

IPv6 DNS transition and deployment

Rob Austein <sra@hactrn.net>
50th IETF, Minneapolis

The problems

We have two complete IPv6 DNS solutions

One is standard, the other is deprecated

Known implementations use the deprecated one

This is becoming a real issue for IPv6 deployment

Much concern about complexity of newer stuff and whether we really need it

Some of the new stuff requires extensive infrastructure upgrades

Strong case for the advanced features of the new stuff has not been made

Overview of proposed approach for A6

Write AAAA -> A6 transition spec

Almost certainly requires protocol fiddling, hence DNSEXT work

Almost certainly will require updating or augmenting A6 spec

Write "Case For A6" or admit that we can't make one

Recruiting security folks to help with time-to-resign issues

Need to identify and address any other issues

Goal is to have both docs ready by IETF 51 in London

Yes, this is aggressive

Why A6 is worth talking about

A6 does provide features that AAAA can not provide

"Degenerate" case of A6 semantically identical to AAAA

We do not yet know whether we need A6's extra features and may not until it's too late

Paranoia therefore suggests that:

- We should deploy A6 in case we need it

- We should only use it in the degenerate case for now

None of the above to be construed as lessening our need for a "Case For A6" doc

Overview of A6 transition plan

NB: This is still wet, and smells faintly of beer

"Real" data will be A6, degenerate case only for now

Stub clients wanting AAAA to be supported by synthesis from A6 data

Synthesis to be performed by entity providing recursive service

Synthesized data probably will not be signed

If a query does not contain EDNS indicator, additional section IPv6
addresses to be AAAA

Root and TLD zones contain only (degenerate) A6, not AAAA

Binary labels

Proposal: punt 'em

Binary labels do not provide any features that can't be provided by "nibbles"

Both are ugly. Both need better user interfaces.

Binary labels are painful to deploy, because of the new label type

DNAME can ease some of the pain of the "nibble" solution

DNAME

Very dangerous, but also potentially useful

DNAME does provide new functionality that it would be difficult to provide any other way

Not quite impossible (forests of CNAMEs), just prohibitively painful

Deployment problem not as bad as binary labels

Can make "nibble mode" reverse tree less painful

Recommendation: keep DNAME, but discourage gratuitous use

Easy to say, much harder to do