A Practical Guide to (Correctly) Troubleshooting with Traceroute

Richard A Steenbergen <ras@nlayer.net> nLayer Communications, Inc.

Introduction

- Troubleshooting problems on the Internet?
  - The number one go-to tool is “traceroute”
    - Every OS comes with a traceroute tool of some kind.
    - There are thousands of websites which can run a traceroute.
    - There are dozens of “visual traceroute” tools available, both commercially and free.
  - And it seems like such a simple tool to use
    - I type in the target IP address and it shows me some routers.
    - And where the traceroute stops, or where the latency goes up a lot, that’s where the problem is, right?
    - How could this possibly go wrong?
  - Unfortunately, reality couldn’t be any further away.

Introduction

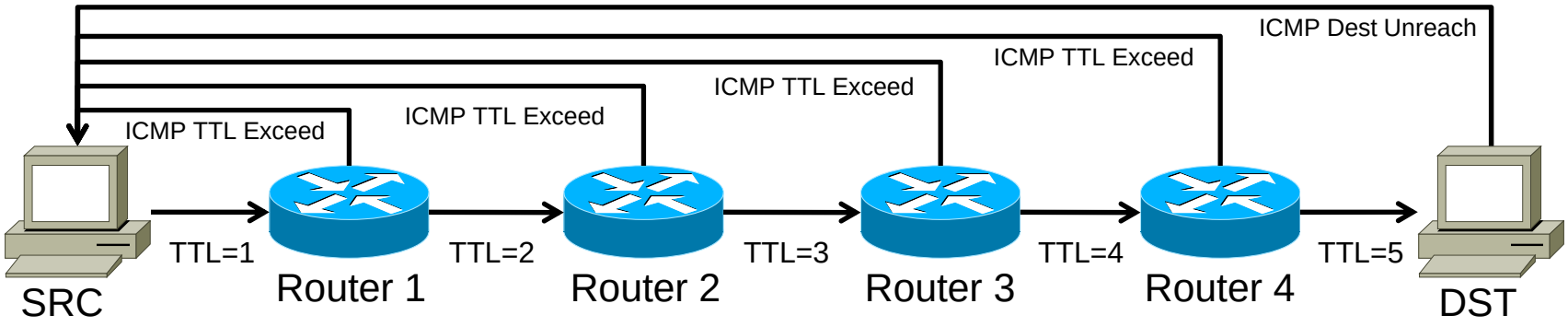
- So what's wrong with traceroute?
  - Most modern networks are actually well run
    - So simple issues like congestion or routing loops are becoming a smaller percentage of the total network issues encountered.
    - And more commonly, the encountered issues are complex enough that a naïve traceroute interpretation is utterly useless.
  - Few people are skilled at interpreting traceroute
    - Most ISP NOCs and even most mid-level engineering staff are not able to correctly interpret complex traceroutes.
    - This leads to a significant number of misdiagnosed issues and "false reports", which flood the NOCs of networks worldwide.
    - In many cases the problem of false reports is so bad, it is all but impossible for a knowledgeable outside party to submit a traceroute related ticket about a real issue.

Traceroute Topics

- Topics to discuss
  - How traceroute works
  - Interpreting DNS in traceroute
  - Understanding network latency
  - Asymmetric paths
  - Multiple paths
  - MPLS and traceroute
- Random Traceroute Factoid
  - The default starting port in UNIX traceroute is 33434. This comes from 32768 (2^15, or the max value of a signed 16-bit integer) + 666 (the mark of Satan).

Traceroute – The 10,000 Ft Overview

- 1. Launch a probe packet towards DST, with a TTL of 1
- 2. Each router hop decrements the TTL of the packet by 1
- 3. When TTL hits 0, router returns ICMP TTL Exceeded
- 4. SRC host receives this ICMP, displays a traceroute “hop”
- 5. Repeat from step 1, with TTL incremented by 1, until…
- 6. DST host receives probe, returns ICMP Dest Unreach.
- 7. Traceroute is completed.



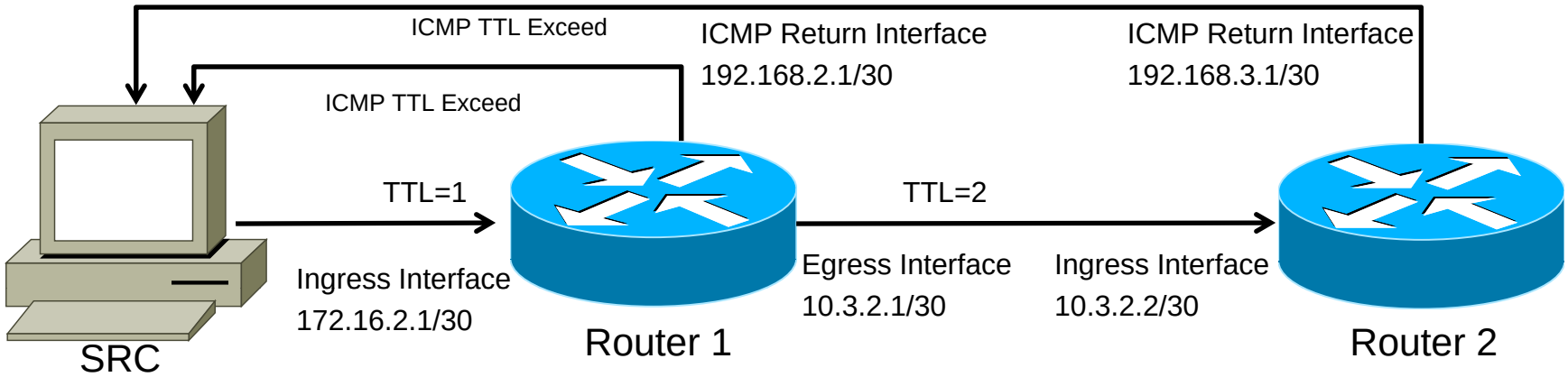
Traceroute – A Little More Detail

- Multiple Probes
  - Most traceroute implementations send multiple probes.
  - The default is 3 probes per TTL increment (“hop”).
    - Hence the normal 3 latency results, or 3 *’s if no response.
  - Each probe uses a different DST Port to distinguish itself
    - So any layer 4 hashing can send each probe on different paths.
    - This may be visible to traceroute in the case of ECMP hashing.
    - Or invisible, in the case of 802.3ad style Layer 2 aggregation.
    - But the result is the same, some probes may behave differently.
- Not all traceroute implementations use UDP
  - Windows uses ICMP, other tools may even use TCP.

Traceroute – Latency Calculation

- How is traceroute latency calculated?
  - Timestamp when the probe packet is launched.
  - Timestamp when the reply ICMP is received.
  - Subtract the difference to determine round-trip value.
  - Routers along the path do not do any time “processing”
    - They simply reflect the original packet’s data back to the SRC.
    - Many implementations encode the original launch timestamp into the probe packet, to increase accuracy and reduce state.
  - But remember, only the ROUND TRIP is measured.
    - Traceroute is showing you the hops on the forward path.
    - But showing you latency based on the forward PLUS reverse paths. Any delays on the reverse path will affect your results!

Traceroute – What Hops Are You Seeing?



- Packet with TTL 1 enters router via ingress interface
- ICMP TTL Exceed is generated as the TTL hits 0
  - ICMP source address is that of the ingress router interface.
  - This is how traceroute sees the address of a “hop”, the ingress IP.
  - The above traceroute will read: 172.16.2.1 10.3.2.2
- Random factoid: This behavior is actually non-standard
  - RFC1812 says the ICMP source MUST be from the egress iface.
  - If obeyed, this would prevent traceroute from working properly.

How to Interpret DNS in a Traceroute

Interpreting DNS in a Traceroute

- Interpreting DNS is one of the most useful important aspects of correctly using traceroute.
- Information you can discover includes:
  - Location Identifiers
  - Interface Types and Capacities
  - Router Type and Roles
  - Network Boundaries and Relationships

Interpreting Traceroute - Location

- Knowing the geographical location of the routers is an important first step to understanding an issue.
  - To identify incorrect/suboptimal routing.
  - To help you understand network interconnections.
  - And even to know when there isn't a problem at all, i.e. knowing when high latency is justified and when it isn't.
- The most commonly used location identifiers are:
  - IATA Airport Codes
  - CLLI Codes
  - Attempts to abbreviate based on the city name.
  - But sometimes you just have to take a guess.

Location Identifiers – IATA Airport Codes

- IATA Airport Codes
  - Good International coverage of most large cities.
  - Most common in networks with a few big POPs.
  - Examples:
    - Santo Domingo = SDQ
    - San Jose California = SJC
  - Sometimes represented by pseudo-airport codes
    - Especially where multiple airports serve a region
    - Or where the airport code is non-intuitive
      - New York, NY is served by JFK, LGA, and EWR airports.
        - But is frequently written as NYC.
      - Northern VA is served by IAD, Washington DC by DCA.
        - But both may be written as WDC.

Location Identifiers – CLLI Codes

- Common Language Location Identifier
  - Full codes maintained (and sold) by Telecordia.
  - Most commonly used by Telephone Companies
    - Example: **HSTNTXMOCG0**
  - In a non-Telco role, may only use the city/state identifiers
    - Examples:
      - HSTNTX = Houston Texas
      - ASBNVA = Ashburn Virginia
  - Well defined standard covering almost all US/CA cities
    - Commonly seen in networks with a larger number of POPs.
    - Not an actual standard outside of North America
      - Some providers fudge these, e.g. AMSTNL = Amsterdam NL

Location Identifiers – Arbitrary Values

- And then sometimes people just make stuff up
  - Chicago IL
    - Airport Code: ORD (O'Hare) or MDW (Midway)
    - CLLI Code: CHCGIL
    - Example Arbitrary Code: CHI
  - Toronto ON
    - Airport Code: YYZ or YTC
    - CLLI Code: TOROON
    - Example Arbitrary Code: TOR
  - Frequently based on the good intentions of making thing readable in plain English, even though these may not follow any standards.

Common Locations – US Major Cities

Location Name	Airport Codes	CLLI Code	Other Codes
Ashburn VA	IAD	ASBNVA	WDC, DCA
Atlanta GA	ATL	ATLNGA	
Chicago IL	ORD, MDW	CHCGIL	CHI
Dallas TX	DFW	DLLSTX	DAL
Houston TX	IAH	HSTNTX	HOU
Los Angeles CA	LAX	LSANCA	LA
Miami FL	MIA	MIAMFL	
Newark NJ	EWR	NWRKNJ	NEW, NWK
New York NY	JFK, LGA	NYCMNY	NYC, NYM
San Jose CA	SJC	SNJSCA	SJO, SV, SF
Palo Alto CA	PAO	PLALCA	PAIX, PA
Seattle CA	SEA	STTLWA	

Common Locations – Non-US Major Cities

Location Name	Airport Codes	CLLI Code (*)	Other Codes
Amsterdam NL	AMS	AMSTNL	
Frankfurt GE	FRA	FRNKGE	
Hong Kong HK	HKG	NEWTHK	
London UK	LHR	LONDEN	LON
Madrid SP	MAD	MDRDSP	
Montreal CA	YUL	MTRLPQ	MTL
Paris FR	CDG	PARSFR	PAR
Singapore SG	SIN	SNGPSI	
Seoul KR	GMP, ICN	SEOLKO	SEL
Sydney AU	SYD	SYDNAU	
Tokyo JP	NRT	TOKYJP	TYO
Toronto CA	YYZ, YTC	TOROON	TOR

Interpreting DNS – Interface Types

- Most networks will try to put interface info in DNS
  - Often to help them troubleshoot their own networks.
  - Though this many not always be up to date.
    - Many large networks use automatically generated DNS.
  - Can potentially help you identify the type of interface
    - As well as capacity, and maybe even the make/model of router.
- Examples:
  - xe-11-1-0.edge1.NewYork1.Level3.net
    - XE-#/#/# is Juniper 10GE port. The device has at least 12 slots.
    - It’s at least a 40G/slot router since it has a 10GE PIC in slot 1.
    - It must be Juniper MX960, no other device could fit this profile.

Common Interface Naming Conventions

Interface Type	Cisco IOS	Cisco IOS XR	Juniper
Fast Ethernet	Fa#/#		fe-#/#/#
Gigabit Ethernet	Gi#/#	Gi#/#/#/#	ge-#/#/#
10 Gigabit Ethernet	Te#/#	Te#/#/#/#	xe-#/#/#
SONET	Pos#/#	POS#/#/#/#	so-#/#/#
T1	Se#/#		t1-#/#/#
T3			t3-#/#/#
Ethernet Bundle	Po# / Port-channel#	BE####	ae#
SONET Bundle	PosCh#	BS####	as#
Tunnel	Tu#	TT# or TI#	ip-#/#/# or gr-#/#/#
ATM	ATM#/#	AT#/#/#/#	at-#/#/#
Vlan	Vl###	Gi#/#/#/#.###	ge-#-#-#.###

Interpreting DNS – Router Types/Roles

- Knowing the role of a router can be useful
  - But every network is different, and uses different naming conventions.
  - And just to be extra confusion, they don't always follow their own naming rules.
- Generally speaking, you can guess the context and get a basic understanding of the roles.
  - Core routers – CR, Core, GBR, BB, CCR, EBR
  - Peering routers – BR, Border, Edge, IR, IGR, Peer
  - Customer routers – AR, Aggr, Cust, CAR, HSA, GW

Network Boundaries and Relationships

- Identifying Network Boundaries is Important
  - These tend to be where routing policy changes occur.
    - For example, different return paths based on Local Preference.
  - These also tend to be areas where capacity and routing are the most difficult, thus likely to be problems.
- Identifying the relationship can be helpful too
  - Typically: a) Transit Provider, b) Peer, or c) Customer.
  - Many networks will try to indicate demarcs in their DNS
    - Examples:
      - Clear names like network.customer.alter.net
      - Or always landing customers on routers named “gw”

Network Boundaries and Relationships

- It's easy to spot where the DNS changes
  - 4 te1-2-10g.ar3.DCA3.gblx.net (67.17.108.146)
  - 5 sl-st21-ash-8-0-0.sprintlink.net (144.232.18.65)
- Or, look for "remote party" name in the DNS
  - 4 po2-20G.ar5.DCA3.gblx.net (67.16.133.90)
  - 5 cogent-1.ar5.DCA3.gblx.net (64.212.107.90)
  - Common where one side controls the /30 DNS, and the other side doesn't provide interface information.
- For more info, look at the other side of the /30
  - > nslookup 64.212.107.89
  - Result: te2-3-10GE.ar5.DCA3.gblx.net

Understanding Network Latency

Understanding Network Latency

- Three primary types of network induced latency
  - Serialization Delay
    - The delay caused by having to transmit data through routers/switches in packet sized chunks.
  - Queuing Delay
    - The time spent in a router's queues waiting for transmission. This is mostly related to line contention (full interfaces), since without congestion there is very little need for a measurable queue.
  - Propagation Delay
    - The time spent "in flight", in which the signal is traveling over the transmission medium. This is primarily a limitation based on the speed of light, or other electromagnetic propagation.

Latency – Serialization Delay

- Delay caused by packet-based forwarding
  - Packets move through the network as a single unit.
  - Can’t transmit the next packet until last one is finished.
- Not much as an issue in modern networks
  - Speeds have increased by orders of magnitude over the years, while packet sizes have stayed the same (small).
  - 1500 bytes over a 56k link (56Kbps) = 214.2ms delay
  - 1500 bytes over a T1 (1.536Mbps) = 7.8ms delay
  - 1500 bytes over a FastE (100Mbps) = 0.12ms delay
  - 1500 bytes over a GigE (1Gbps) = 0.012ms delay

Latency – Queuing Delay

- First you must understand “Utilization”
  - A 1GE doing 500Mbps is said to be “50% utilized”
  - But in reality, an interface is either transmitting (100% utilized) or not transmitting (0% utilized) at any instant
  - The above is really “used 50% of the time, over 1 second”
- Queueing is a natural function of routers
  - When a packet is ready to send but the interface is in use, it must be queued until the interface is free.
  - As an interface reaches saturation, the probability of a packet being queued rises exponentially.
  - When an interface is extremely full, a packet may be queued for many hundreds or thousands of miliseconds.

Latency – Propagation Delay

- Delay caused by signal propagation over distance.
  - Light travels through a vacuum at ~300,000km/sec
  - Fiber cores have a refractive index of ~1.48
  - 1/1.48 = ~0.67c, light through fiber = ~200,000km/sec
  - 200,000km/sec = 200km (or 125 miles) per millisecond.
  - Divide by 2 for round-trip time (RTT) measurements.
- Example:
  - A round-trip around the world at the equator, via a perfectly straight fiber route, would take ~400ms due solely to speed-of-light propagation delays.

Identifying the Latency Affecting You

- So, how do you determine if latency is normal?
  - Use location identifiers to determine geographical data.
  - See if the latency fits with propagation delay.
  - For example:

    3 xe-3-0-0.cr1.nyc3.us.nlayer.net (69.22.142.74) 6.570ms

    4 xe-0-0-0.cr1.lhr1.uk.nlayer.net (69.22.142.10) 74.144ms

    New York NY to London UK in 67.6ms? 4200 miles? Yup!
  - Another example:

    5 cr2.wswdc.ip.att.net (12.122.3.38) [MPLS: Label 17221 Exp 0] 8 msec 8 msec 8 msec

    6 tbr2.wswdc.ip.att.net (12.122.16.102) [MPLS: Label 32760 Exp 0] 8 msec 8 msec 8 msec

    7 ggr3.wswdc.ip.att.net (12.122.80.69) 8 msec 8 msec 8 msec

    8 192.205.34.106 [AS 7018] 228 msec 228 msec 228 msec

    9 te1-4.mpd01.iad01.atlas.cogentco.com (154.54.3.222) [AS 174] 228 msec 228 msec 228 msec

    Washington DC to Washington DC in 220ms? Nope!

Prioritization and Rate Limiting

“To It” vs. “Through It”

- Architecture of a modern router:
  - Packets forwarded through the router (data plane)
    - Fast Path: hardware based forwarding of ordinary packets
      - Example: Almost every packet in normal Internet traffic.
    - Slow Path: software based handling of “exception” packets
      - Example: IP Options, ICMP Generation (including TTL Exceeded)
  - Packets being forwarded TO the router (control plane)
    - Example: BGP, IGP, SNMP, CLI access (telnet/ssh), ping, or any packets sent directly to a local IP address on the router.
  - These CPUs tend to be relatively underpowered
    - A 320-640+ Gbps router may only have a 600MHz CPU
  - ICMP Generation is *NOT* a priority for the router.

The Infamous BGP Scanner

- On many platforms the slow-path data plane and the control-plane share the same resources.
  - And often don’t have the best schedulers for the CPU
  - As a result, control-plane activity such as BGP churn, CLI use, and periodic software processes can consume CPU and slow the generation of ICMP TTL Exceeds.
  - This results in random “spikes” in traceroute latency, which is often misinterpreted as a network issue.
- The most infamous process which causes these spikes is called “BGP Scanner”, and runs every 60 seconds on all Cisco IOS devices.

Rate Limited ICMP Generation

- Most routers also rate limit their ICMP generation
  - Often with arbitrary hard-coded limits.
  - Which may be insufficient under heavy traceroute load.
- Juniper
  - Hard limit of 50pps per interface, 250pps on FPC3s
  - Hard limit of 500pps per PFE as of JUNOS 8.3+
- Foundry
  - Hard limit of 400pps per interface
- Force10
  - Hard limit of 200pps or 600pps per interface

Spotting The Fake Latency

- The most important rule of all
  - If there is an actual issue, the latency will continue or increase for all future hops:
    - Example (Not a real issue in hop 2):

      1 ae3.cr2.iad1.us.nlayer.net 0.275 ms 0.264 ms 0.137 ms

      2 xe-1-2-0.cr1.ord1.us.nlayer.net 18.271 ms 18.257 ms 68.001 ms

      3 tge2-1.ar1.slc1.us.nlayer.net 53.373 ms 53.213 ms 53.227
  - Latency spikes in the middle of a traceroute mean absolutely nothing if they do not continue forward.
    - At worst it could be the result of an asymmetric path.
    - But it is probably an artificial rate-limit or prioritization issue.
    - By definition, if regularly forwarded packets are being affected you should see the issue persist on all future hops.

Asymmetric Paths

Asymmetric Paths

- The number one plague of traceroute
- Traceroute shows you the forward path only
  - But the latency shown for each hop is based on
    - The time it took for the probe packet to reach the hop, PLUS
    - The time it took for the TTL Exceed reply to come back.
- The reverse path itself is completely invisible
    - Not only does traceroute not reveal anything about it, but…
    - It can be completely different at every hop in the forward path.
    - The only solution is to look at both forward and reverse traceroutes
    - And even then, it can't catch potential asymmetric paths in the middle.

Asymmetric Paths and Network Boundaries

- Asymmetric paths often start at network boundaries
  - Why? Because that is where admin policies change.

    te1-1.ar2.DCA3.gblx.net (69.31.31.209) 0.719 ms 0.560 ms 0.428 ms
    te1-2-10g.ar3.DCA3.gblx.net (67.17.108.146) 0.574 ms 0.557 ms 0.576 ms
    sl-st21-ash-8-0-0.sprintlink.net (144.232.18.65) 100.280 ms 100.265 ms 100.282 ms
    144.232.20.149 (144.232.20.149) 102.037 ms 101.876 ms 101.892 ms
    sl-bb20-dc-15-0-0.sprintlink.net (144.232.15.0) 101.888 ms 101.876 ms 101.890 ms

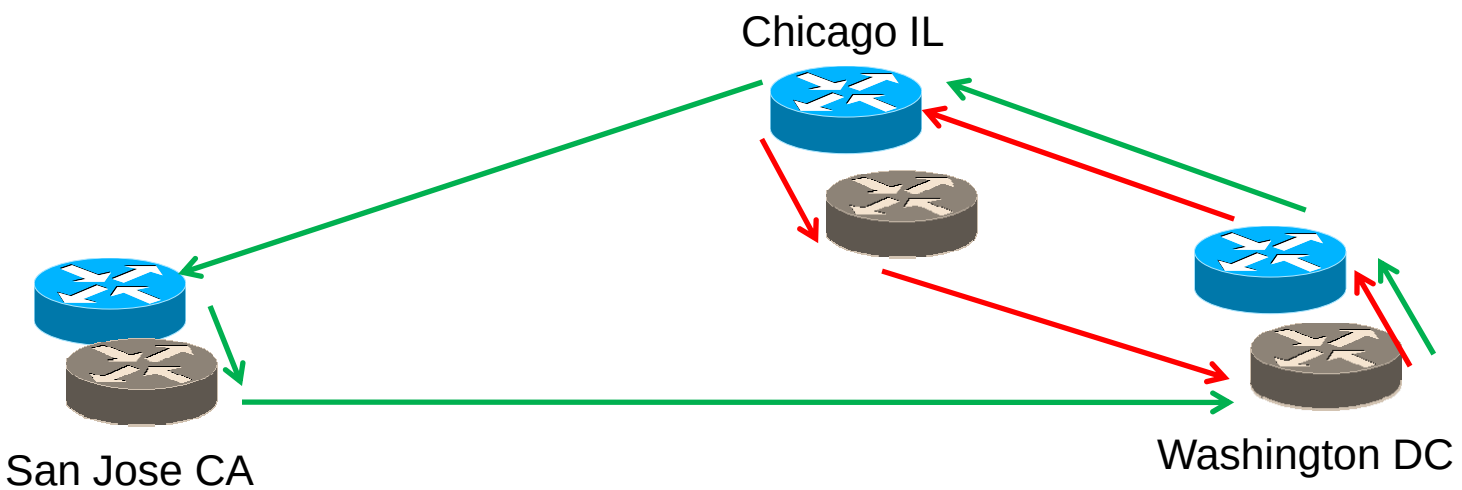
  - What's wrong in the path above?
    - It COULD be congestion between GBLX and Sprint.
    - But it could also be an asymmetric reverse path.
    - At this GBLX/Sprint boundary, the reverse path policy changes.
    - This is often seen in multi-homed network with multiple paths.
    - In the example above, Sprint's reverse route goes via a circuit that is congested, but that circuit is NOT shown in the traceroute.

Using Source Address in your Traceroute

- How can you work around asymmetric paths?
  - The most powerful option is to control your SRC address.
  - In the previous example, assume that:
    - You are multi-homed to Global Crossing and Level3
    - Global Crossing reaches you via Global Crossing
    - Sprint reaches you via Level3
    - There is a problem between Sprint and Level3.
  - How can you prove the issue isn't between GX and Sprint?
    - Run a traceroute using your side of the GBLX /30 as your source.
    - This /30 comes from your provider (GBLX)'s larger aggregate.
    - The reverse path will be guaranteed to go Sprint->GBLX
    - If the latency doesn't persist, you know the issue is on the reverse.

Asymmetric Paths

- But remember, asymmetric paths can happen anywhere
- Especially where networks connect in multiple locations
  - And use closest-exit (hot potato) routing, as is typically done.
  - Hop 1 (red) returns via a Chicago interconnection
  - Hop 2 (green) returns via a San Jose interconnection



Using Source Address in your Traceroute

- But what if the /30 is numbered out of my space?
  - As in the case of a customer or potentially a peer.
- You can still see some benefits from setting SRCs
  - Consider trying to examine the reverse path of a peer who you have multiple interconnection points with.
    - A traceroute sourced from your IP space (such as a loopback) may come back via any of multiple interconnection points.
    - But if the remote network carries the /30s of your interconnection in their IGP (i.e. they redistribute connected into their IGP)…
    - Then the traffic will come back over their backbone, and return to you via the /30 you are testing from.
    - Trying both options can give you different viewpoints.

Default Source Addresses

- When tracerouting from a router…
  - Most routers default to using the source address of the egress interface that the probe leaves from.
  - This may or may not be what you want to see.
  - Some platforms can be configured to default to a loopback address rather than the egress interface.
    - For example, Juniper.

Multiple Paths and Load Balancing

Multiple Paths

- Because each (UDP/TCP) traceroute probe uses a different layer 4 port, equal-cost multi-path may make multiple paths show up within a single “hop”
  - This is relatively harmless, but may be confusing.
  - Example:

```
6  ldn-bb2-link.telia.net (80.91.251.14)  74.139 ms  74.126 ms
   ldn-bb1-link.telia.net (80.91.249.77)  74.144 ms
7  hbg-bb1-link.telia.net (80.91.249.11)  89.773 ms
   hbg-bb2-link.telia.net (80.91.250.150)  88.459 ms  88.456 ms
8  s-bb2-link.telia.net (80.91.249.13)  105.002 ms
   s-bb2-link.telia.net (80.239.147.169)  102.647 ms  102.501 ms
```

  - Of the 3 probes, 2 go over one path, 1 goes over another.

Multiple Paths - Examples

- A slightly more complex example

```
4  p16-1-0-0.r21.asbnva01.us.bb.verio.net (129.250.5.21)  0.571 ms  0.604 ms  0.594 ms
5  p16-1-2-2.r21.nycmny01.us.bb.verio.net (129.250.4.26)  7.279 ms  7.260 ms
   p16-4-0-0.r00.chcgil06.us.bb.verio.net (129.250.5.102)  25.981 ms
6  p16-2-0-0.r21.sttlwa01.us.bb.verio.net (129.250.2.180)  71.027 ms
   p16-1-1-3.r20.sttlwa01.us.bb.verio.net (129.250.2.6)  66.730 ms  66.535 ms
```

- ECMP between two parallel but different paths
  - Ashburn VA – New York NY – Seattle WA
  - Ashburn VA – Chicago IL – Seattle WA
- Also harmless, but potentially confusing.

Multiple Unequal Length Paths

- A much worse scenario is ECMP where the load-balanced paths are of unequal hop length.
  - This can make the traceroute appear to go back and forth, and is extremely confusing and difficult to read.
  - Traceroute hops end up looking like this
    - 1 A A A
    - 2 B X B
    - 3 C B C
    - 4 D C D
    - 5 E D E

Handling Multiple Paths

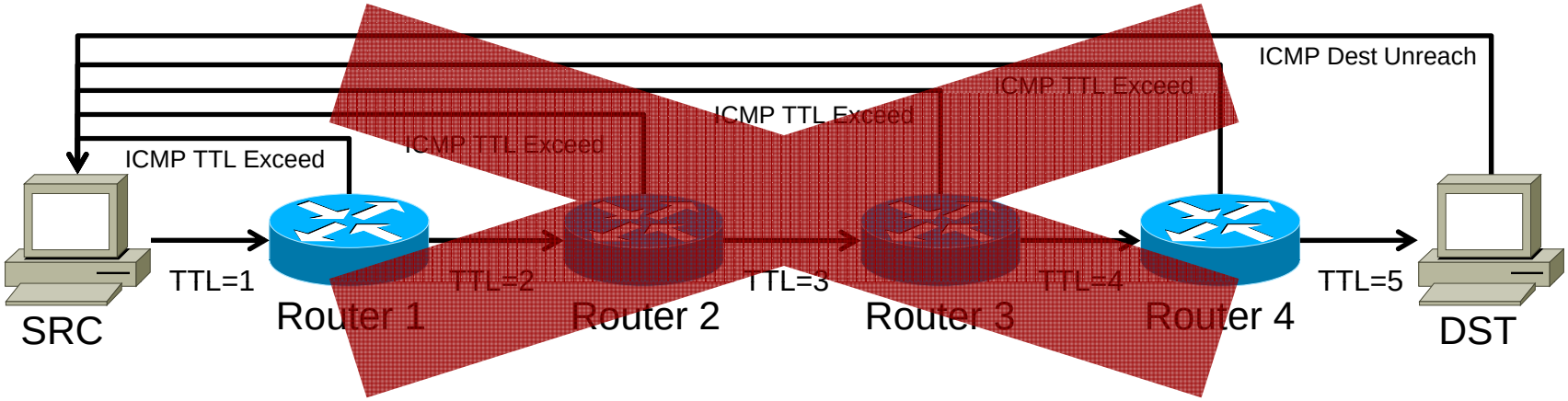
- When in doubt, only look at a single path
  - Set your traceroute client to only send a single probe.
  - But be aware that this may not be the path which your actual traffic forwards over.
    - One way to try out different paths which may be available is to increment the dest IP by 1 or try different source IPs.

MPLS and Traceroute

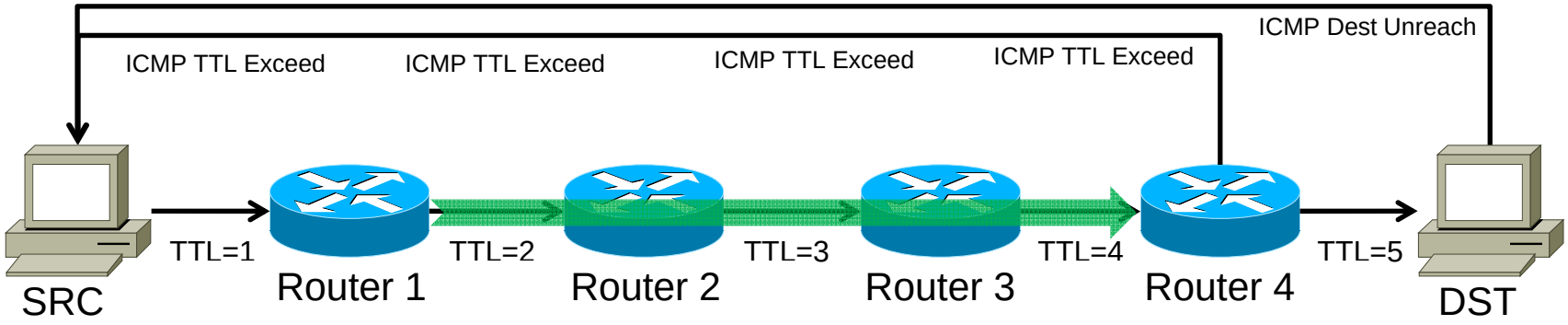
MPLS ICMP Tunneling

- Many large networks operate an MPLS based core
- Some devices don’t even carry an IP routing table
  - This is fine for switching MPLS labeled packets
  - But presents a problem when ICMPs are generated
  - How does the MPLS-only router deliver an ICMP?
- One solution is called ICMP Tunneling
  - If generating an ICMP about a packet inside an LSP
  - Then put the generated ICMP back into the same LSP
  - Works for delivering the message, but…
  - It can make traceroutes look really WEIRD!

MPLS ICMP Tunneling Diagram



All returned ICMP packets must travel to the end of the LSP before going back to the sender. This makes every hop in the LSP appear to have the same RTT as the final hop.



MPLS ICMP Tunneling Example

1. te2-4.ar5.PAO2.gblx.net (69.22.153.209) 1.160 ms 1.060 ms 1.029 ms
2. 192.205.34.245 (192.205.34.245) 3.984 ms 3.810 ms 3.786 ms
3. tbr1.sffca.ip.att.net (12.123.12.25) 74.848 ms 74.859 ms 74.936 ms
4. cr1.sffca.ip.att.net (12.122.19.1) 74.344 ms 74.612 ms 74.072 ms
5. cr1.cgcil.ip.att.net (12.122.4.122) 74.827 ms 75.061 ms 74.640 ms
6. cr2.cgcil.ip.att.net (12.122.2.54) 75.279 ms 74.839 ms 75.238 ms
7. cr1.n54ny.ip.att.net (12.122.1.1) 74.667 ms 74.501 ms 77.266 ms
8. gbr7.n54ny.ip.att.net (12.122.4.133) 74.443 ms 74.357 ms 75.397 ms
9. ar3.n54ny.ip.att.net (12.123.0.77) 74.648 ms 74.369 ms 74.415 ms
10. 12.126.0.29 (12.126.0.29) 76.104 ms 76.283 ms 76.174 ms
11. route-server.cbbtier3.att.net (12.0.1.28) 74.360 ms 74.303 ms 74.272 ms

Send questions, complaints, to:

Richard A Steenbergen <ras@nlayer.net>