IPv6 & recursive resolvers:

How do we make the transition less painful?

11.5.2009

Overview of the problem

IPv6 rollout may not impact production IPv4

Rolling out dedicated IPv6 hostnames is not a good long-term solution

- Good for early adopters, not good for general public

Today, enabling AAAA on the production hostnames would adversely impact IPv4 reachability

- 0.078% of users drop off the grid
  - Assuming a user base of 600M, that's **470K** users that you broke!
- Additionally, client time-outs for IPv4 fallback when AAAA fails is between 21 and 186 **seconds**
- That's a lot of breakage!
- This is a barrier for a lot of content players

What can we do about it?

Don't roll out IPv6
- Not very practical

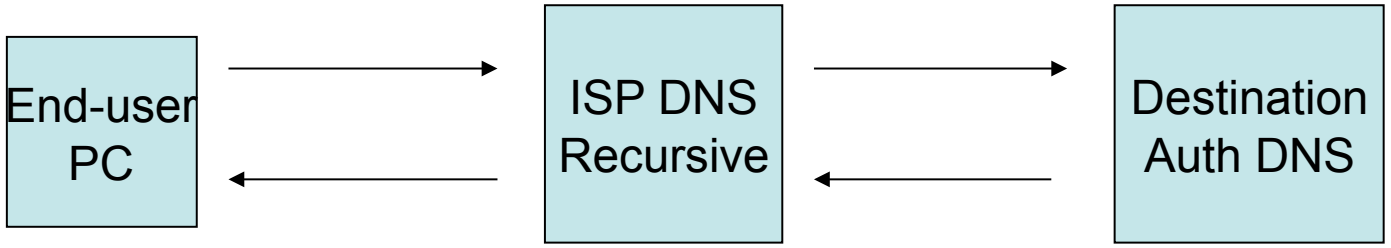
Roll out IPv6, accept the breakage
- Not very realistic

Prefer A over AAAA
- This ship has already sailed, unfortunately

Work with OS/app vendors to fix IPv6 issues
- Awful long lead times/upgrade cycles

Don't let users with broken IPv6 connectivity know about AAAA records
- Sounds good, **how do we do that?**

How do we accomplish this?



Options:

- Let the auth server decide when to serve AAAA or A records. Keep track of seemingly broken IPv6 users behind a resolver.
  - Requires a lot of instrumentation to set-up
  - Collateral damage for working IPv6 users
- ISP DNS Recursive servers does not return AAAA for users who have broken IPv6 connectivity
  - How to accurately measure working users when you are not an endpoint?
- ISP DNS recursive server only returns AAAA for users who have known working IPv6 connectivity
  - OK, sounds too good to be true, how does that work?

Proposed solution

- Only way of **knowing** the user has working IPv6 connectivity, is if the AAAA query came over IPv6!
  - ISP must roll out native IPv6 on their network, and have IPv6-addressable recursive servers deployed
  - Hand out IPv6 && IPv4 recursive server addresses to the end-users
  - Selectively return 0 answers for AAAA under limited conditions
    - Query comes over IPv4
    - DO=0 and AD=0
    - “A” record exists for same name
  - Auth DNS server now only has to worry about IPv6 reachability to the Recursive server
    - A lot easier to resolve problems at the ISP level then with individual end-users
    - A few broken IPv6 users don’t adversely impact everyone else

What does this do?

Benefits:

- Allows for IPv6 reachability issues to be resolved between NOCs
- Less support calls for “what is this IPv6 thing that broke my internets?!?!?!”
- Fewer “brokenness” with deploying IPv6 = more people may deploy it sooner

Side-effects:

- Trust -- now we have recursive servers modifying authoritative records
- This effectively turns off IPv6 for OS’s that can only do DNS queries over IPv4 (ie Windows XP)

Question posed at ISOC Operators RT and at NANOG:

Is this worth pursuing further?

Implementation - ISC

Expected to become mainline (9.7.0b2 perhaps?)

disable-aaaa-on-v4-transport (yes | no | break-dnssec);

Upon receipt of a query for an AAAA record:

- If the request has DNSSEC turned on (DO or AD bit set), return the record as requested.
- If the request comes in over IPv6 transit, return the reocrd as requested.
- If the request is over IPv4 and an A record exists at the same label, respond with NOERROR but with 0 answers, forcing the client to fall back to an A record query.

Feedback from NANOG BoF

Michigan 2009

This is a really ugly hack.

People however think this may be necessary to get widespread IPv6 adoption

Needs ability to restrict behavior based on ACL

(ie, all AAAA to get through for selected v4 addresses, and stop it from queing through for selected v6 ones -- mostly to make 6RD work)

ISC is to get back to Yahoo! re: additional cost

Other Implementations

Based on last week's ISOC presentation, two other vendors have been contacted, with requests to provide a similar implementation.

PowerDNS – Bert Hubert likes the idea, plans on implementing it

Secure64 – requested by a large access provider; unsure of the status

Questions?

Email: jfesler@yahoo-inc.com

or igor@yahoo-inc.com