Pings, Traces and other network troubleshooting commands (and how to read them)

Bryce L. Meyer

12 Sep 2018

St. Louis UNIX Users Group

Overview

- Ping
  - Why ping is not a panacea
- Trace(route)
  - How does it work
  - Reading traceroutes across the internet
  - CLLI and Airport Codes
  - Using Whois
- Route Servers and Looking Glass Servers
- Troubleshooting

Pings, Pings

- 'Ping' sends a set of test packets of specified size to an IP address
  - Default ICMP, can be turned off to many devices so be warned.
  - Note Flood Pings can DOS an interface, so ICMP ping packets are typically flushed..
  - Routers allow other protocols…Change to UDP, TCP, HTTP (port 80) or other ports and try there...sometimes that works (aka a poor geek's Port Scan...)
  - Often used between servers to measure performance on a network and for debugging
    - Care to watch by packet size and timing is important!
      - Some pipes might let small packets far apart in, but are broken enough to block larger packets or a tighter sweep may show drops.
  - Ping as close as possible to target (you saw this in Class #7)

UNIX Syntax (z/OS): PING

- ping *hostname –options*
  - hostname can be an ip address or url
- options:
  - *-h or -?* Help (man)
  - ***-A ipv4 or ipv6***
    - Use ipv4 or ipv6
  - ***-c echo***
    - If not specified, 1 is default, except if verbose, then 3
    - If 0, ping sends pings until ^c
    - Can be 0 to 2,147,483,647.
  - ***-l size***
    - In bytes, if not specified by using l, 256 is default, values can be from 0 to 65,487. Larger pings are more likely to be dropped.
  - *For z/OS: -i interface*
    - Specifies which local interface to use. If –A must match version(?).
  - For SUSE: *-i interval*
    - Time between packets in seconds (default it 1 second)
  - For Linux/SUSE, admin: *-f*
    - FLOOD PING: send as many as possible to test interfaces. Aka DOS attack?
  - *-n*
    - Do not resolve option. Must also set –P too for this to work.
  - *-P yes or ignore.*
    - yes: Specifies that the outbound echo request packets are not fragmented at the local host or in the network and that the MTU value, determined by path MTU discovery for the destination, are use
    - ignore: Specifies that the outbound echo request packets are not fragmented at the local host or in the network, and that any MTU values determined by path MTU discovery for the destination, are ignored.
  - *-p tcpname*
    - Specifies the name of the TCP/IP stack to be used.
  - *-s srcip*
    - *Source Ip address to be used*
  - ***-t seconds***
    - How long to wait for a response in seconds, 1-100 seconds
  - ***-v***
    - ***Verbose mode***

https://www.ibm.com/support/knowledgecenter/en/SSLTBW_2.3.0/com.ibm.zos.v2r3.halu101/zoping.htm

Verbose ping

When the **-v** parameter is specified, the following information is displayed:

`Echo reply details`

`Ping #`*`n`*` from `*`address`*

Processing echo reply counter and IP address of the echo reply sender.

`bytes=`*`nn`*

Number of bytes for the ICMP packet (ICMP header and data portions) from the echo reply.

`seq=`*`nn`*

ICMP sequence number of the echo reply.

`ttl=`*`nn`*` (for IPv4)`

Time-to-live value for the echo reply.

`hoplim=`*`nn`*` (for IPv6)`

Hop limit value for the echo reply.

`time=`*`nn`*` ms`

Round-trip time (RTT), in milliseconds.

`Ping statistics summary`

`Sent`

Total number of echo request packets sent.

`Received`

Total number of echo reply packets received.

`Lost (`*`n`*`% loss)`

Total number of lost echo packets (echo reply packets not received) and the percentage loss.

`Approximate round trip times (RTT) in milliseconds`

`Minimum`

Minimum RTT value of Ping requests that were sent.

`Maximum`

Maximum RTT value of Ping requests that were sent.

`Average`

Average RTT value of Ping requests that were sent.

`StdDev`

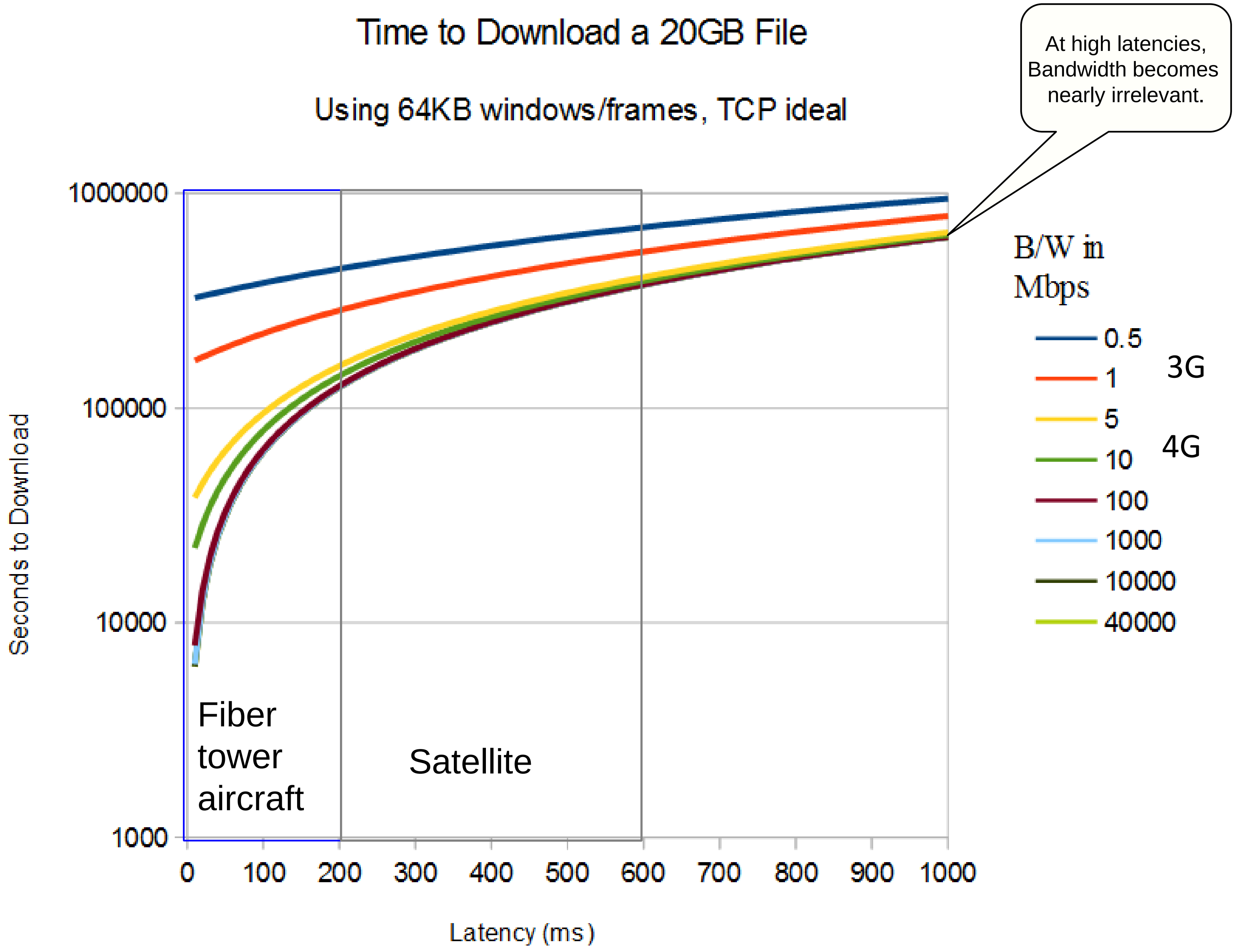
Standard deviation of all RTT values of Ping requests that were sent.

https://www.ibm.com/support/knowledgecenter/en/SSLTBW_2.3.0/com.ibm.zos.v2r3.halu101/zoping.htm

Graphing Jitter, Packet Loss, and Latency

- Round Trip Latency: Time a packet takes to get to and from a certain location
  - Can be roughly measured with a ping
- Packet Loss: % of packets lost of test packets (or actual packets) sent
- Jitter: Variation in Packet Loss (Std Dev. Or similar)
- Availability: Time Pcket Loss is below a set threshold.

Round Trip Latency over Distance: The Pain of Latency on TCP

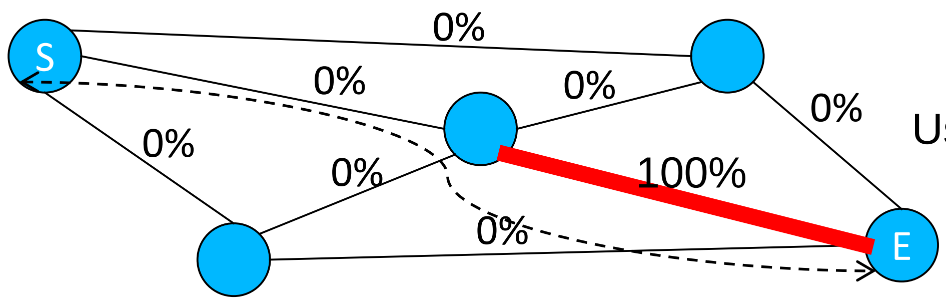


Packet Loss

- Many, Many Things cause lost packets/bits/bytes.
  - Over-utilization of your Pipe
  - Failing Cards
  - CPU issues
  - EM Interference (over the air, via satellite)
  - Time-Outs due to long latency
  - Bad Optics + Pinched or Poorly connected fiber
  - Corroded Copper
  - etc. etc. etc.
- Packet loss graph shape is important as the number itself for troubleshooting
- Comparison to traps, Utilization, and Error rate are very important!
- Packet Loss can also slow data transfers (due to repeat packet transfers)
- All latency issues + Weather/EM-Interference for Broadband Wireless

Intermittent Packet Loss

- Trickier than a straight failure
- IF there is no graphing, you will need very long pings (i.e. lots of ping signals) to catch
- Look for the link with highest percentage of loss, it is likely the issue, especially for variable routing
- Can be related to Size Setting Mismatches (see Packet loss with Packet Size)

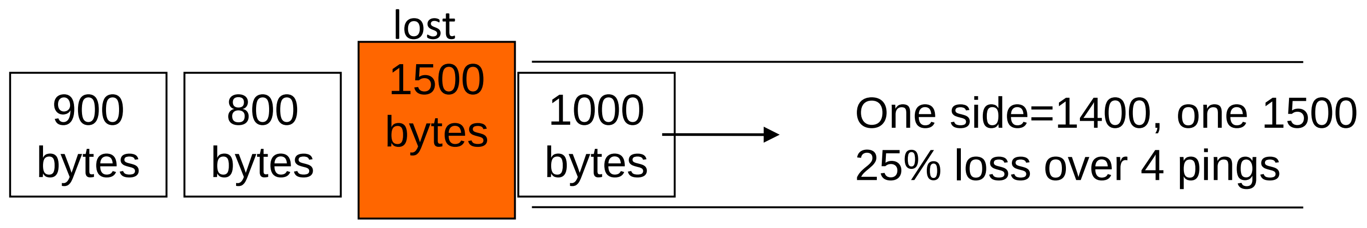


Use Logs and many pings!

S to E: 0% to 25%

Packet Loss with Packet Size

- Common with MTU/Cell Size/Ethernet Duplex issues
  - Mismatches cause packet failure for packets bigger than smallest setting
  - Since traffic is of multiple sizes, only biggest packets get chopped
- Also Common with memory issues or over utilization
  - Packets lost marked with a ‘C’ for congestion in pings sometimes.

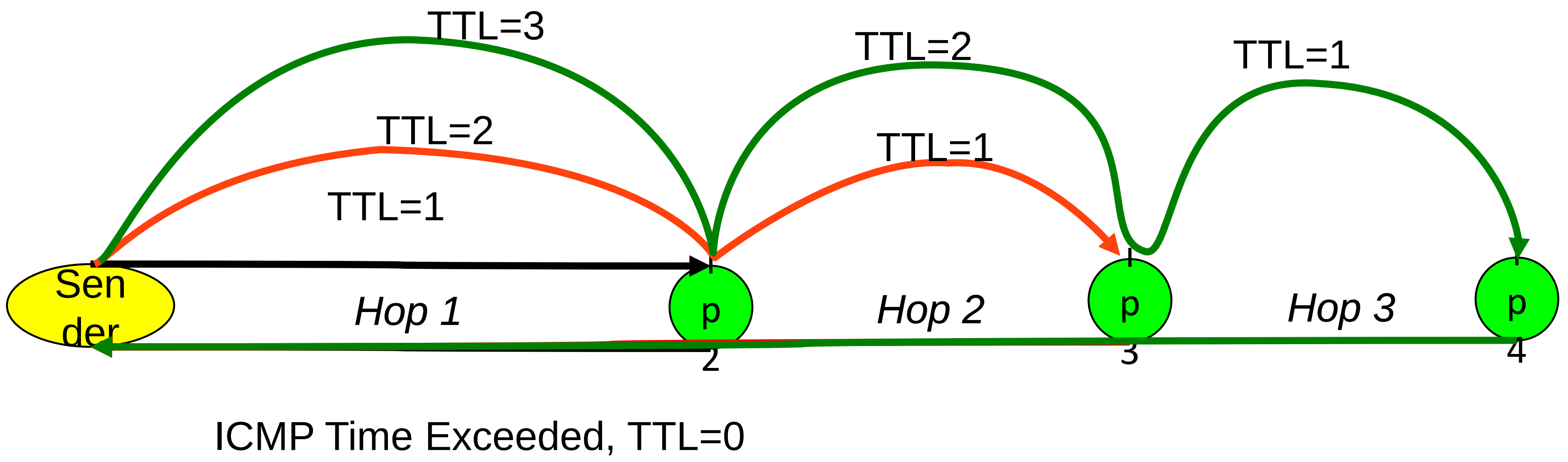


My Favorite Troubleshooting Tips for Packet Loss

- Run a long ping, and sweep sizes, to duplicate issue of the same path
- Check for mtu and duplex mismatches
- Check Utilization and Error Rate on interfaces between issue sites
    - Any over use is a dead giveaway
    - Ditto for high error rates
- Check processing load on boxes (i.e. look for traps that show overruns or card errors)..should be alarms fore these…or sh mem and sh proc
- Then Check for anything else

How does Traceroute work?

- (fyi, tracert in windows, trace or traceroute for short in UNIX/Apple)
- Default is ICMP in Win, UDP in UNIX/LINUX/etc.
  - You can specify protocol in command, and other characteristics
- Sender sends packets of time to live (TTL, aka 'hops') of increasing size
- Time between send and recieving ICMP Time Exceeded = latency



Warning: Trace route is an IMPLIED route

https://en.wikipedia.org/wiki/Time_to_live https://en.wikipedia.org/wiki/Traceroute

Trace(route) syntax (SUSE)

- **`traceroute –`*`options`*` url_or_ip address`**
  - (note in windows: tracert, and some routers: trace)
- **`-h or -?`**
  - man page
- **`-4 or -6`**
  - set either ipv4 or ipv6
- **`-F`**
  - do not fragment. IF a packet is too big (over MTU) then ignore.
- **`-f`** ***`first_ttl`***
  - TTL for first packet (default is 1)
- **`-g`** ***`gateway`***
  - Set source routing (i.e. from gateway), though usually doesn't work
- **`-I`**
  - Use ICMP in lieu of UDP packets
- **`-i`** ***`interface`***
  - Specifies which of your interfaces to use to send packets from.
- **`-m`** ***`max-hops`***
  - Specified maximum number of hops, default is 30.
- **`-N`** ***`concurrent_hops`***
  - Sets number of concurrent packets to send out at once, though may result in devices ignoring/dropping sessions, and making result less reliable. Default is 6.
- **`-n`**
  - Don't lookup ip addresses , just list the addresses in lieu of URLs

https://www.unix.com/man-page/suse/1/traceroute/

https://www.suse.com/support/kb/doc/?id=7001152
https://www.freebsd.org/cgi/man.cgi?query=traceroute
https://www.suse.com/documentation/opensuse113/book_opensuse_startup/data/sec_shell_commands.html

Trace(route) syntax (SUSE) (cont.)

- **`traceroute –`*`options`*` url_or_ip address`**
  - (note in windows: tracert, and some routers: trace)
- **`-p port`**
  - Specifies which UDP port to use. Default port is 33434.
- **`-q numqueries`**
  - Sets the number of probe packets per hop. The default value is 3.
- **`-r`**
  - Bypass the normal routing tables and send directly to a host on an attached network.
- -R (NOT IMPLEMENTED YET?)
  - Set the loose source route option on outgoing packets, asking intermediate routers to record their address as the packet passes.
- **`-S source_addr`**
  - Chooses an alternative source address. Note that you must select the address of one of the interfaces. By default, the address of the outgoing interface is used.
- **`-T`**
  - Use TCP instead of UDP packets when probing the route. This option is available to the super user only, as this requires a raw ICMP socket.
- **`-t tos`**
  - Set the IP Type of Service (TOS) and Precedence value. Useful values are 16 (low delay) and 8 (high throughput).
- **`-V`**
  - Print the version and exit.
- **`-w sec`**
  - Wait for sec seconds before sending the next probe packet.

https://www.unix.com/man-page/suse/1/traceroute/

https://www.suse.com/support/kb/doc/?id=7001152
https://www.freebsd.org/cgi/man.cgi?query=traceroute
https://www.suse.com/documentation/opensuse113/book_opensuse_startup/data/sec_shell_commands.html

De facto Standards

- Emerged from use and informal agreements, to allow cross internet troubleshooting and database lookup
  - CLLI codes: One way of naming WAN servers and devices (and URLs) preserved from old AT&T Bell (pre-breakup) networks, still very much in use, with customizations, on many networks worldwide:
    - Ex: dllstx2wch010.wcg.Level3.net : dllstx2wch010: dlls=Dallas, tx=Texas, 2W: 2nd Williams PoP, ch010=Cisco Backbone Router #10 (look for the 'c')
    - Often used in reverse too: dtr02stprmo
- Airport Codes: used as well: STL = St. Louis, PIT=Pittsburgh, etc. With a 3 letter description of device type, and a number or name.
- Most fiber runs along easements.

https://en.wikipedia.org/wiki/International_Air_Transport_Association_airport_code

https://en.wikipedia.org/wiki/CLLI_code

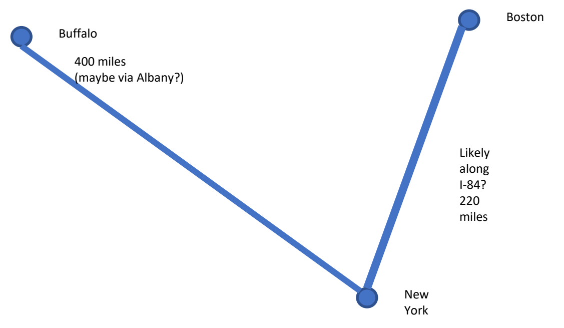
Mapping a traceroute

- Ex: to yahoo from Boston cogent:
- traceroute to www.yahoo.com (72.30.35.10), 30 hops max, 60 byte packets
- 1 gi0-0-0-15.216.agr22.bos01.atlas.cogentco.com (66.250.250.25) 0.740 ms 0.744 ms **BOSTON COGENT**
- 2 be3664.ccr31.bos01.atlas.cogentco.com (154.54.87.105) 0.553 ms **BOSTON COGENT, I bet ccr = cisco core router?, be may mean bonded ethernet? Atlas is one of their networks?**
- be3665.ccr32.bos01.atlas.cogentco.com (154.54.87.109) 0.910 ms **BOSTON COGENT**
- 3 be3472.ccr42.jfk02.atlas.cogentco.com (154.54.46.34) 6.144 ms **NY JFK COGENT**
- be3471.ccr41.jfk02.atlas.cogentco.com (154.54.40.154) 6.178 ms**NY JFK COGENT**
- 4 be3362.ccr31.jfk04.atlas.cogentco.com (154.54.3.10) 6.365 ms 6.212 ms **NY JFK COGENT**
- 5 phx-b1-link.telia.net (213.248.81.248) 6.215 ms 6.226 ms **Telia Phoenix AZ??? Not likely, likely a naming oops…they left the name on a router they moved.**
- 6 nyk-bb3-link.telia.net (62.115.140.222) 6.522 ms 6.533 ms **note NYK is in Kenya, but really this is likely NY (Manhattan?)….Telia**
- 7 buf-b1-link.telia.net (80.91.246.36) 16.130 ms 16.032 ms **Buffalo NY Tellia**
- 8 yahoo-ic-315726-buf-b1.c.telia.net (213.248.82.10) 16.640 ms 16.365 ms **Buffalo NY Tellia**
- 9 et-1-0-1.pat1.bfz.yahoo.com (72.30.223.5) 16.170 ms 16.205 ms **Likely Buffalo NY Yahoo peering connection**
- 10 et-19-1-0.msr1.bf1.yahoo.com (74.6.227.133) 16.199 ms et-19-1 -0.msr2.bf1.yahoo.com (74.6.227.141) 16.271 ms **Buffalo NY Yahoo**
- 11 et-0-1-0.clr2-a-gdc.bf1.yahoo.com (74.6.122.17) 16.227 ms et-19-0-0.clr1-a-gdc.bf1.yahoo.com (74.6.122.33) 16.658 ms **Buffalo NY Yahoo**
- 12 eth-18-3.bas2-1-flk.bf1.yahoo.com (98.139.128.75) 16.533 ms eth-18-3-bas1-1-flk.bf1.yahoo.com (98.139.128.73) 18.979 ms **Buffalo NY Yahoo**
- 13 media-router-fp2.prod1.media.vip.bf1.yahoo.com (72.30.35.10) 16.141 ms 16.149 ms **Buffalo NY Yahoo**

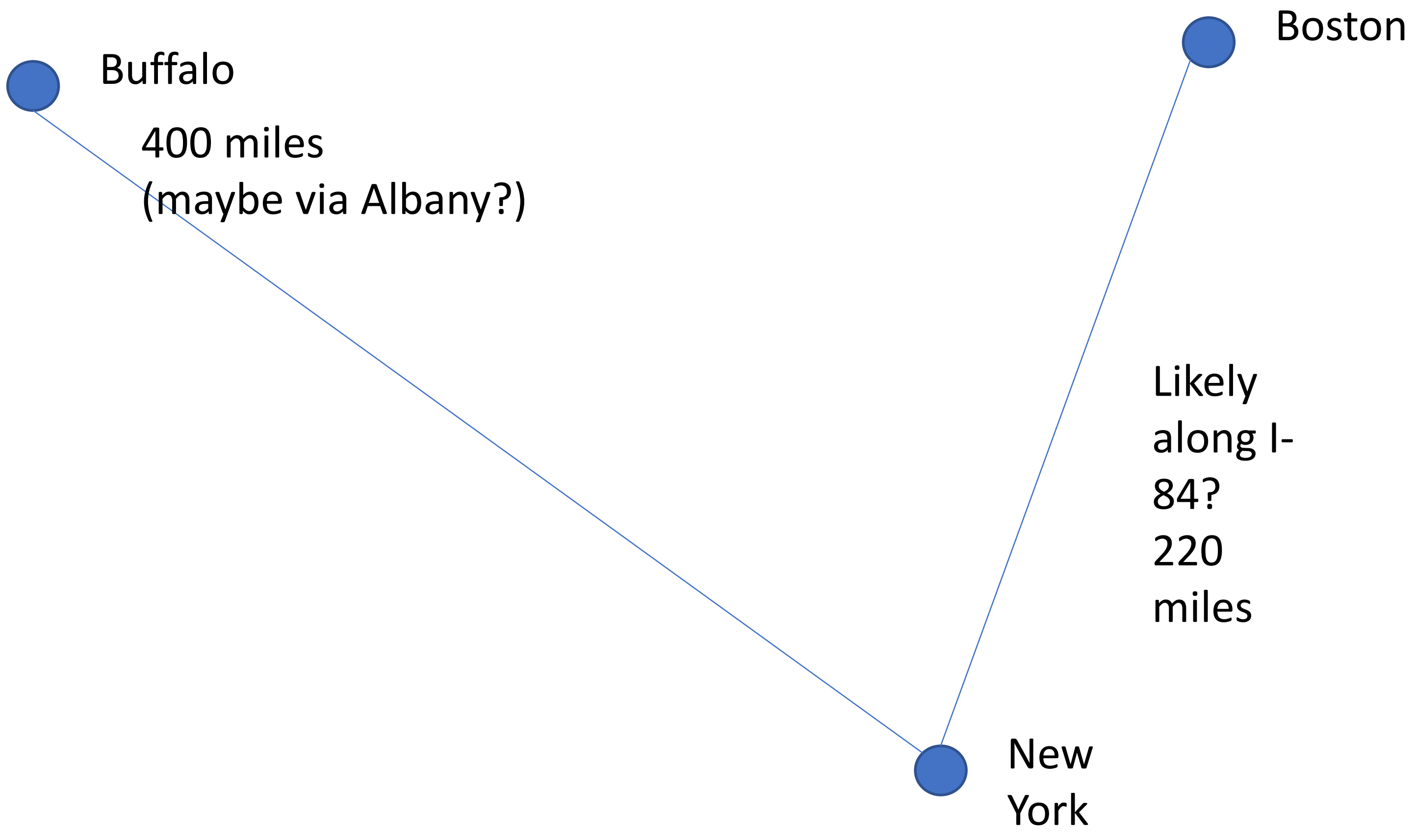
http://www.cogentco.com/en/network/network-map

http://www.teliacarriermap.com/

https://www.peeringdb.com/net/27



Physically


Buffalo
400 miles
(maybe via Albany?)
Boston
Likely along I-84?
220 miles
New York

Using Looking Glass Sites

- Looking Glass locations allow pinging and trace routes from outside locations, on other networks (and bgp commands)
- Can prevent hackers from finding out who traced them
- Used to troubleshoot across peers, pinpoint sites from multiple angles.
- http://www.bgplookingglass.com/ Is one source of many.
- Best to stick to major backbone looking glass servers.

backup

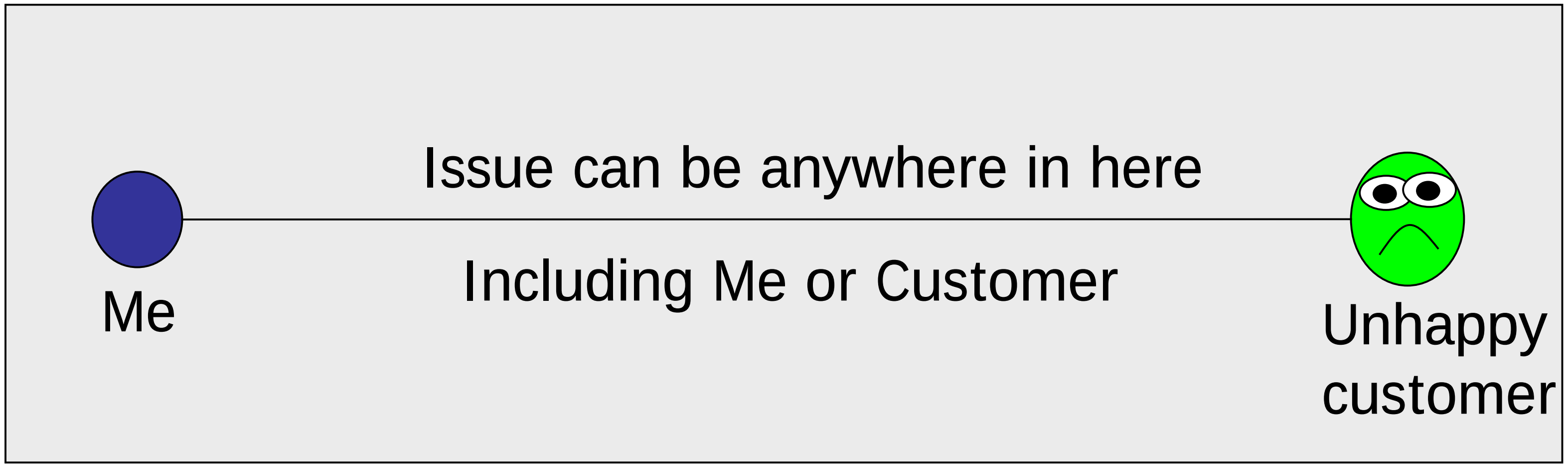
Diagnostics 101

- Job of Tiers in a Commercial Network is to quickly (=minimal steps) locate and resolve issues
  - With minimal impact to this customer, other customers, and network design/processes
- Keeping detailed track of data, no matter how trivial, will allow later analysis (maybe only hours later in some cases)
- Keeping calm is key! Standard procedures and adapt as required using processes essential.
  - Knowledge and process will reduce stress in troubleshooters.

Tier 1 Diagnostic Priorities

- Stay on issue path until evidence says otherwise
- Exhaust 'non-human required' means before escalating
  - i.e. Metrics graphs, Pings, traces, logs, etc. are cheap.
  - Balance against severity of problem and duration of issue
- Restore service then worry about long term fixes (but make sure follow-up will occur, band-aids do fall off eventually)
- Know when to escalate

Point to Point Logic: Single Failure



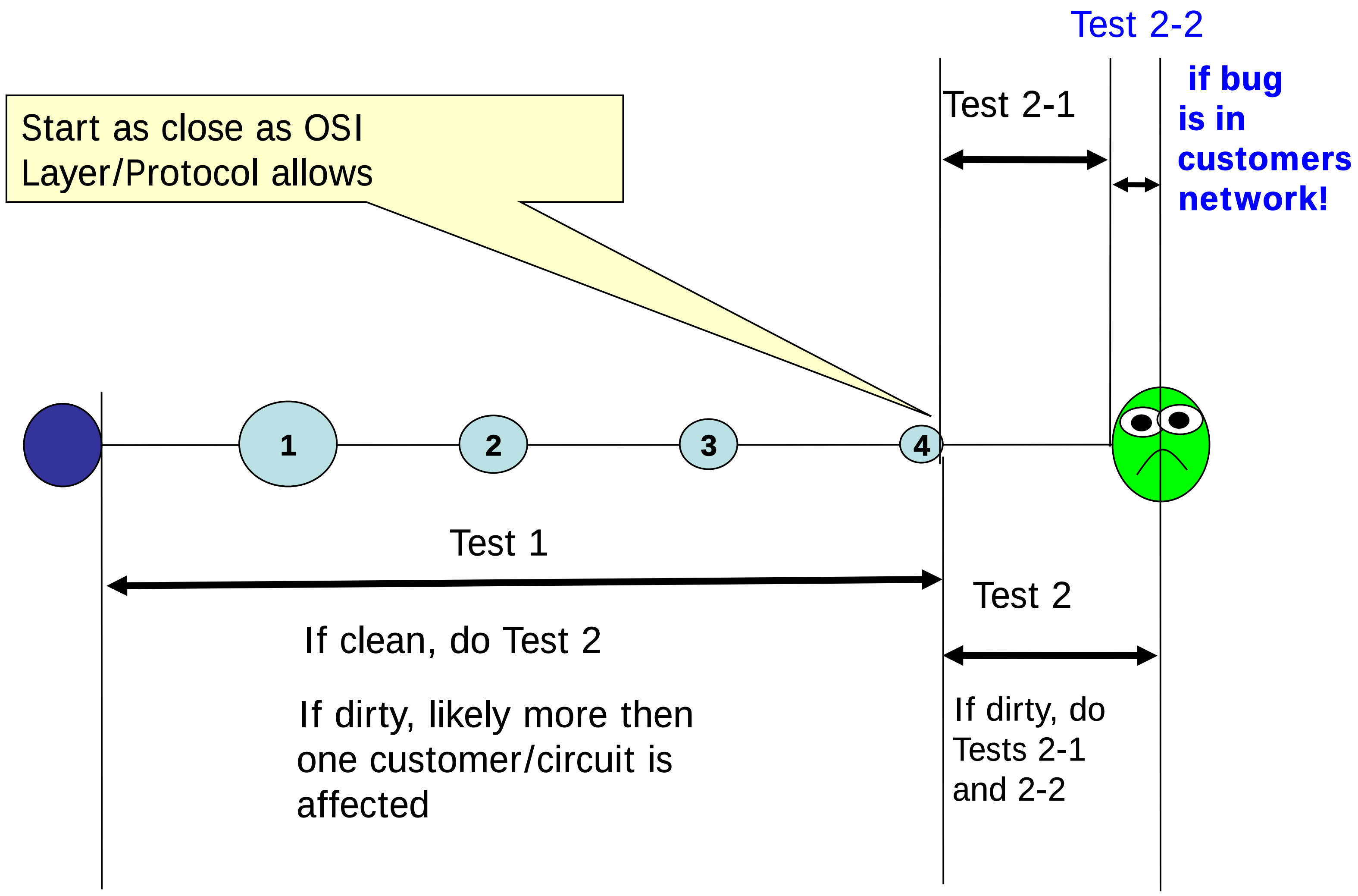
First Step: Determine if there really is a problem, and that it is stated correctly (while not insulting the customer!)

KEY WISDOM:
If no one else is having issues, it is likely closer to the customers end.
I want to clear as much of my network as possible as fast as possible.

Picturing your issue, and Maps

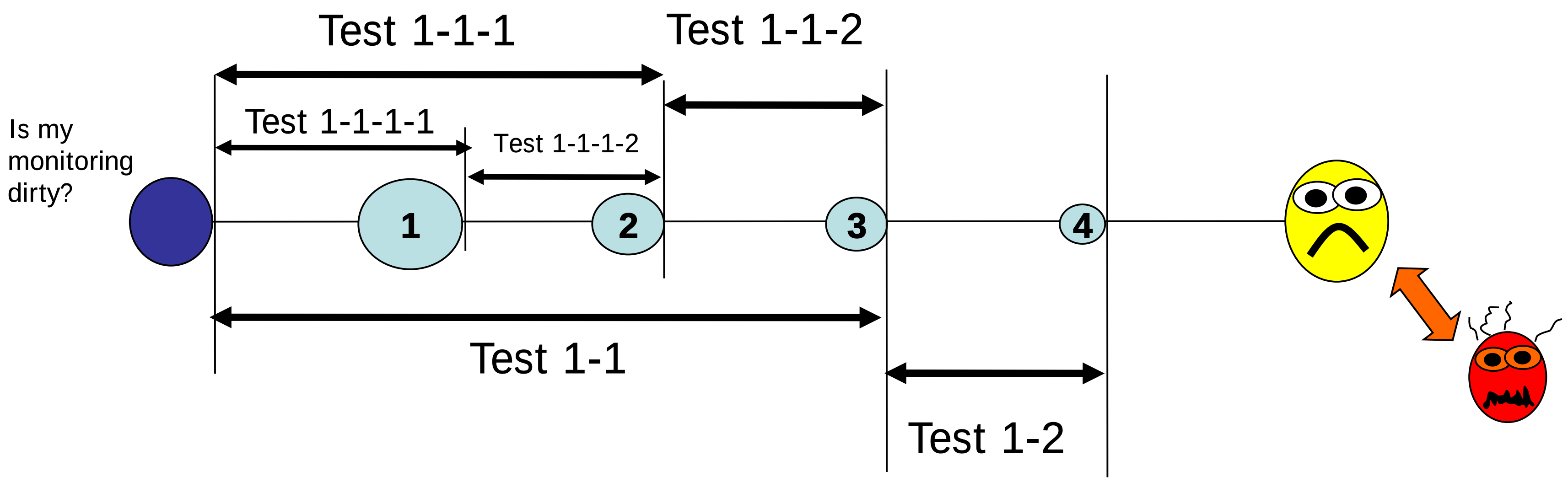
- If it is a difficult problem draw a figure (so you can hand it around), copy and paste from OSS
  - Start point and end point items, start simple.
  - All important (affected) devices of same network layer in between (use a Trace if possible, topology if available)
  - May need multiple drawings for multiple layers
  - May become more complex if more than one issue site is involved (more later)

Where is my Bug?


Test 2-2
Test 2-1
if bug is in customers network!
Start as close as OSI Layer/Protocol allows
1
2
3
4
Test 1
If clean, do Test 2
If dirty, likely more then one customer/circuit is affected
Test 2
If dirty, do Tests 2-1 and 2-2

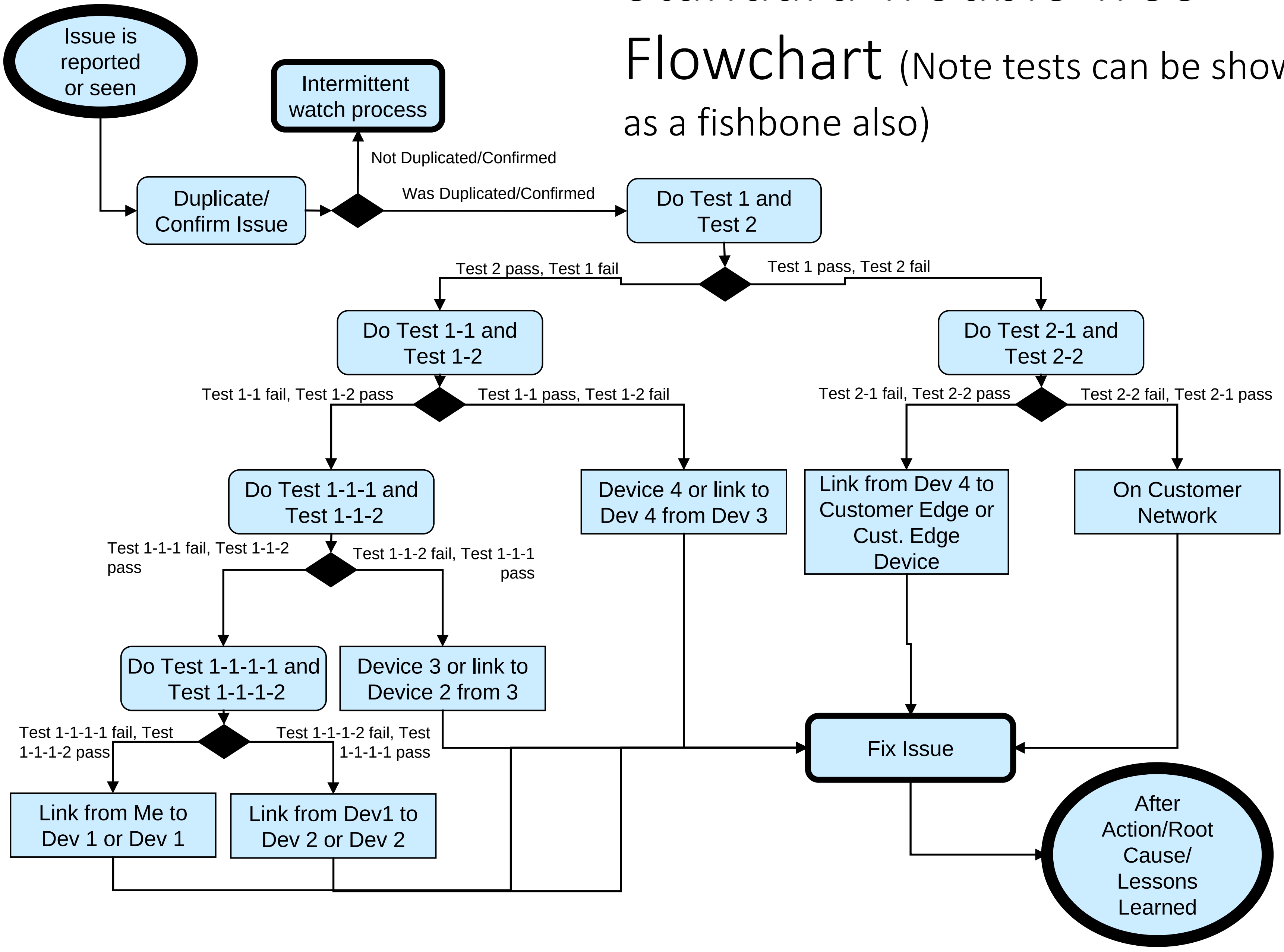
Dirty = contains degradation from a bug

On My End?



1. If Test 1 is dirty, do a Test 2 to confirm cleanness to the customer.
2. Then it gets tricky, but the same logic applies: if no one else is having issues, it is likely closer to the customers end. Also, I want to clear as much of my network as possible as fast as possible. If it is a bigger issue I need to know quickly.
3. Therefore Do tests 1-1 and 1-2. if 1-1 is clean, and 1-2 is dirty, start looking at the connection between 3 and 4 and at device 4.
4. If Test 1-1 is dirty, and Test 1-2 is clean, do tests 1-1-1 and 1-1-2

Standard Trouble Tree Flowchart (Note tests can be shown as a fishbone also)

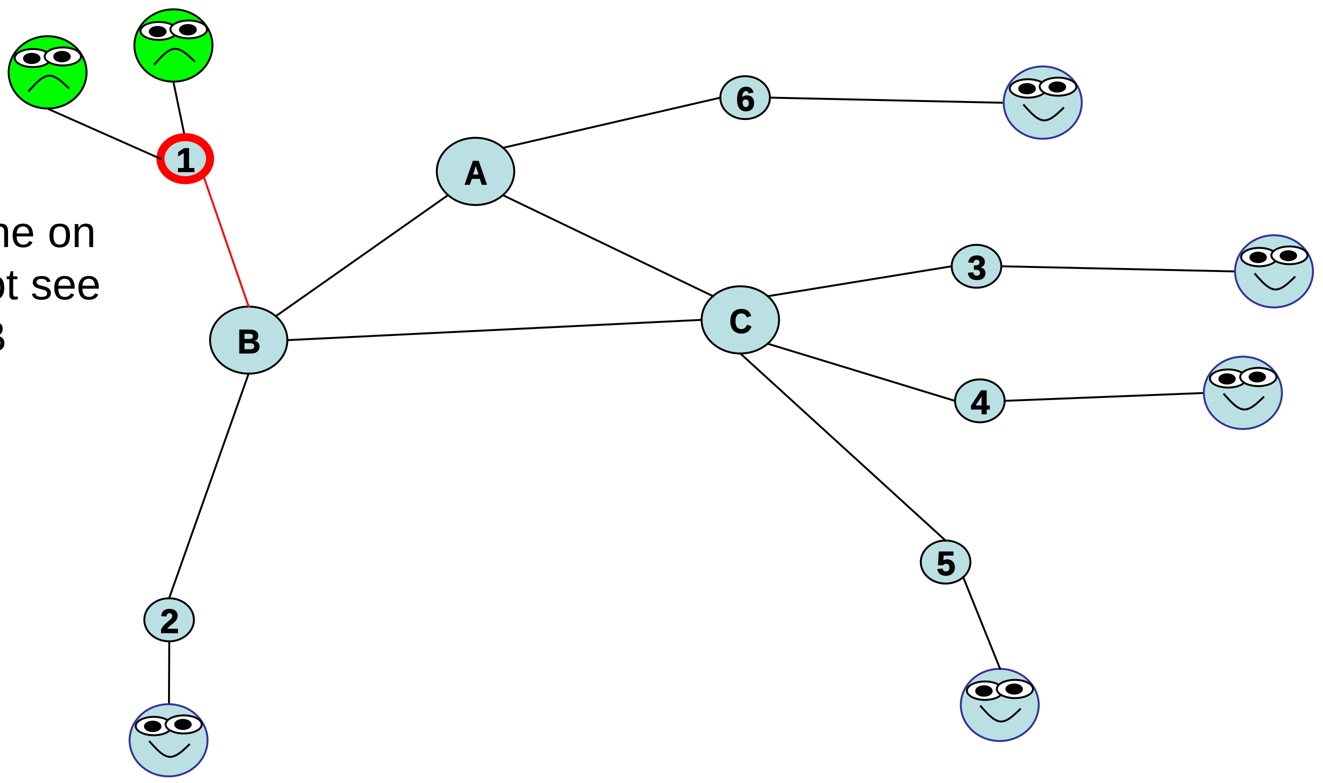


Issue is reported or seen
Duplicate/ Confirm Issue
Intermittent watch process
Not Duplicated/Confirmed
Was Duplicated/Confirmed
Do Test 1 and Test 2
Test 2 pass, Test 1 fail
Test 1 pass, Test 2 fail
Do Test 1-1 and Test 1-2
Do Test 2-1 and Test 2-2
Test 1-1 fail, Test 1-2 pass
Test 1-1 pass, Test 1-2 fail
Test 2-1 fail, Test 2-2 pass
Test 2-2 fail, Test 2-1 pass
Do Test 1-1-1 and Test 1-1-2
Device 4 or link to Dev 4 from Dev 3
Link from Dev 4 to Customer Edge or Cust. Edge Device
On Customer Network
Test 1-1-1 fail, Test 1-1-2 pass
Test 1-1-2 fail, Test 1-1-1 pass
Do Test 1-1-1-1 and Test 1-1-1-2
Device 3 or link to Device 2 from 3
Test 1-1-1-1 fail, Test 1-1-1-2 pass
Test 1-1-1-2 fail, Test 1-1-1-1 pass
Link from Me to Dev 1 or Dev 1
Link from Dev1 to Dev 2 or Dev 2
Fix Issue
After Action/Root Cause/ Lessons Learned

Multiple site issues

- 1) Look for simplest common item. Where do my problems share a common element?
- 2) Ignore items (for now) that are allowing traffic
- 3) Will look at more complex problems (i.e. layer level issues later)

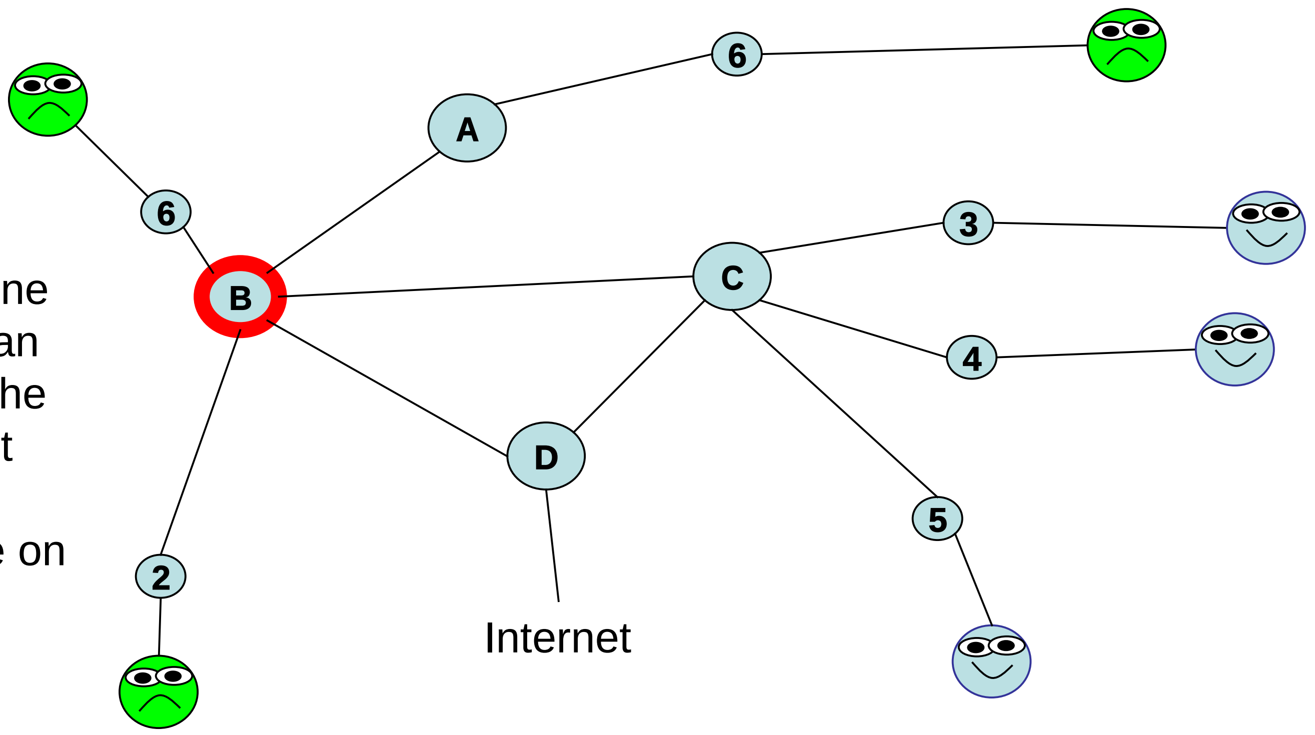
Ex:
Everyone on 1 cannot see 2-6 or B



Multiple site issues

- 1) Look for simplest common item. Where do my problems share a common element?
- 2) Ignore items (for now) that are allowing traffic
- 3) Will look at more complex problems (i.e. layer level issues later)

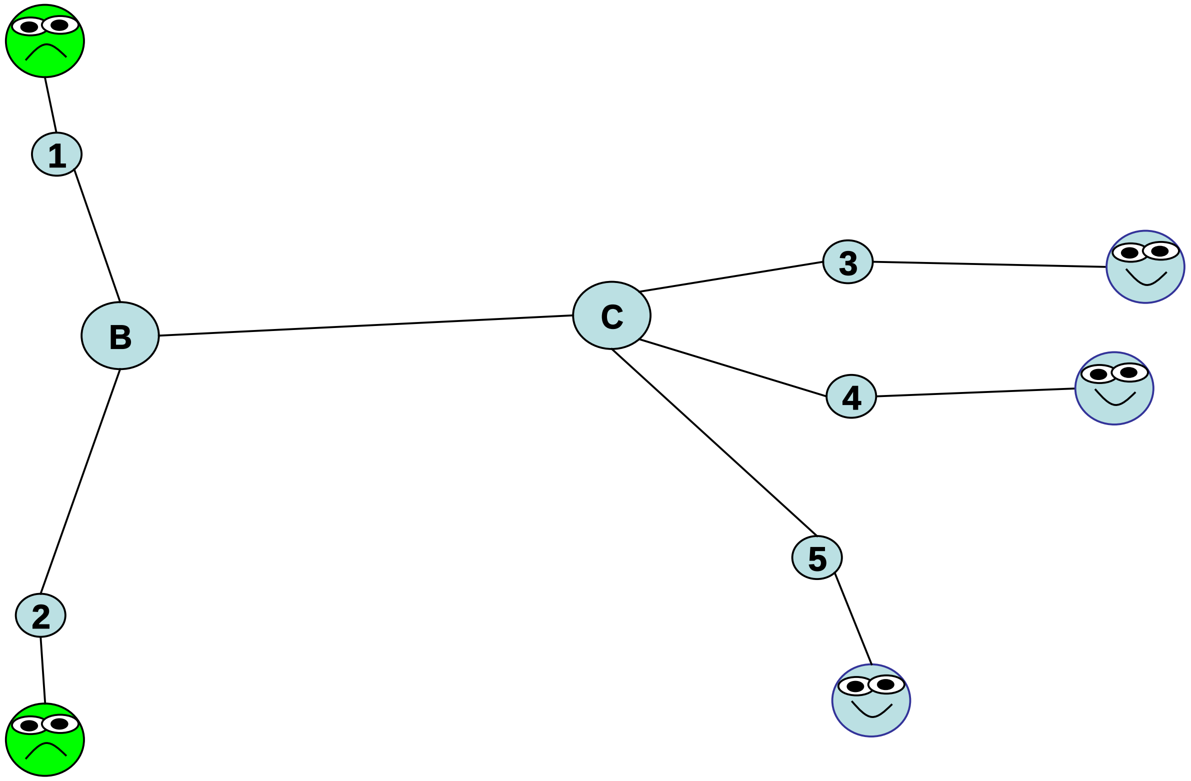
Ex:
Everyone on C can get to the Internet (or D)
No one on B can.



Trivia Point: Was a Real World one for me! Someone decommissioned B!

Multiple Sites

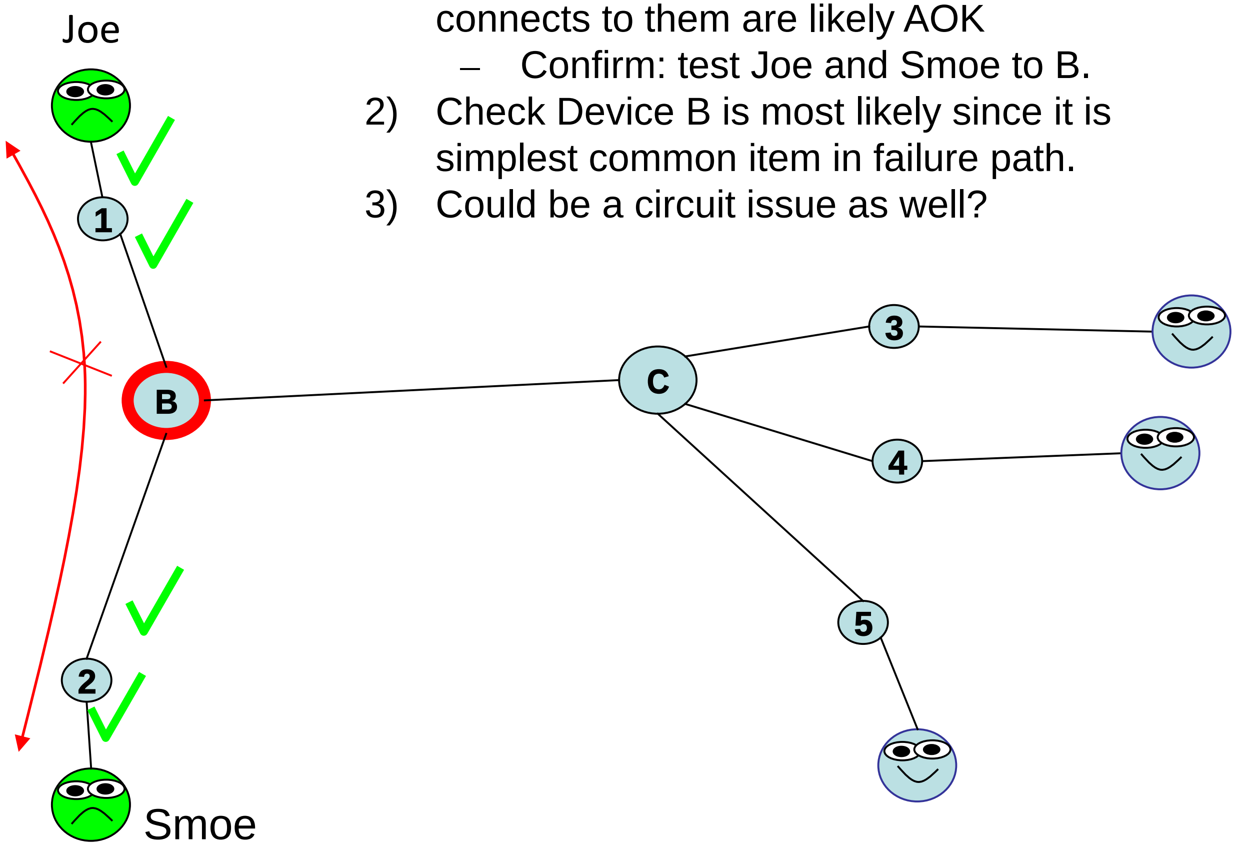
- EX: Look for simplest common item: 1 needs to go to 2,3,4,5, but 1 and 2 cannot see each other, 3-5 can see each other, but not 1 or 2.



Multiple site issues

EX: Look for simplest common item: 1 needs to go to 2,3,4,5, but 1 and 2 cannot see each other, 3-5 can see each other, but not 1 or 2.

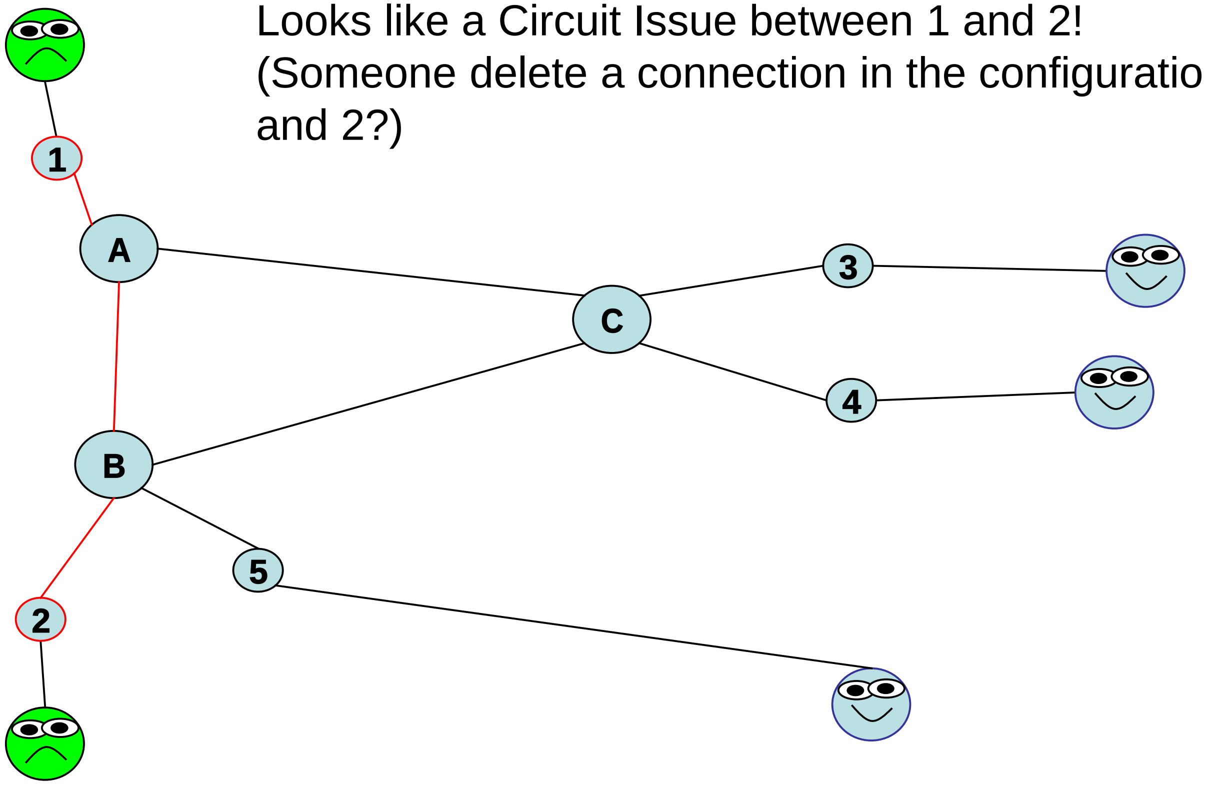
1) Two ends are down so boxes 1 and 2 and connects to them are likely AOK
   - Confirm: test Joe and Smoe to B.
2) Check Device B is most likely since it is simplest common item in failure path.
3) Could be a circuit issue as well?



Multiple site issues

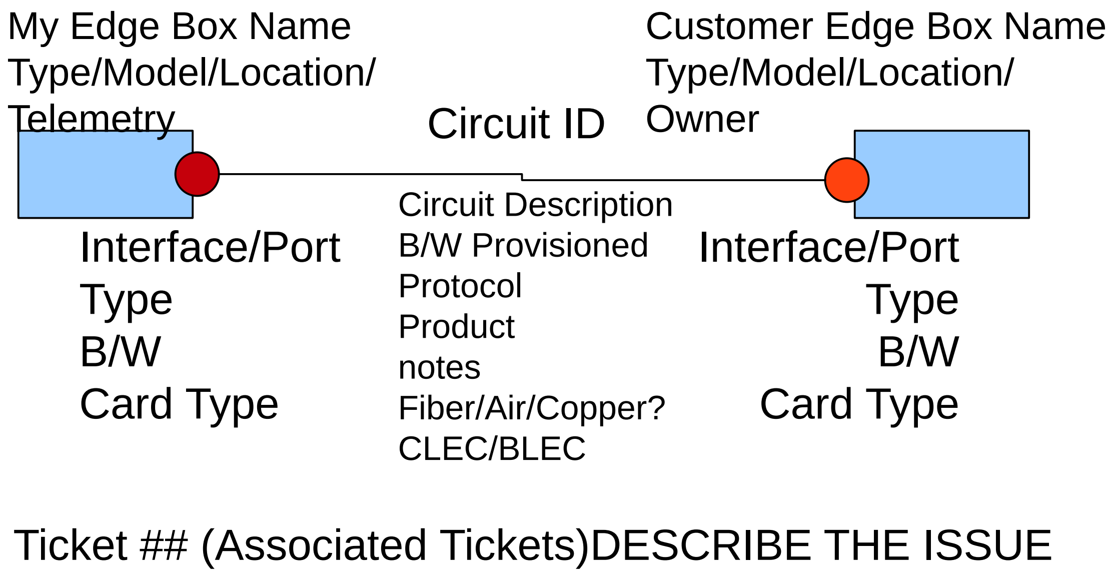
- Look for simplest common item

Looks like a Circuit Issue between 1 and 2!
(Someone delete a connection in the configurations of 1 and 2?)



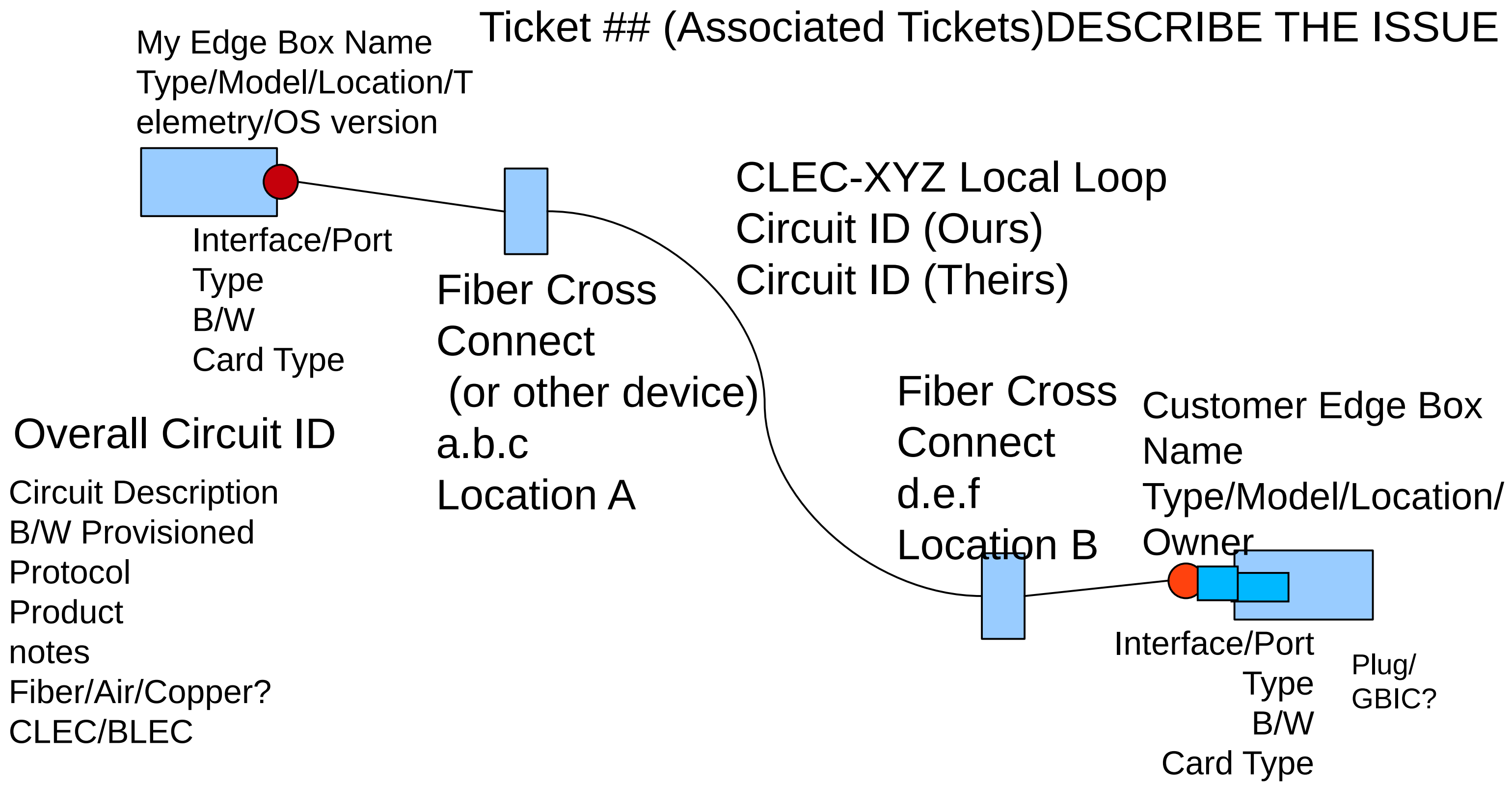
Picturing for Real

- Use a fishbone diagram to track progress! See later slides.
- Don't re-draw the entire network map, only the affected portions at the layer or layers you need
- If the problem is in multiple places, draw back to common point
- If single place start with problem ends and work backward
- Keep known clean areas un-complex



More Picturing

- Look-up or discover intermediate devices if they may affect issue


Ticket ## (Associated Tickets)DESCRIBE THE ISSUE
My Edge Box Name Type/Model/Location/Telemetry/OS version
Interface/Port Type B/W Card Type
Fiber Cross Connect (or other device) a.b.c Location A
CLEC-XYZ Local Loop Circuit ID (Ours) Circuit ID (Theirs)
Overall Circuit ID
Circuit Description B/W Provisioned Protocol Product notes Fiber/Air/Copper? CLEC/BLEC
Fiber Cross Connect d.e.f Location B
Customer Edge Box Name Type/Model/Location/ Owner
Interface/Port Type B/W Card Type
Plug/ GBIC?

https://www.unix.com/man-page/suse/1/traceroute/

- TRACEROUTE(1) General Commands Manual TRACEROUTE(1)
- NAME
-     traceroute - print the route packets take to network host
- SYNOPSIS
-     traceroute [-46FInrRTV] [-f first_ttl] [-p port]
-                [-m max_hops] [-N concurrent_hops]
-                [-t tos] [-w timeout] [-q nqueries]
-                [-S source_addr] [-i interface]
-                [-g gateway] host [packetlen]
-     traceroute6 [options]
- DESCRIPTION
-     traceroute tracks the route packets take across a TCP/IP network on their way to a given
-     host. It utilizes the IP protocol's time to live (TTL) field and attempts to elicit an
-     ICMP TIME_EXCEEDED response from each gateway along the path to the host.

-     traceroute6 is just another name for the same program, and is equivalent to invoking
-     traceroute with the -6 option.

-   Tracing with UDP
-     The default mode of operation is to use UDP packets for path detection.   This will send
-     packets   of a given size to a range of destination ports, usually in the high port range,
-     and increment the destination port number for each probe packet sent). Intermediate gate-
-     ways will return ICMP time exceeded errors when the packet's TTL reaches zero. When a
-     packet reaches the remote host, and the chosen port is not in use, the host will return an
-     ICMP port unreachable error. If the port is in use, the probe packet will be delivered to