Practical Reverse Traceroute

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

Affiliates, October 2008

This work partially supported by Cisco, Google, NSF
How do we communicate on Internet?
How do we communicate on Internet?

**Traceroute:** Tool to measure path from you to anywhere
What is traceroute used for?

*The most widely used Internet diagnostic tool*

- Network operators use traceroute to answer:
  - Is a destination reachable?
    - If yes, what is the route taken?
    - If no, where does it seem to be broken?
  - Is path longer than necessary?

- Researchers at UW use traceroute to:
  - Map the Internet
  - Predict performance and compare ISPs
  - Detect anomalies
Traceroute’s Fundamental Limitation

**Traceroute**: Tool to measure path FROM YOU to anywhere

**Our goal**: Reverse traceroute, w/o control of destination
IP Options to Identify Reverse Hops

- Set *IP options* on probe packets to instruct routers on path

- *Record Route (RR)* option
  - Record first 9 routers on path
  - If destination within 8, will record reverse
  - … but average path is 15 hops

- *Timestamp (TS)* option
  - Also used, but will not discuss during talk
Spoof to Best Use VPs and IP Options
Spoofing?? Isn’t that bad?

- We use only a restricted version
  - Only spoofing as nodes we control
    - Like a “reply to” address
  - Rate limit, restrict destinations (no broadcast IPs)
- Millions of spoofed probes sent to 10s of thousands of IPs, no complaints
- **Hubble** and this work show utility
- 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
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
To: S  
Fr: R1  
Ping!  
RR: $h_1, \ldots, h_6, R1, R2, R3$

- Iterate, performing TTL=8 pings and spoofed RR pings for each router we discover on return path
To: S  
Fr: R1  
Ping!  
RR: $h_1, \ldots, h_6, h_7, R3, R4$
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
How well does it work?

- Often able to determine complete reverse path
- When not, can often get minus last few hops
- Would improve further with more spoofing vantage points

Reverse paths from 200 random destinations across Internet back to 11 PlanetLab sites around the world

What is closest hop to destination for which we can calculate complete reverse path?

- Median: 1 hop from dst
- 40+: complete path from dst
**Debug inflation example:** Content provider detects poor performance to client
- 150ms round-trip latency from Florida to Seattle (2x expected)
- *(Current practice)* Issue traceroute, check if path is indirect:

<table>
<thead>
<tr>
<th>Hop no.</th>
<th>DNS name / IP address</th>
<th>Location</th>
<th>RTT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>132.170.3.1</td>
<td>Orlando, FL</td>
<td>0ms</td>
</tr>
<tr>
<td>2</td>
<td>198.32.155.89</td>
<td>–</td>
<td>0ms</td>
</tr>
<tr>
<td>3</td>
<td>jax-flrcore-7609-1-te23-v1820-1.net.flrnet.org</td>
<td>Jacksonville, FL</td>
<td>3ms</td>
</tr>
<tr>
<td>4</td>
<td>atlantaix.cox.com</td>
<td>Atlanta, GA</td>
<td>9ms</td>
</tr>
<tr>
<td>5</td>
<td>ashbbbrj02-ac0.0.r2.as.cox.net</td>
<td>Ashburn, VA</td>
<td>116ms</td>
</tr>
<tr>
<td>6</td>
<td>core2.te5-1-bbnet1.wdc002.pnap.net</td>
<td>Washington, DC</td>
<td>35ms</td>
</tr>
<tr>
<td>7</td>
<td>cr1.wdc005.inappnet-62.core2.wdc002.internap.net</td>
<td>Washington, DC</td>
<td>26ms</td>
</tr>
<tr>
<td>8</td>
<td>cr2-cr1.wdc005.internap.net</td>
<td>Washington, DC</td>
<td>24ms</td>
</tr>
<tr>
<td>9</td>
<td>cr1.mia004.inappnet.cr1.wdc005.internap.net</td>
<td>Miami, FL</td>
<td>53ms</td>
</tr>
<tr>
<td>10</td>
<td>cr1.sea002.inappnet.cr1.mia004.internap.net</td>
<td>Seattle, WA</td>
<td>149ms</td>
</tr>
</tbody>
</table>

But does not explain huge latency jump from 9 to 10

- *(With our tool)* Issue reverse traceroute, check reverse path:

<table>
<thead>
<tr>
<th>Hop no.</th>
<th>DNS name / IP address</th>
<th>Location</th>
<th>RTT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>cr1.sea002.inappnet.cr1.mia004.internap.net</td>
<td>Seattle, WA</td>
<td>148ms</td>
</tr>
<tr>
<td>2</td>
<td>cr1.sea002.inappnet.cr1.lax009.internap.net</td>
<td>Seattle, WA</td>
<td>141ms</td>
</tr>
<tr>
<td>3</td>
<td>internap-peer.lsanca01.transitrail.net</td>
<td>Los Angeles, CA</td>
<td>118ms</td>
</tr>
<tr>
<td>4</td>
<td>te4-1-4016.tr01-lsanca01.transitrail.net</td>
<td>Los Angeles, CA</td>
<td>118ms</td>
</tr>
<tr>
<td>5</td>
<td>te4-1-160.tr01-plalca01.transitrail.net</td>
<td>Palo Alto, CA</td>
<td>109ms</td>
</tr>
<tr>
<td>6</td>
<td>te4-1.tr01-sttlwa01.transitrail.net</td>
<td>Seattle, WA</td>
<td>92ms</td>
</tr>
<tr>
<td>7</td>
<td>te4-1.tr01-chcgil01.transitrail.net</td>
<td>Chicago, IL</td>
<td>41ms</td>
</tr>
<tr>
<td>8</td>
<td>te2-1-583.tr01-asbnva01.transitrail.net</td>
<td>Ashburn, VA</td>
<td>23ms</td>
</tr>
<tr>
<td>9</td>
<td>132.170.3.1</td>
<td>Orlando, FL</td>
<td>0ms</td>
</tr>
<tr>
<td>10</td>
<td>planetlab2.eecs.ucf.edu.</td>
<td>Orlando, FL</td>
<td>0ms</td>
</tr>
</tbody>
</table>

Indirectness: FL→DC→FL
Bad rev path causes inflated round-trip delay
Conclusion

- Traceroute is a very useful tool, but it cannot provide reverse path.

- Our reverse traceroute system fixes the limitation, providing complementary information.

- Complete reverse path in 40% or more of cases.

- Coverage is tied to the distribution of spoofing vantage points.
  - Have any we can use?