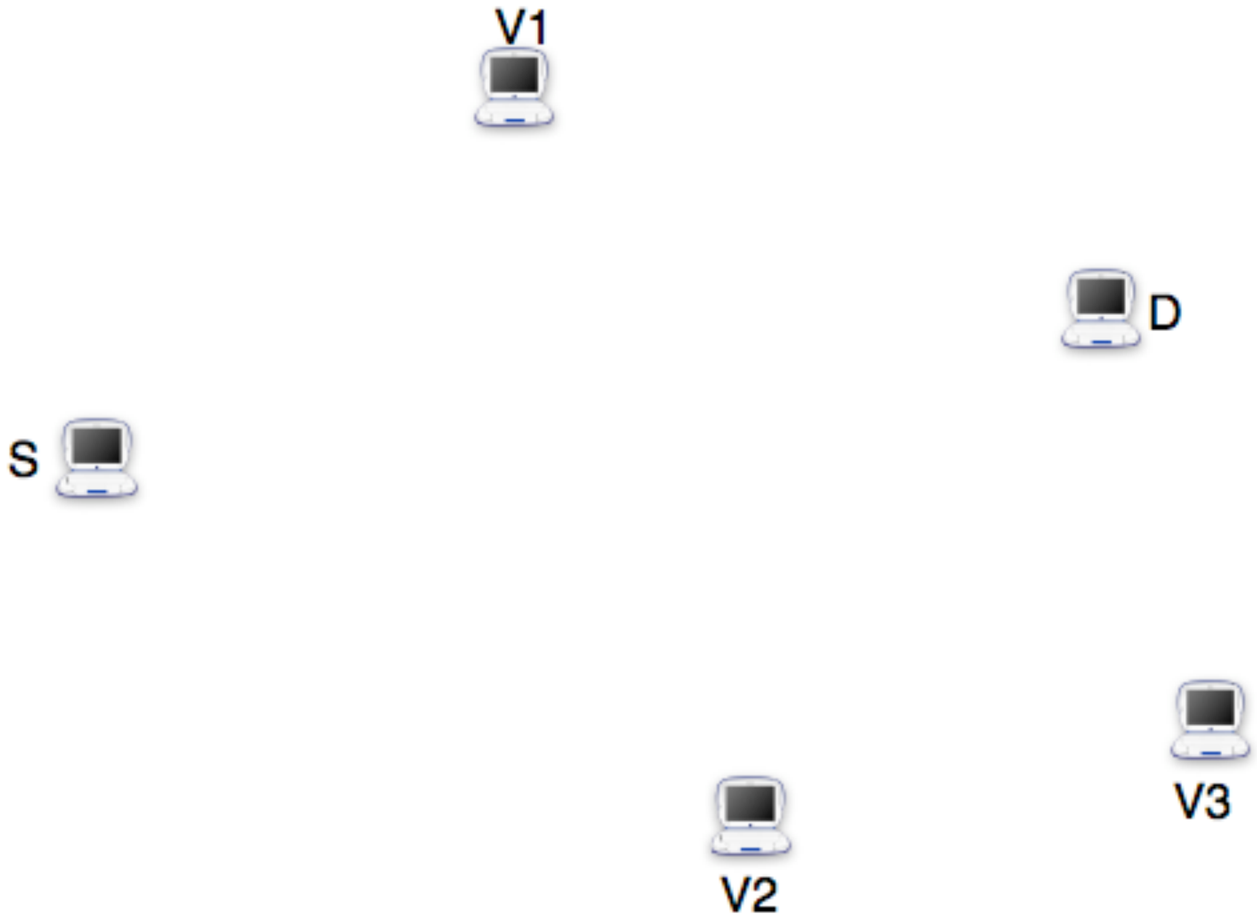
Example 2: Reverse Traceroute

“The number one go-to tool is traceroute.
The number one plague of traceroute
[is path asymmetry, because]
the reverse path itself is completely invisible”

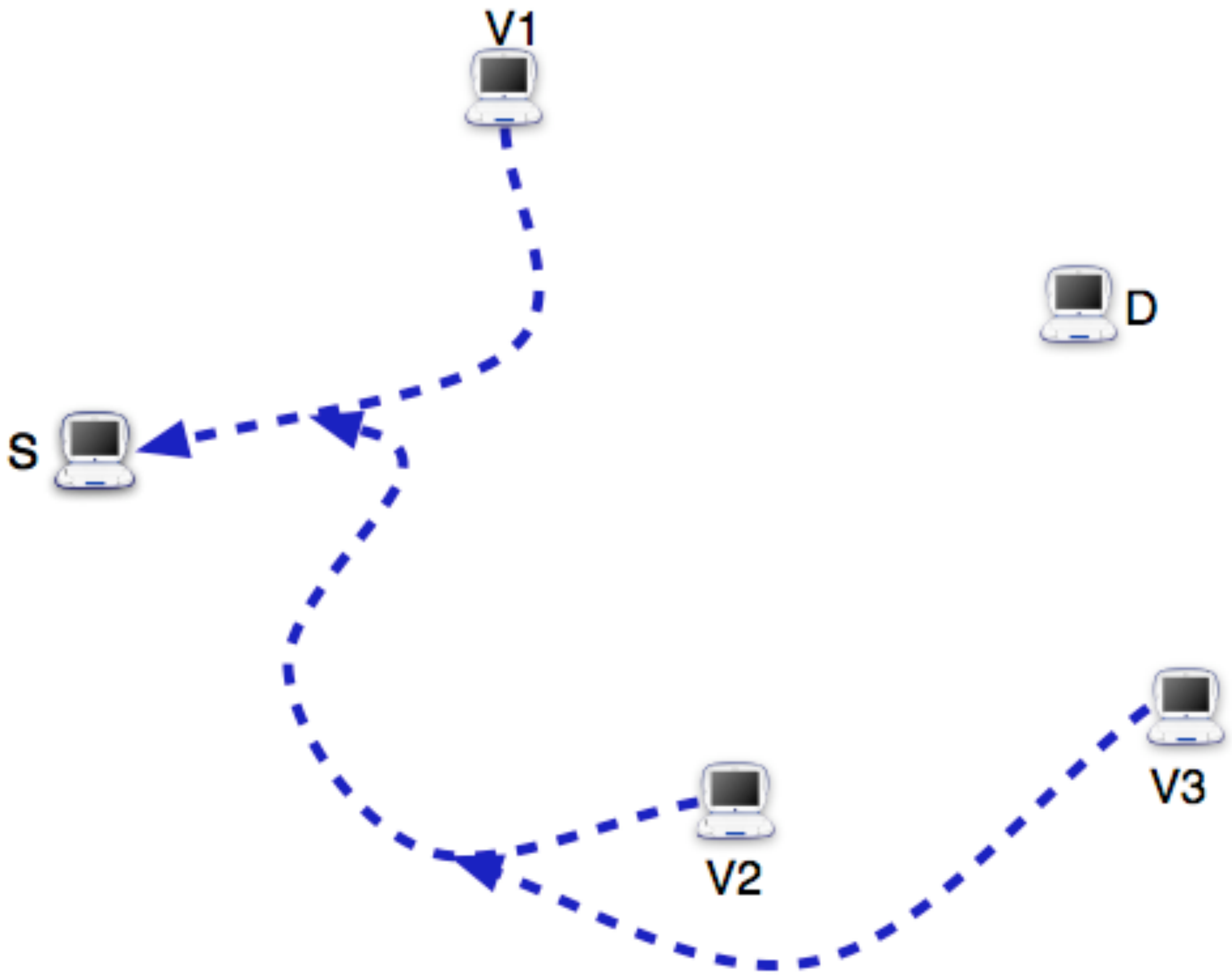
Richard Steenbergen
CTO, nLayer Communications
Troubleshooting tutorial
NANOG 45, January 2009

IP Options to Identify Reverse Hops

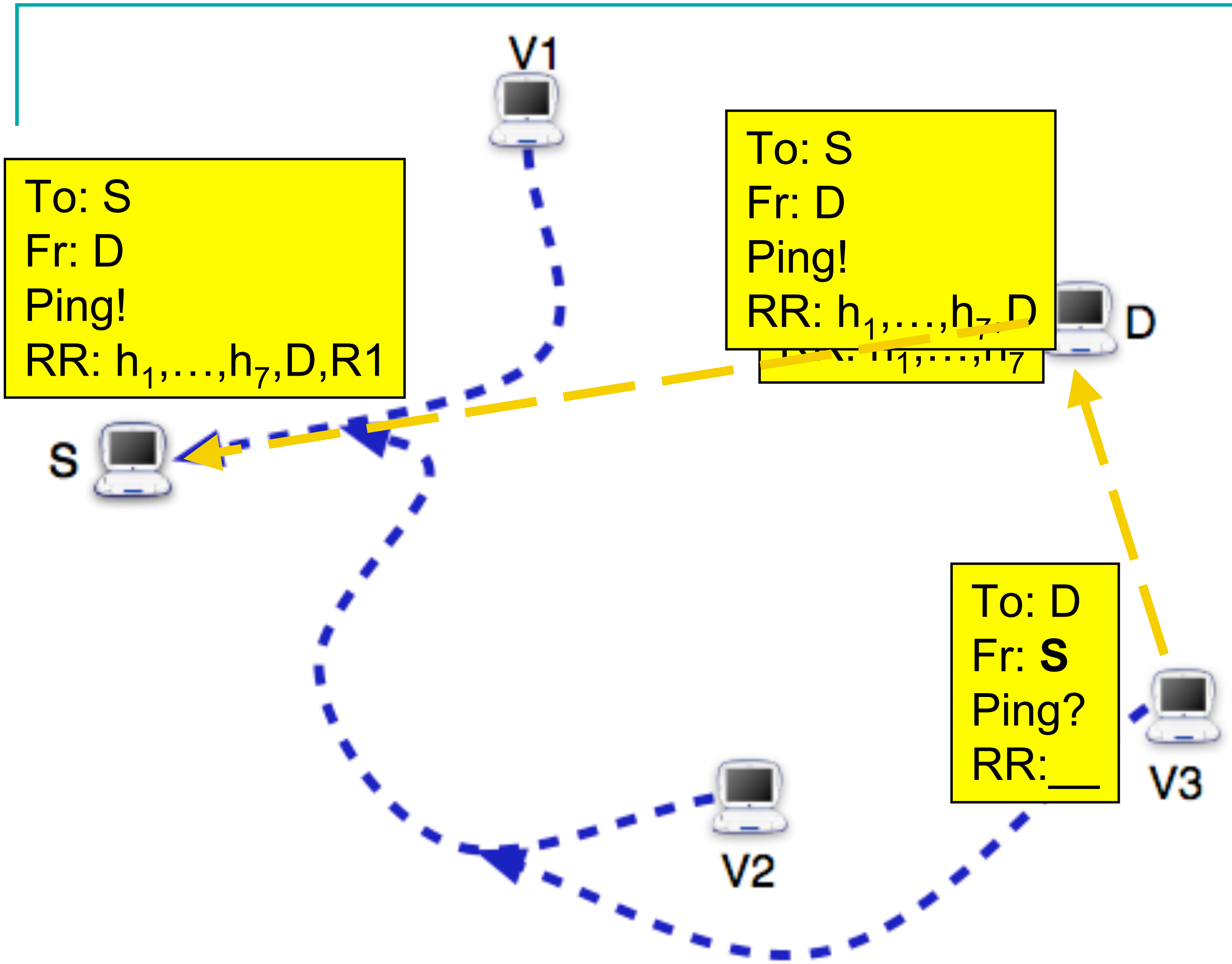
- Unlike TTL, *IP Options* reflected in reply, so work on forward and reverse path
- *Record Route (RR)* option
  - Record first 9 routers on path
  - If destination within 8, reverse hops fill rest of slots
  - … but average path is 15 hops, 30 round-trip
- If vantage point within 8 hops, probe from there spoofing as source to gather reverse hops



- Want reverse path from **D** back to **S**, but don’t control **D**
- Set of vantage points, some of which can spoof



- Traceroute from all vantage points to **S**
- Gives atlas of paths to **S**; if we hit one, we know rest of path



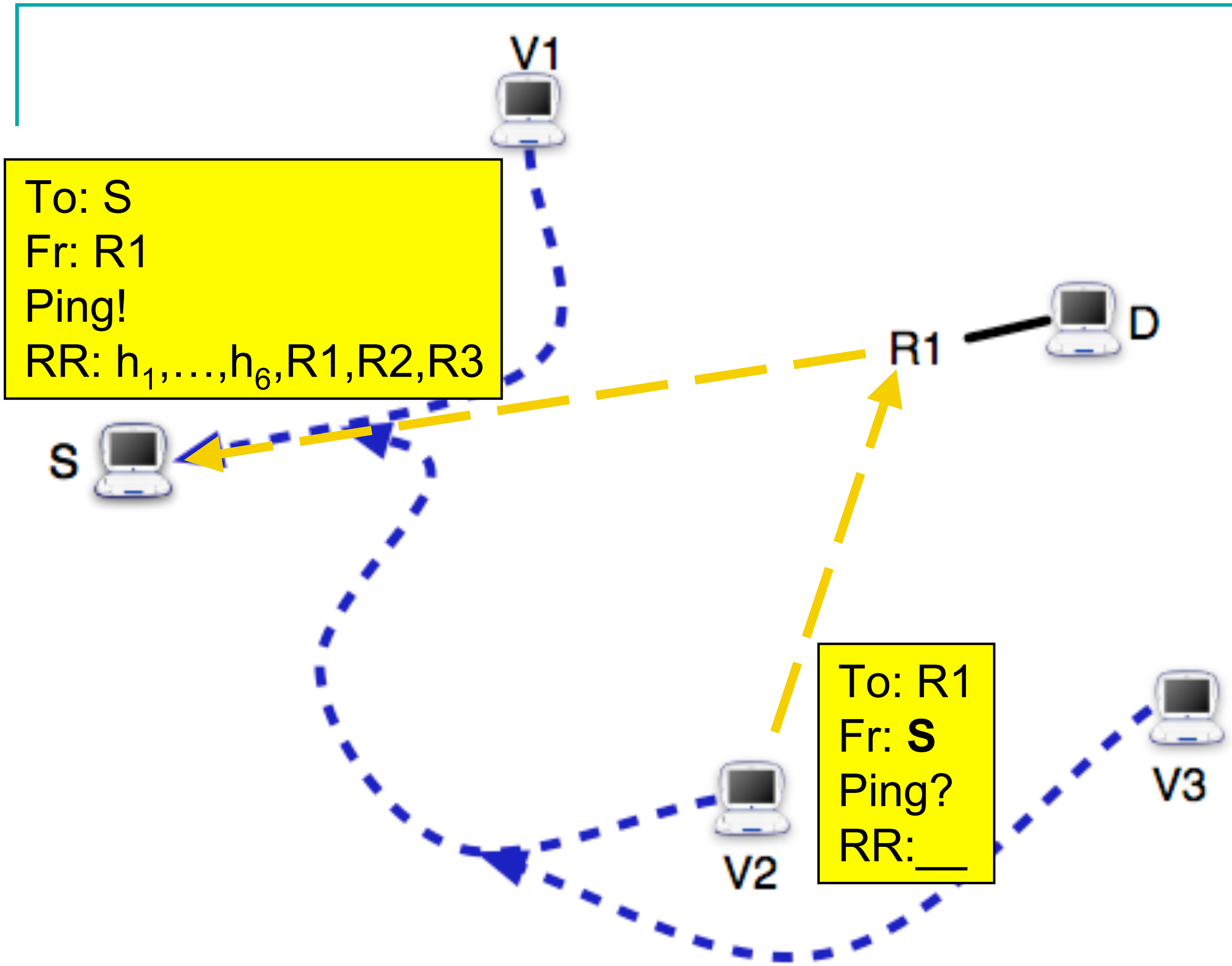
To: S
Fr: D
Ping!
RR: h_1,…,h_7,D,R1

To: S
Fr: D
Ping!
RR: h_1,…,h_7,D

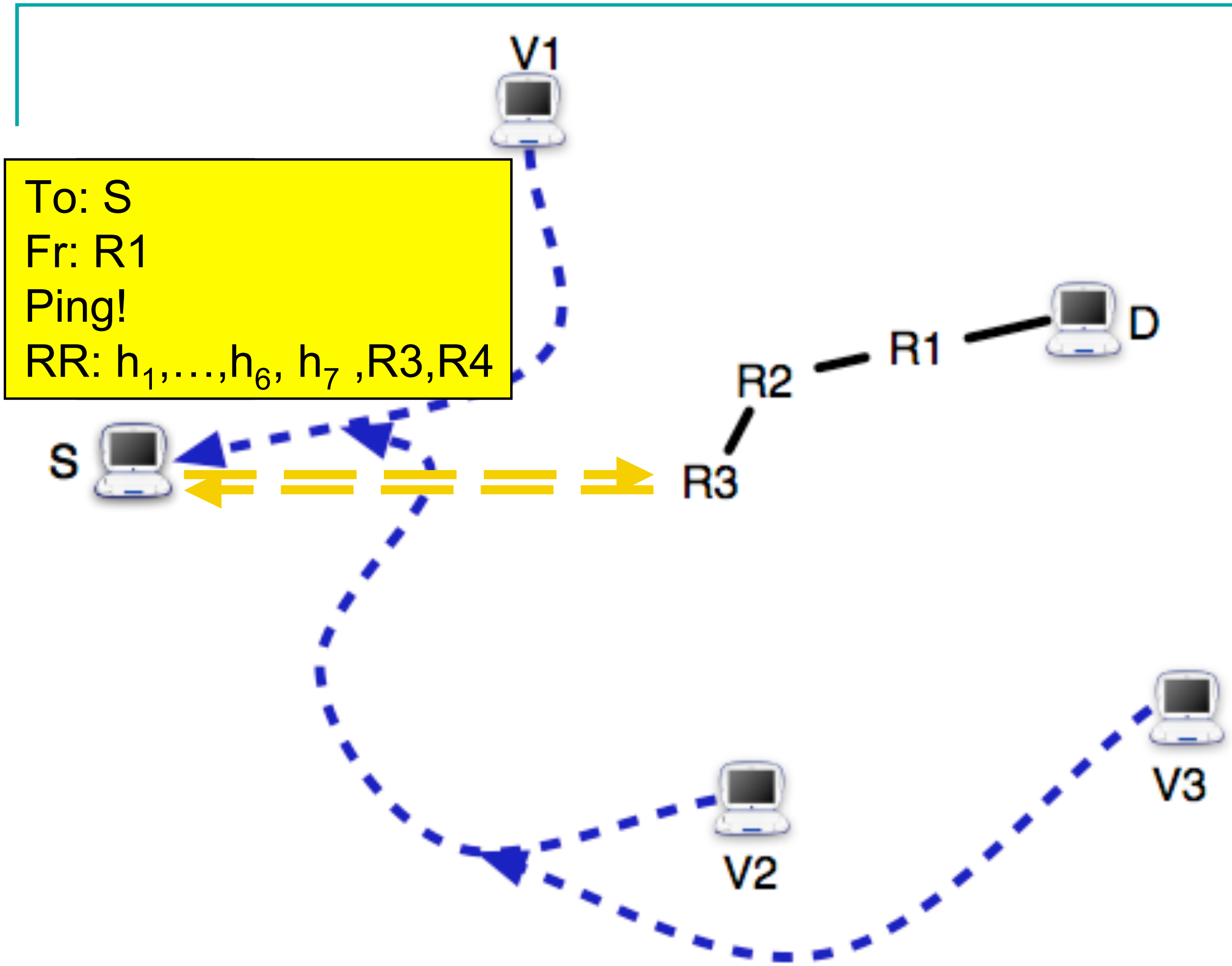
RR: h_1,…,h_7

To: D
Fr: **S**
Ping?
RR:__

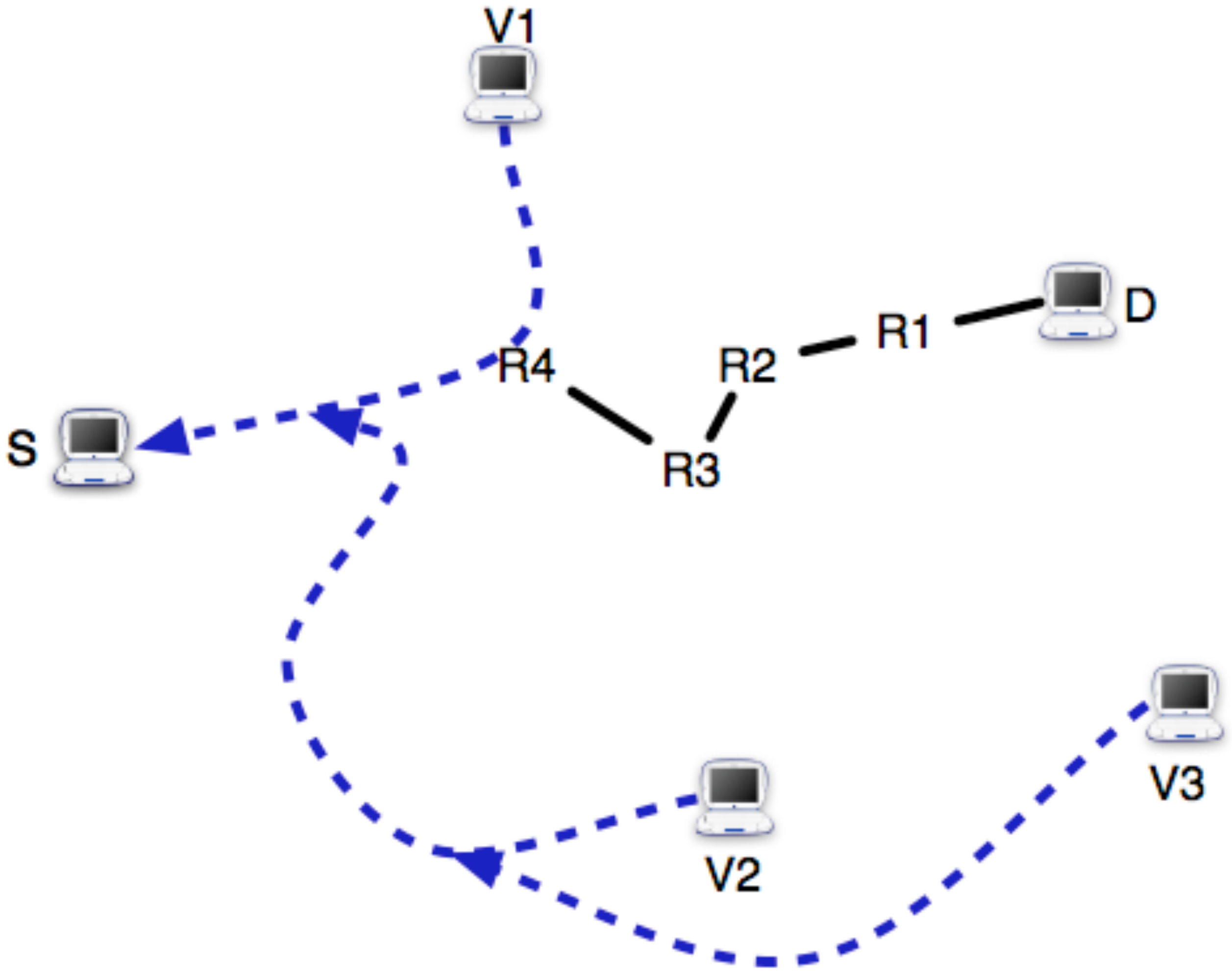
- From vantage point within 8 hops of **D**, ping **D** spoofing as **S** with record route option
- **D**’s response will contain recorded hop(s) on return path

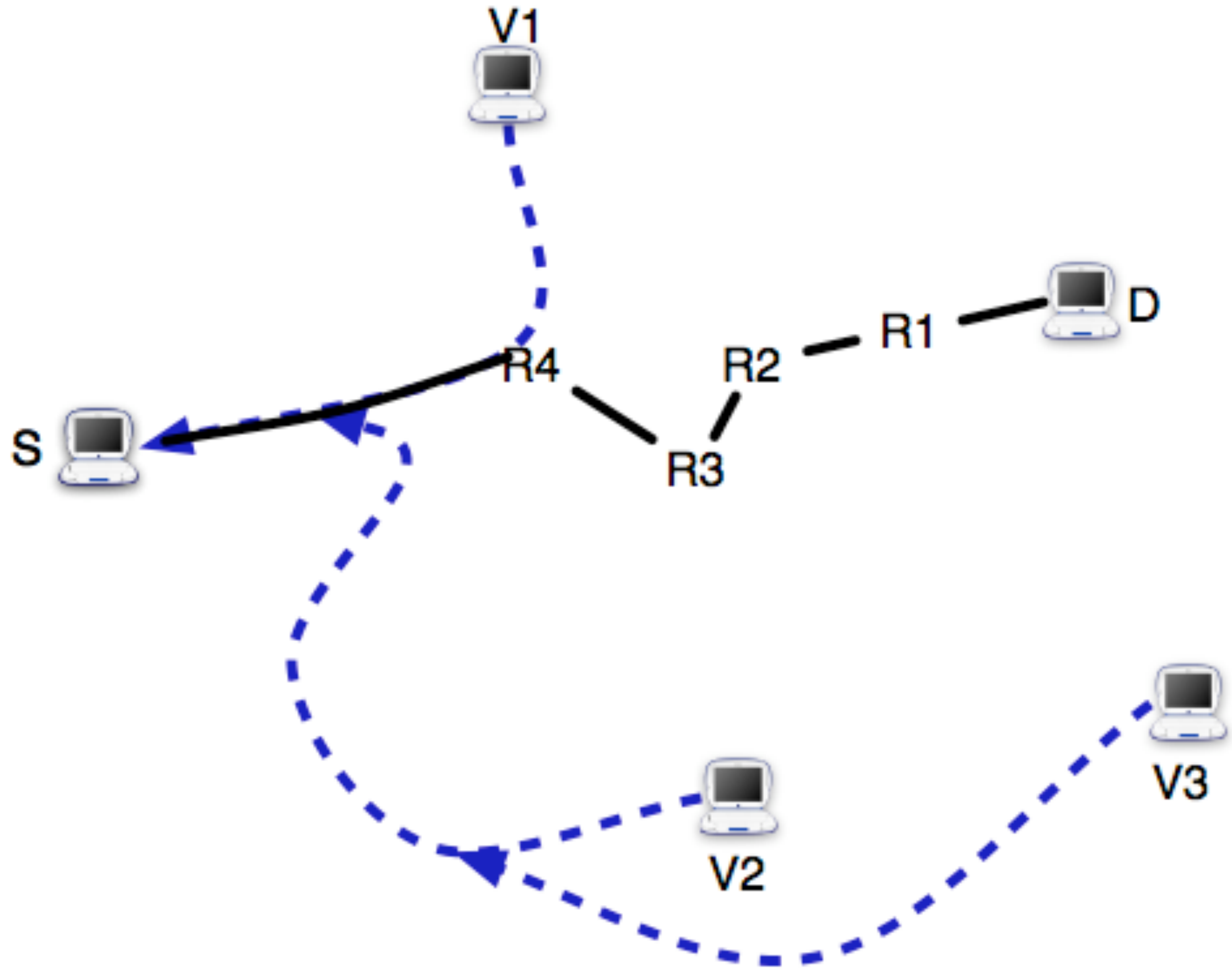


- Iterate, performing TTL=8 pings and spoofed RR pings for each router we discover on return path

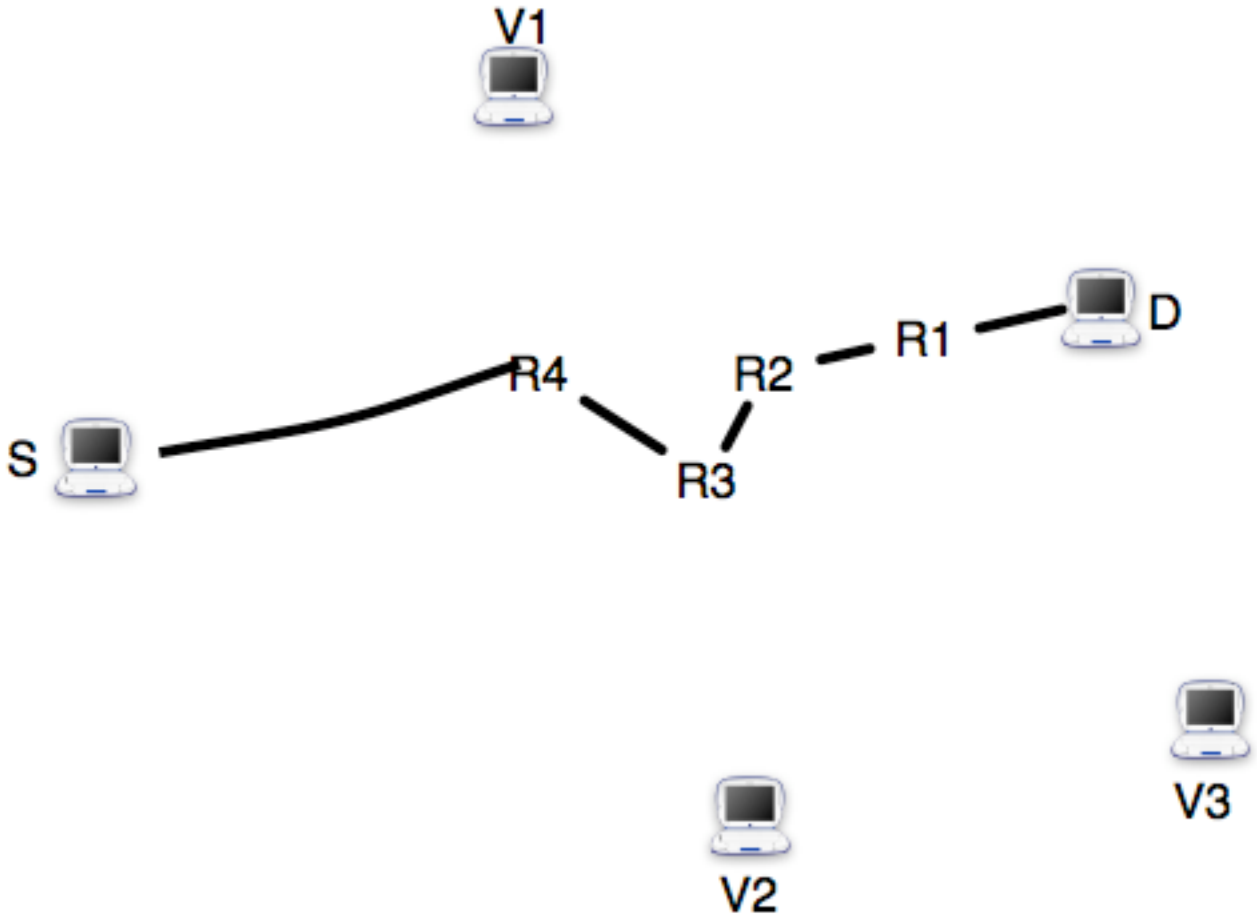


To: S
Fr: R1
Ping!
RR: $h_1, \ldots, h_6$, h_7 ,R3,R4


V1
D
R1
R2
R4
S
R3
V3
V2

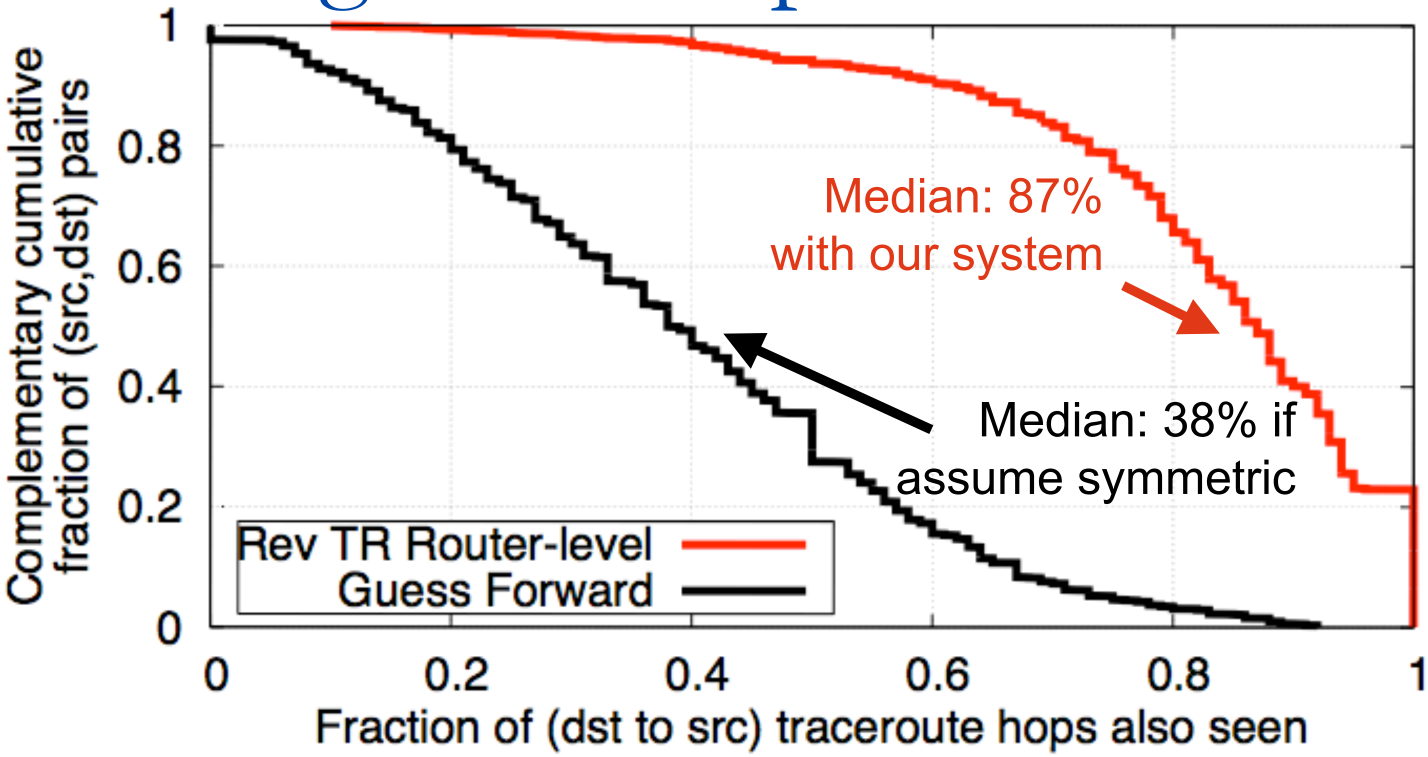


- Once we see a router on a known path, we know remainder



- Techniques combine to give us complete path
- We have additional techniques for inferring reverse hops

Does it give same path as traceroute?

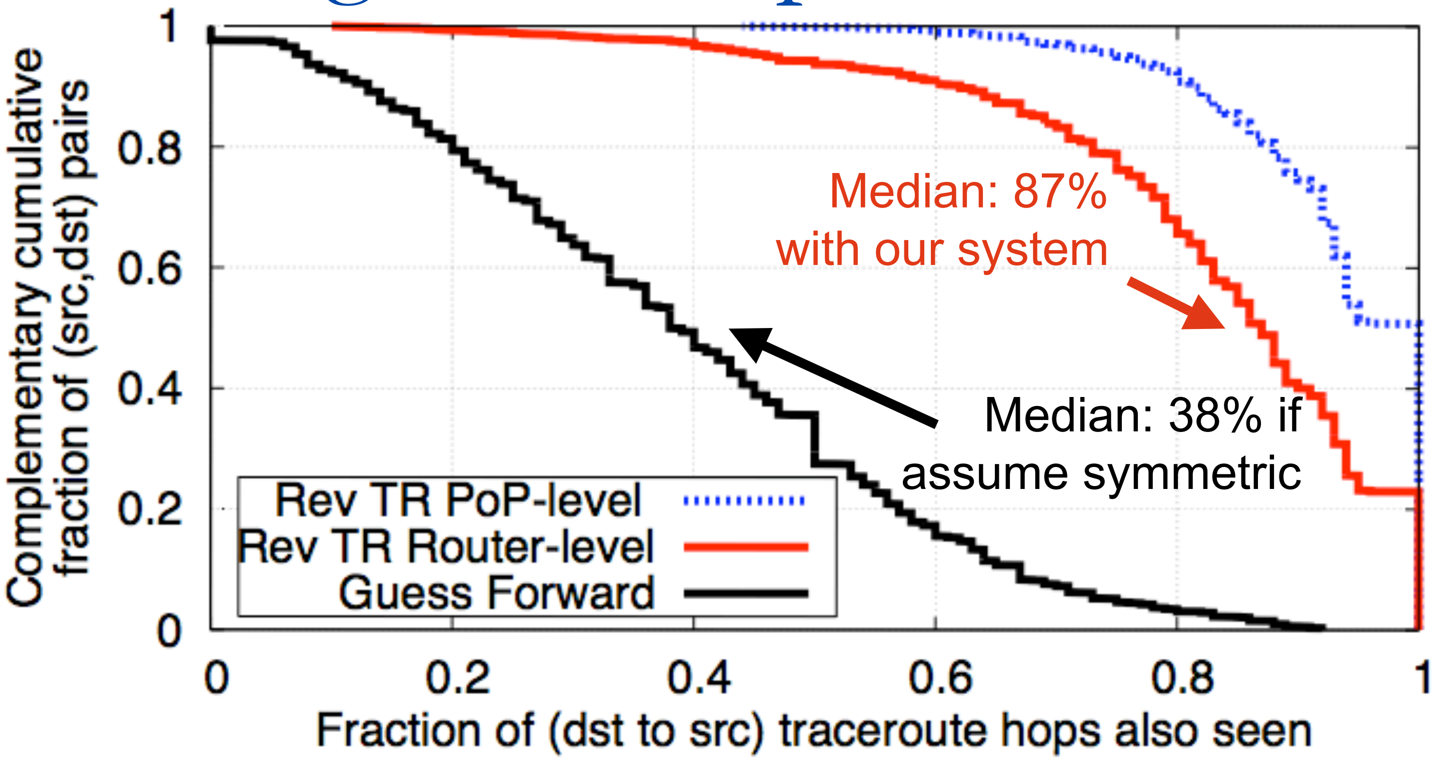


Median: 87% with our system

Median: 38% if assume symmetric

- 200 PlanetLab destinations, where we can directly traceroute "reverse" path
- Usually identify most hops seen by traceroute
- Hard to know which interfaces are on the same router

Does it give same path as traceroute?



Median: 87% with our system

Median: 38% if assume symmetric

- 200 PlanetLab destinations, where we can directly traceroute “reverse” path
- Usually identify most hops seen by traceroute
- Hard to know which interfaces are on the same router
  - If we consider PoPs instead, median=100% accurate

Applications of Reverse Traceroute

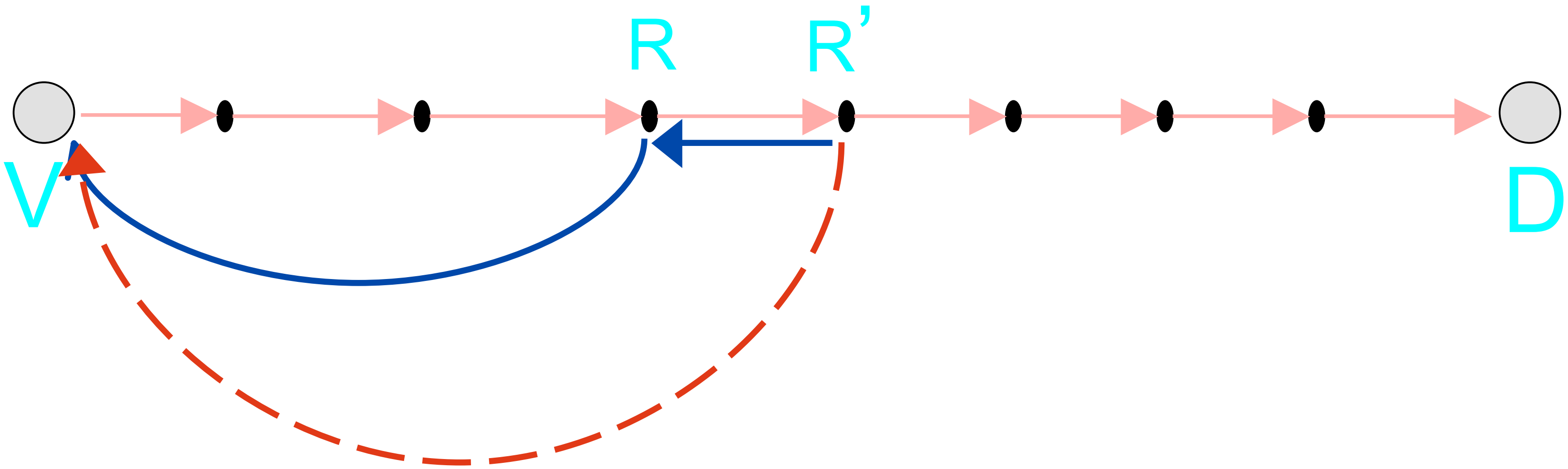
- Debugging path inflation
- Troubleshooting unreachability
- Topology discovery
  - Especially of hidden peer-to-peer links
- One-way link latency/ tomography

- More we have not looked at yet

Reverse Tracroute Application: Measure One-way Latency

- Traceroute/ping give round-trip time (RTT)
- … but many apps want one-way link latency
  - Troubleshooting poor performance
  - Latency estimation (iPlane)
  - ISP comparison (Netdiff)
  - Geolocation (Octant, TBG)

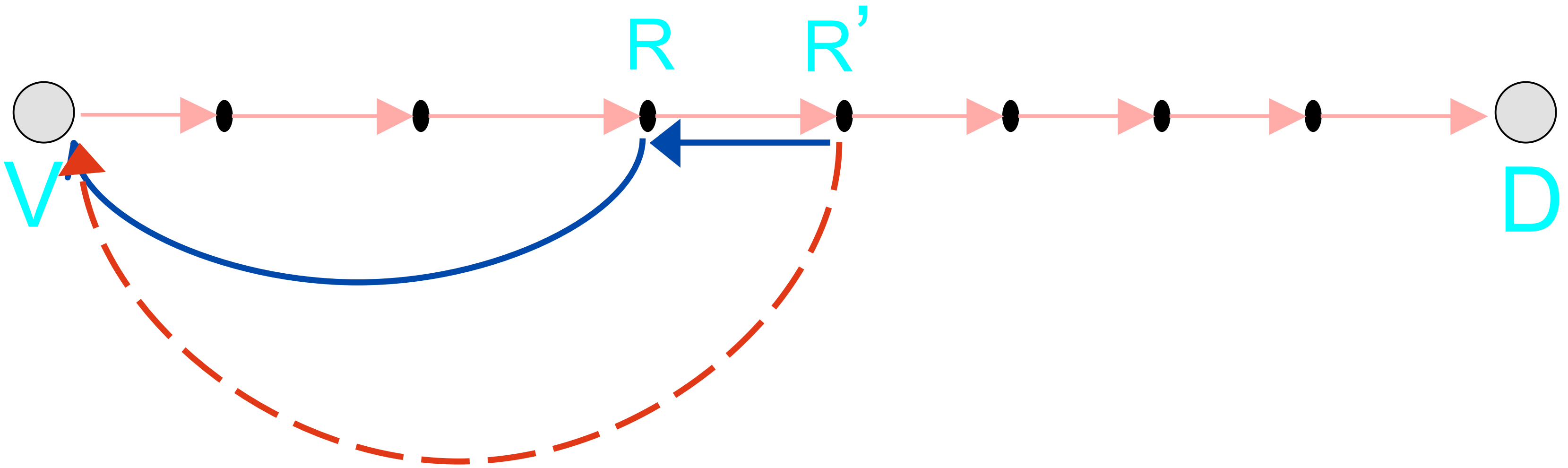
Measuring Link Latency



- Straightforward approach:

  Latency(R, R') = (RTT(V, R') – RTT(V, R)) / 2
- Asymmetry skews link latency inferred from traceroutes

Reverse Traceroute Detects Symmetry



- Reverse traceroute identifies symmetric traversal
  - Identify cases when we can use RTT difference
  - Many links traversed symmetrically from some vantage points, not others

Reverse TR Constrains Link Latencies

- Build up system of constraints on link latencies to intermediate routers
  - Traceroutes and reverse traceroutes to all hops
  - TR Links + Reverse TR Links = RTT
- Preliminary study: 10 PlanetLab site mesh
  - 280 links in initial mesh, 917 with intermediate paths
  - 221 of 280 links bound and solvable by constraints
  - No ground truth makes verification hard. Ideas?
  - For 61 intra-PoP links, gives latencies < 0.7ms, consistent with expectations
- Similar approach applies to other tomography

Outline

- *Spoofing lets us probe on direction of path*
- *Examples of spoofing to probe one direction*
  - *Isolate direction of failure*
  - *Reverse traceroute*
    - *Application: One-way latency*
- Discussion of spoofing
  - Operators and ISPs
  - Testbeds and how to spoof without complaints

Operator Response to Spoofing

- NANOG thread about our use of spoofing
  - Bill Manning (USC-ISI) was not such a big fan
  - "Great work on a tough problem."
    Randy Bush (IIJ), NANOG mailing list
- Providing tools/ services encourages support for techniques
  - **Hubble** presented at RIPE meeting
  - Reverse TR presented at NANOG meeting
- Operators donated hosts to the systems, including all PoPs of an international backbone

Spoofing and ISPs

- Rate limit options and spoofed packets
- Restrict destinations (no broadcast IPs)
- Only requires small number of spoofing vantage points and ports
  - Can filter everywhere else

These restrictions limit malicious uses of spoofing while enabling measurement uses

Spoofing and Testbeds

- Against PlanetLab AUP
  - Evaluating limited access
- But useful, so safe support by:
  - Encouraging sites to allow
  - Vetting experiments/ experimentors
  - Filtering/ rate-limiting
  - Only spoof as other testbed sites?

How to Spoof Without Complaints

- Standard measurement best practices
  - Issue measurements locally first
  - Ramp up # sources, destinations, rate slowly
  - Careful probing endhosts
- Start by verifying which sites allow spoofing
- Only spoof as a machine you control
- Issue an equivalent non-spoofed probe first

Conclusions

- Spoofing useful
- Possible to do it safely and without complaints
  - Also possible to screw it up for everyone
- When you might use it (example app)
  - Round-trip path broken (isolate direction of failure)
  - Round-trip path lacks property (reverse traceroute)
  - Avoid problematic routers (bypass timestamp filters)
  - Differentiate forward/reverse properties (one-way delay)
- Need to encourage ISP/ testbed buy-in

Questions?

From me:

- Ideas on vantage points we can use?
- Ideas on clock syncing?
- Ideas on verifying one-way link latency?

For me?