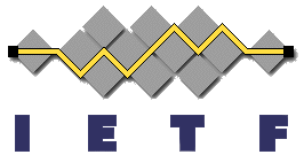
Avoiding NAT64 with dual-stack host for local networks

IETF #80 Behave WG, 29-March-2011

draft-korhonen-edns0-synthesis-flag
draft-savolainen-heuristic-nat64-discovery
draft-korhonen-behave-nat64-learn-analysis

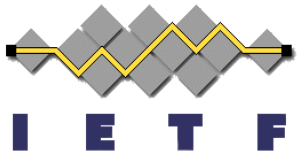
Jouni Korhonen, Teemu Savolainen



Contents

- Updates since IETF#79
  - Analysis draft
  - EDNS0 solution draft
  - Heuristics solution draft
- Experimenting
- Next steps

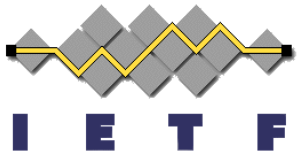
Update of “Analysis of ... hosts to learn NAT64 prefix”



- Issue #4 (The problem of supporting changing NSP) expanded:
  - The NSP learned by the host may become stale for multiple reasons. For example, the host might move to a new network that uses different NSP, thus making the previously learned NSP stale. Also, the NSP used in the network may be changed due administrative reasons, thus again making previously learned NSP stale.

- Issue #5 (The problem of supporting multiple NSPs) expanded:
  - A network may be configured with multiple NSPs for address synthesis. For example, for load-balancing purposes each NAT64 device in the same network could be assigned with their own NSP. It should be noted that learning a single NSP is enough for an end host to successfully perform local IPv6 address synthesis but to avoid NAT64 the end host needs to learn all NSPs used by the access network.

Update of “Analysis of ... hosts to learn NAT64 prefix” cont’d



- Added a note & references to referral objects
  - “... synthesized addresses are not distinguishable from public IPv6 addresses...” [ED: this text needs to be revisited]
- Added new discovery mechanisms:
  - Access technology/link layer specific mechanisms.
- Shaped & rearranged our conclusions:
  - “... recommend publishing the Well-Known DNS Name heuristic-based method as an Informational IETF document for applications and host implementers to implement as-is. If Standards Track work is seen beneficial, then our recommendation is the standardization of ENDS0 option..”

Update of “EDNS0 Option ... Record Synthesis and Format”

- DNSSEC considerations added:
  - “... When sending AAAA query for the known name a host MUST set "Checking Disabled (CD)" bit to zero, as otherwise the DNS64 will not perform IPv6 address synthesis..”

- Host behavior clarifications:
  - Check all received AAAAs in case of synthesis.
  - How to handle NXDOMAIN or empty response.
  - Pref64 selection order (NSP::/96, WKP::/96 and NSP::/nn) to minimize issues with RFC6052 suffixes ‘reserved for future use’. No relation to Default Address Selection.
  - Use of the EDNS0 option without the *well-known name*.

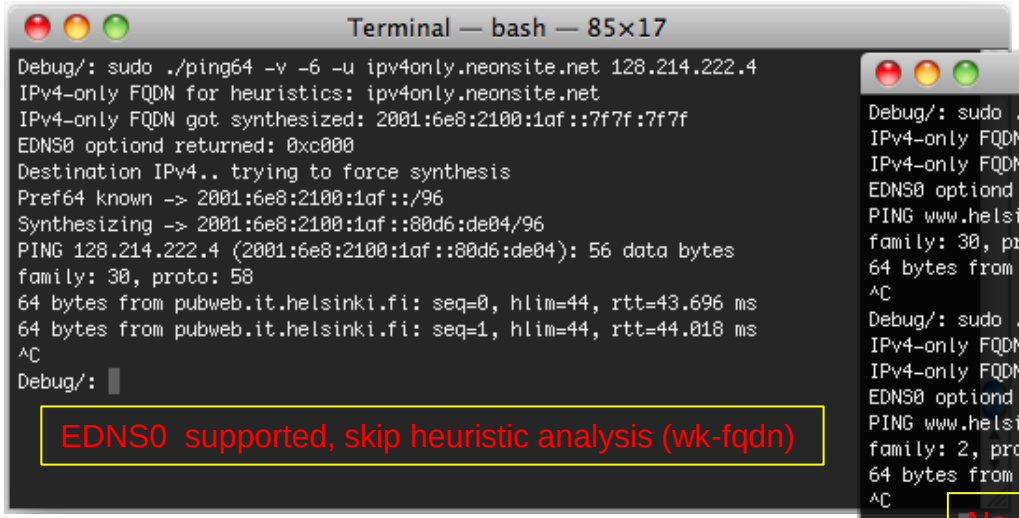
Update of “Discovery of ... NAT64 Prefix using a Well-Known Name”

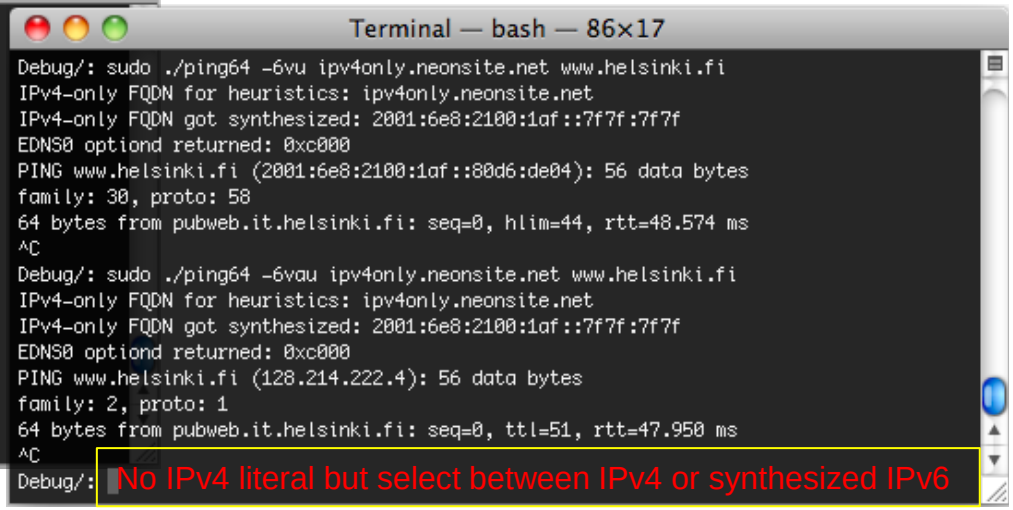
- DNSSEC considerations added – see previous slide.
- Host behavior clarifications – see previous slide (except for EDNS0 specifics).
- Guidelines for IPv4 addresses used in Well-Known Name (e.g. use of 127.127.127.127):
  - For combined connectivity testing purposes the IPv4 address would be a public routable address.. (query for A RR as well).
- Discussion about non-RFC6052 address formats and how to handle them.

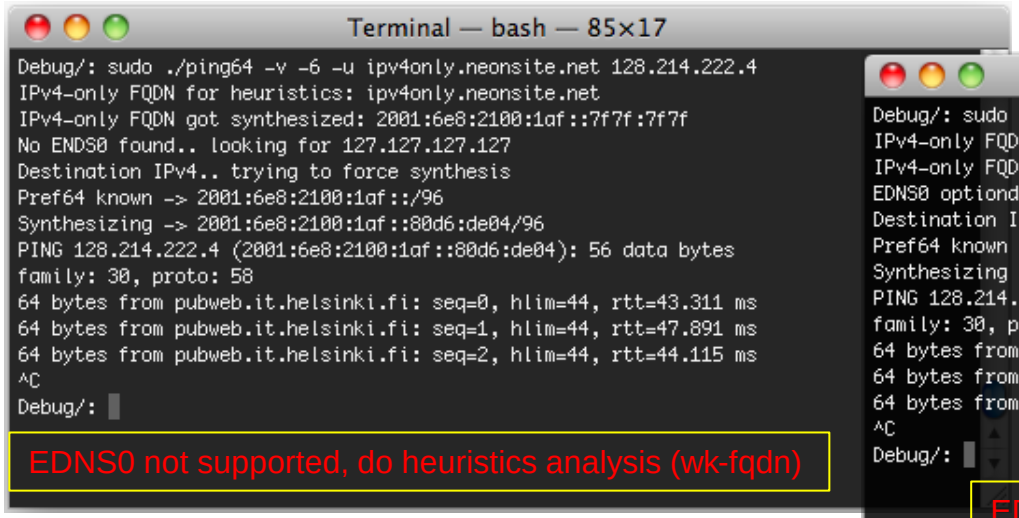
Experimenting

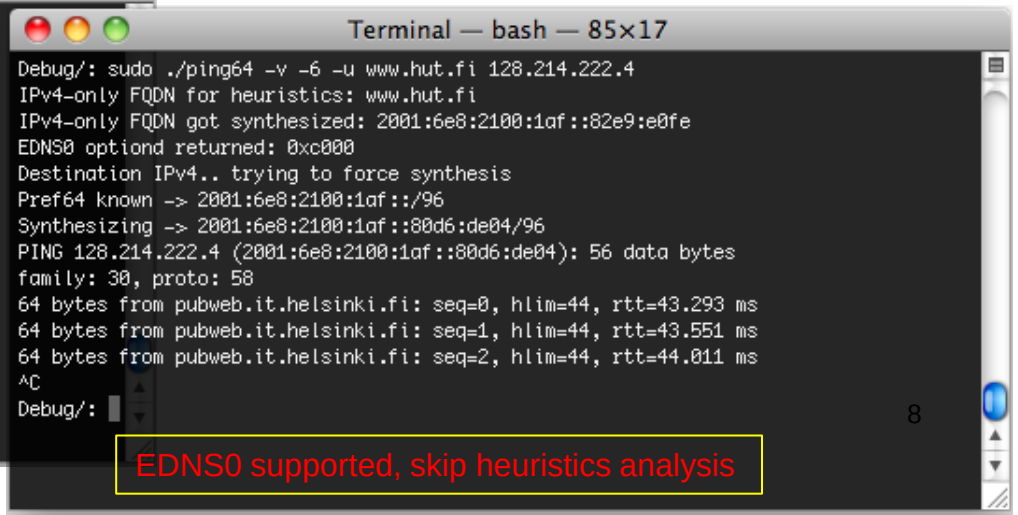
- A quick ‘ping64’ hack using both:
  - EDNS0 option aware “resolver”.
  - Heuristics (127.127.127.127 as the IPv4 pattern)
  - ...not all nits from drafts are included though..
- Also EDNS0 modification to DNS64 (Ecdysis’ bind).
- Well.. it works ;-) and was easy & straightforward. Adding support to applications is minor (and if you do heuristics, EDNS0 is not much extra to add).

I can see a ping..









Next steps?

- Analysis draft... get it finally published?

- Solutions (due April 2011 as per charter)
  - Two proposals on table..
  - Pick one and get it published?