Scriptroute: A facility for distributed Internet measurement

www.scriptroute.org

Neil Spring, David Wetherall, and Tom Anderson

University of Washington

NANOG 26
Why Scriptroute?

Everyone measures the network

Network measurement is hard
  ● Development - arcane interfaces, need root.
  ● Infrastructure - distributed tools are harder.

Scriptroute is an environment for distributed lightweight active network measurements
  ● Development simplified - scripted measurements.
  ● 42 sites ready to execute new measurements.
Ideal Net Measurement Platform

1. Any network measurement
   - Connectivity, performance
   - Topology, routing

2. From any host
   - Vantage points
   - Comprehensive coverage for debugging

Can’t get there, but how close can we get?
Dedicated Testbeds

NIMI, RIPE NCC, AMP, etc.

1. Just about any measurement.

2. Dedicated machines
   - Uniform hardware and software image
   - Special hardware for timestamping

Great flexibility, inherently limited deployment.
Public Traceroute Servers

1. Just traceroute and ping.

2. Lots of servers, even routers via web gateways.

Open system:
- Provides service to any client.
- Anyone can run a server.

Wide deployment, inherently limited flexibility.
Scriptroute: Flexible measurement

1. Support measurements with:
   - Few packets, bytes
   - Safe packets
   - Little resources

2. On shared, ordinary servers
   - Interpreter is the environment

Combine the best attributes of both systems.
What we’ve used it for

**Small tools:**

<table>
<thead>
<tr>
<th>Tool</th>
<th>Description</th>
<th>Lines</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>traceroute, tcptraceroute</strong></td>
<td>Router paths</td>
<td>25</td>
</tr>
<tr>
<td><strong>ping</strong></td>
<td>1 second or Poisson intervals</td>
<td>27</td>
</tr>
<tr>
<td><strong>ally</strong></td>
<td>Group interface addresses to routers</td>
<td>90</td>
</tr>
<tr>
<td><strong>sprobe</strong></td>
<td>Packet-pair bandwidth estimation</td>
<td>63</td>
</tr>
</tbody>
</table>

**Distributed tools:**

<table>
<thead>
<tr>
<th>Tool</th>
<th>Description</th>
<th>Lines</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>GNP Evaluation</strong></td>
<td>Ping 8,000 hosts from 33 servers to test a host distance estimation scheme.</td>
<td>90</td>
</tr>
<tr>
<td><strong>Reverse Path Tree</strong></td>
<td>Build a tree of paths taken from all servers to an address.</td>
<td>340</td>
</tr>
</tbody>
</table>
Reverse path tree (whole)

Traceroute +: Alias resolution merges IP addresses to routers.

Network friendly: Doesn’t retrace hops, lookup names.
Reverse path tree (zooming)

I’m just showing how easy it is to capture a snapshot.
Now, focus in on the root.
Reverse path tree (zoomed)

sttl-dnva and sttl-snva are aliases listed together.
http://www.scriptroute.org/cgi-bin/rpt
Reverse path tree summary

Efficient:

- Name lookups at the client
- Tree not retraced; One probe per hop
- 195 traceroute packets instead of 975.
- 1.5 minutes tracing, 2.5 minutes alias resolution.

Foundation for future tools:

- Annotate with link properties, e.g., latency.
- Group by service provider, render on a world map.
- Find servers that share paths.
DNS: Lists active servers by AS, country
Front-End: Web server on port 3355
Interpreter: Resource limited, sandboxed execution
Core: Provides, restricts access to the network
Measurement scripts are expressed in Ruby:

- Object-oriented, type-safe, general-purpose
- Safe mode prevents system calls

Interpreter enhanced with:

- `send_train()` sends probes, collects responses
- Packet types, constants
- `Transparent hton[]`, `inet_aton()`.

Core deals with checksums and matching responses.
Traceroute for Scriptroute

#!/usr/bin/srinterpreter
probe = Scriptroute::Udp.new(12)
probe.ip_dst = ARGV[0]
port_unreach = false
catch (:port_unreachable) do
  (1..64).each { |ttl|
    (1..3).each { |
      rep|
      probe.ip_ttl = ttl
      pkts = Scriptroute::send_train([Struct::DelayedPacket.new(0,probe)])
      resp = (pkts[0].response) ? pkts[0].response.packet : nil
      rtt = (resp) ? ((pkts[0].response.time - pkts[0].probe.time) * 1000.0) : '*'
      if(resp.is_a?(Scriptroute::Icmp)) then
        puts ttl.to_s + ' ' + resp.ip_src + ' ' + rtt.to_s + ' ms'
        port_unreach = true if(resp.icmp_type == 3 & & resp.icmp_code == 3)
      end
    }
  }
  throw :port_unreachable if(port_unreach)
end
TCP-Traceroute: An easy mod

#!/usr/bin/srinterpreter

probe = Scriptroute::Tcp.new(0)
probe.ip_dst = ARGV[0]; tcp_rst = false

catch (:tcp_rst) do
  (1..64).each do |ttl|
    (1..3).each do |rep|
      probe.ip_ttl = ttl
      pkts = Scriptroute::send_train([Struct::DelayedPacket.new(0,probe)])
      resp = (pkts[0].response) ? pkts[0].response.packet : nil
      rtt = (resp) ? ((pkts[0].response.time - pkts[0].probe.time) * 1000.0) : ''
      if(resp.is_a?(Scriptroute::Icmp)) then
        puts ttl.to_s + ' ' + resp.ip_src + ' ' + rtt.to_s + ' ms'
      elsif(resp.is_a?(Scriptroute::Tcp)) then
        puts ttl.to_s + ' ' + resp.ip_src + ' ' + rtt.to_s + ' ms'
        tcp_rst = true
      end
    end
  end
throw :tcp_rst if(tcp_rst)
Security model: Open but restricted

Protecting Scriptroute hosts:
- Safe mode interpreter as nobody in a chroot
- Kernel resource limits
- Duration and number of scripts limited

Protecting the network and remote hosts
- Byte, packet, and SYN rate limits
- Filter bad traffic, e.g. broadcast, fragments
- Destination-specific filter repository
- Logging prevents anonymous reflection
Finding servers

73.as.servers.scriptroute.org

servers  listed by AS, continent, country, randomly.
policy  destination-specific filters to block traffic.
Conclusion

Scriptroute distributed lightweight measurement

Flexible: Remotely executed scripts.

Safe: Limit script behavior.

Open: Unauthenticated users, arbitrary servers.

Local control: Admins control limits, packets.

One system: Servers listed in DNS, easy distributed measurements.

Alpha deployment on PlanetLab (www.planet-lab.org).

For information, source, and packages:

www.scriptroute.org