Experience with IPv6 path probing

draft-naderi-ipv6-probing-00

Habib Naderi
Brian Carpenter
The University of Auckland

IEPG
November 2014

Why?

- We expect IPv6 hosts to end up with multiple addresses.
  - <rant on why this will never work deleted>
- Therefore we expect dynamic probing of paths to be common, in various disguises:
  - Happy Eyeballs
  - MPTCP
  - SCTP
  - Connection managers
  - Shim6
  - etc.

What?

- Habib studied Shim6 and MPTCP in particular, to look at probing overhead and fault recovery time.
  - Shim6: in the lab, over the Internet, and simulated at very large scale.
  - MPTCP: simulated.
- We think there are a few conclusions of general interest for any form of dynamic path probing.

Shim6 - method

- LinShim6 experiments, each with 250 path failures, between Auckland and Dublin, measuring recovery time and overhead traffic.
- The same in the lab, so with negligible RTT instead of round-the-world RTT.
- Stochastic Activity Network (SAN) simulation of 10,000 simultaneous sessions on a single site experiencing path failures.
- All running one-way continuous data transfer over TCP or DCCP.

Shim6 – results (1)

- Internet – DCCP recovery from path failure is slower than TCP and generates more probes.
  - Could only run with 2 addresses at each end .
- Lab – DCCP and TCP take about the same time to recover.
  - Difference is due to different acknowledgment strategies and differing RTTs.
  - Could run with 4, 9 or 16 address pairs, but tests with 16 address pairs often fail when the working address pair is located near the end of the list of address pairs. (i.e. recovery time exceeds transport timeout)

Shim6 – results (2)

- Large scale simulation - 10,000 simultaneous SHIM6 sessions on one site with path failures
  - Validated the model against lab tests
  - Average recovery time for 4 address pairs is 10 to 12 s.
  - Probe traffic is mainly prompt, e.g., with 4 address pairs, 93% of probes occur within 10 s. (set as shim6 timeout).
  - Average and max recovery times increase super-linearly with number of address pairs, due to exponential backoff.
  - Recovery exceeds 100 s. with 16 address pairs.
  - Even with 25 address pairs, probe traffic never exceeds about 8 Mbit/s.

MPTCP – method

- Large scale simulation - Stochastic Activity Network (SAN) simulation of MPTCP on various numbers of paths experiencing failures.
  - Validated our model against published results
  - Then ran a series of simulations to measure overhead and recovery times. (In MPTCP there are no separate probes, but TCP messages on working and broken paths act as continuous probes.)

MPTCP – results

- With up to 8 paths with RTTs between 80 and 150 ms., load in the steady state is spread across the paths.
- Total throughput increases sublinearly with more paths: e.g., we simulated a scenario where steady state throughput for 8 paths was ~25% greater than for 1 path.
- With 4 paths, throughput recovers after a single path failure in ~6 s., significantly faster than SHIM6 due to MPTCP's effectively continuous probing.
- When a path is restored, recovery to the previous steady state also takes ~6 s.
- For loss rates varying randomly up to 1%, MPTCP maintains its steady state throughput.

Operational comments

- Most site firewalls drop Shim6 extension headers.
- Because of BCP38, Source Address Dependent Routing (SADR) is necessary for effective use of multiple paths.

Conclusions for probing mechanism design

- Interaction between RTT, transport layer acknowledgement, and the path failure detection significantly affects the time taken to start recovery.
- If probing is linked to congestion control, packet loss rates also affect recovery times.
- Probe traffic is unlikely to cause overload.
- Exponential backoff leads to significantly slower recovery.
- Probing all paths in parallel leads to faster recovery times with only a minor increase in traffic.
- Probe packets should look like normal data packets.
- SADR is good.
- There is little to gain by having more than 3 alternative paths.