TracePath

tracepath

tracepath and *tracepath6* trace the path to a network host, discovering MTU and asymmetry along this path. As described below, their applicability for path asymmetry measurements is quite limited, but the tools can still measure MTU rather reliably.

Methodology and Caveats

A path is considered asymmetric if the number of hops to a router is different from how much TTL was decremented while the ICMP error message was forwarded back from the router. The latter depends on knowing what was the original TTL the router used to send the ICMP error. The tools guess TTL values 64, 128 and 255. Obviously, a path might be asymmetric even if the forward and return paths were equally long, so the tool just catches one case of path asymmetry.

A major operational issue with this approach is that at least Juniper's M/T-series routers decrement TTL for ICMP errors they originate (e.g., the first hop router returns ICMP error with TTL=254 instead of TTL=255) as if they were forwarding the packet. This shows as path asymmetry.

Path MTU is measured by sending UDP packets with DF bit set. The packet size is the MTU of the host's outgoing link, which may be cached Path MTU Discovery for a given destination address. If a link MTU is lower than the tried path, the ICMP error tells the new path MTU which is used in subsequent probes.

As explained in tracepath(8), if MTU changes along the path, then the route will probably erroneously be declared as asymmetric.

Examples

IPv4 Example

This example shows a path from a host with 9000-byte "jumbo" MTU support to a host on a traditional 1500-byte Ethernet.

```
: leinen@cemp1[leinen]; tracepath diotima
 1:  cemp1.switch.ch (130.59.35.130)                                       0.203ms
pmtu 9000
 1:  swiCE2-G5-2.switch.ch (130.59.35.129)                                 1.024ms
 2:  swiLS2-10GE-1-3.switch.ch (130.59.37.2)                           1.959ms
 3:  swiEZ2-10GE-1-1.switch.ch (130.59.36.206)                     5.287ms
 4:  swiCS3-P1.switch.ch (130.59.36.221)                               5.456ms
 5:  swiCS3-P1.switch.ch (130.59.36.221)                              asymm  4        5.467ms
pmtu 1500
 6:  swiLM1-V610.switch.ch (130.59.15.230)                             4.864ms
 7:  swiLM1-V610.switch.ch (130.59.15.230)                        asymm  6        5.209ms !H
          Resume: pmtu 1500
```

The router (interface) `swiCS3-P1.switch.ch` occurs twice; on the first line (hop 4), it returns an ICMP TTL Exceeded error, on the next (hop 5) it returns an ICMP "fragmentation needed and DF bit set" error. Unfortunately this causes `tracepath` to miss the "real" hop 5, and also to erroneously assume that the route is asymmetric at that point. One could consider this a bug, as `tracepath` could distinguish these different ICMP errors, and refrain from incrementing TTL when it reduces MTU (in response to the "fragmentation needed..." error).

When one retries the `tracepath` , the discovered Path MTU for the destination has been cached by the host, and you get a different result:

```
: leinen@cemp1[leinen]; tracepath diotima
 1:  cemp1.switch.ch (130.59.35.130)                                       0.211ms
pmtu 1500
 1:  swiCE2-G5-2.switch.ch (130.59.35.129)                                 0.384ms
 2:  swiLS2-10GE-1-3.switch.ch (130.59.37.2)                           1.214ms
 3:  swiEZ2-10GE-1-1.switch.ch (130.59.36.206)                     4.620ms
 4:  swiCS3-P1.switch.ch (130.59.36.221)                               4.623ms
 5:  swiNM1-G1-0-25.switch.ch (130.59.15.237)                          5.861ms
 6:  swiLM1-V610.switch.ch (130.59.15.230)                             4.845ms
 7:  swiLM1-V610.switch.ch (130.59.15.230)                        asymm  6        5.226ms !H
          Resume: pmtu 1500
```

Note that hop 5 now shows up correctly and without an "asymm" warning. There is still an "asymm" warning at the end of the path, because a filter on the last-hop router `swiLM1-V610.switch.ch` prevents the UDP probes from reaching the final destination.

IPv6 Example

Here is the same path for IPv6, using `tracepath6` . Because of more relaxed UDP filters, the final destination is actually reached in this case:

```
: leinen@cemp1[leinen]; tracepath6 diotima
 1?: [LOCALHOST]                                                    pmtu 9000
 1:  swiCE2-G5-2.switch.ch                                                1.654ms
 2:  swiLS2-10GE-1-3.switch.ch                                       2.235ms
 3:  swiEZ2-10GE-1-1.switch.ch                                       5.616ms
 4:  swiCS3-P1.switch.ch                                                  5.793ms
 5:  swiCS3-P1.switch.ch                                                asymm  4        5.872ms pmtu
1500
 5:  swiNM1-G1-0-25.switch.ch                                        5. 47ms
 6:  swiLM1-V610.switch.ch                                                5. 79ms
 7:  diotima.switch.ch                                                         4.766ms reached
          Resume: pmtu 1500 hops 7 back 7
```

Again, once the Path MTU has been cached, `tracepath6` starts out with that MTU, and will discover the correct path:

```
: leinen@cemp1[leinen]; tracepath6 diotima
 1?: [LOCALHOST]                                                    pmtu 1500
 1:  swiCE2-G5-2.switch.ch                                                0.703ms
 2:  swiLS2-10GE-1-3.switch.ch                                       8.786ms
 3:  swiEZ2-10GE-1-1.switch.ch                                       4.904ms
 4:  swiCS3-P1.switch.ch                                                  4.979ms
 5:  swiNM1-G1-0-25.switch.ch                                        4.989ms
 6:  swiLM1-V610.switch.ch                                                6.578ms
 7:  diotima.switch.ch                                                         5.191ms reached
          Resume: pmtu 1500 hops 7 back 7
```

References

- Debian package: `iputils-tracepath - Tools to trace the network path to a remote host`
- http://www.linuxmanpages.com/man8/tracepath.8.php - online `tracepath` man page
- http://puck.nether.net/pipermail/juniper-nsp/2005-May/004320.html - Juniper tracepath issues. Note that the mail in error says 'decrements twice' instead of 'decrements once'

-- Main.FrancoisXavierAndreu & Main.SimonMuyal - 06 Jun 2005
-- Main.SimonLeinen - 26 Feb 2006
-- Main.PekkaSavola - 31 Aug 2006