

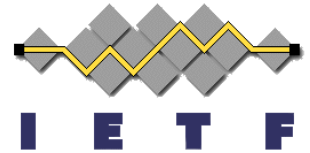
BEHAVE WG

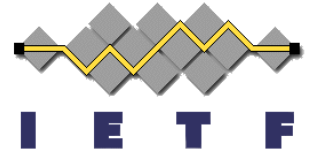
IETF 78

On Learning NAT64 Prefix..

Monday, July 26, 2010

Jouni Korhonen, Teemu Savolainen





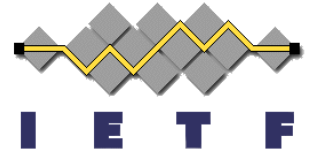
Background

- Two issues actually:
 - 1) Finding out whether a particular address is synthetic.
 - 2) Finding out how to construct an IPv6 address that will be routable to/by the NAT64 if one happen to have only an IPv4 address.

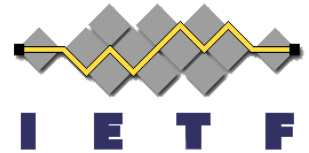
Background cont'd

- Certain applications, operating in translation scenarios, can benefit from knowing the IPv6 prefix used by NAT64:
 - The Framework document Scenario 1 and Scenario 5.
 - Cases where NSPs are used instead of the WKP.
- DNS64 cannot serve applications that are not using DNS:
 - Such applications could still work through NAT64, provided they are able to create locally valid IPv6 presentations of peers' IPv4 addresses.

Examples of cases when knowing NAT64 prefix is useful

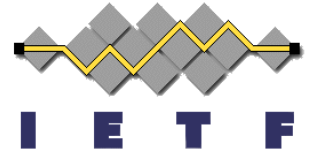


See draft-wing-behave-learn-prefix-04, section 2



Seven ways to skin a cat..

1. EDNS0 flags, <http://tools.ietf.org/html/draft-korhonen-edns0-synthesis-flag-00>
2. EDNS0 OPTION-CODE <http://tools.ietf.org/html/draft-korhonen-edns0-synthesis-flag-00#section-3>
3. Heuristics based on a well-known hostname that has only an A record <http://tools.ietf.org/html/draft-korhonen-edns0-synthesis-flag-00#section-4>
4. Special IANA-assigned DNS name <http://tools.ietf.org/html/draft-wing-behave-learn-prefix-04#section-4.1>
5. Using DHCPv6 to convey the NAT64's address <http://tools.ietf.org/html/draft-wing-behave-learn-prefix-04#section-4.2>
6. IPv6 Router Advertisements <http://tools.ietf.org/html/draft-wing-behave-learn-prefix-03#section-3.3>
7. Special DNS RR (A64) <http://tools.ietf.org/html/draft-boucadir-behave-dns-a64-01>



Proposal #1 – EDNS0 flags

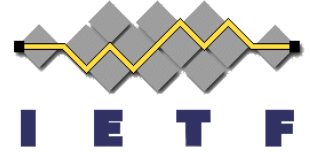
- Defines 3 new flag bits into EDNS0 OPT header.
- Benefits: Querying host learns
 - the presence of NAT64 (solves issue 1)
 - the NSP length of the synthesized AAAA response (solves issue 2)
- Issues:
 - Needs end host support
 - Need to get EDNS0 options and flags from DNS resolver to applications (API enhancements required).

Proposal #2 – new EDNS0 OPTION-CODE



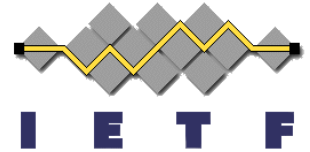
- Functionally the same as proposal #1 but defines new EDNS0 OPTION-CODE instead of flag bits.
- Benefits (over proposal #1):
 - Does not consume scarce flag bits.
- Issues (over proposal #1):
 - Has more on wire overhead.

Proposal #3 – known FQDN that only returns A RRs



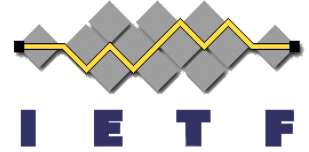
- Use a well-known FQDN that is only provisioned with A records (e.g. `ipv4.example.com`).
- Benefits:
 - Does not require any standards activity in IETF.
 - No resolver/stack impacts for learning translation taking place.
- Issues:
 - Might require solutions 1 & 2 anyway for prefix length.
 - Requires additional DNS queries.
 - Addresses particularly issue 1.

Proposal #4 – special IANA registered names



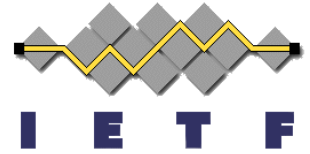
- New U-NAPTR application or TXT record.
- Benefits:
 - No host stack changes. Can be done on applications.
 - Mostly DNS provisioning problem.
 - Solves issues 1 and 2.
- Issues:
 - NAPTR handling needs more detail.
 - Requires resolver support for U-NAPTR.
 - Entails more DNS queries.

Proposal #5 – DHCPv6 conveys the NAT64 prefix

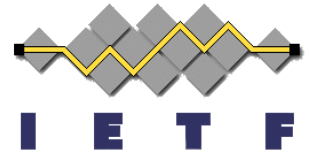


- Use new DHCPv6 option and optionally with DNSSEC to authenticate the learned prefix.
- Benefits:
 - Gives a host the information it needs when it gets the address (or later when retrieving other configuration information).
 - Address change flushes state.
 - Solves issues 1 and 2.
- Issues:
 - Requires tight DHCPv6, DNS and NAT64 sync.
 - New option needs to be understood at DHCPv6 clients.
 - Need to get option from DHCPv6 client to applications (API enhancements required).
 - Might require more DNS queries.

Proposal #6 – RA conveys the NAT64 prefix



- Use new RA option to send prefixes from NAT64 or by “other” routers.
- Benefits:
 - Couple NAT64 related routing information with routers.
- Issues:
 - Requires host stack changes.
 - May be hard to pass information to applications.
 - May need prefix verification (see proposal #5).
 - May need IGP extensions for better deployability.

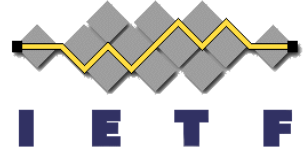


Proposal #7 – New RRTYPE

- Use new Resource Record to find out whether an address has been synthesized.
- Benefits:
 - Synthetic address can be distinguished from a “real” addresses.
- Issues:
 - Requires host resolver changes.
 - Requires standardizing a new RR.
 - Coordinated deployment to be useful..
 - Addresses mainly issue 1.

Comparison table

	Host stack impact	Resolver impact	Provisioning impact for ISP	Host API changes	Protocol change (IETF)	Synthesis detection	NSP learning	Extra signaling round (RTT)
EDNS0 flags	NO	YES	LOW	YES	YES, NEW EDNS0 FLAGS	EXPLICIT	YES	NO
EDNS0 opt-code	NO	YES	LOW	YES	YES, NEW EDNS0 OPTION	EXPLICIT	YES	NO
Heuristics	NO	NO	NONE to LOW	NO	NO	IMPLICIT	HEURISTICS	ONCE
IANA DNS name	NO	NEW U-NAPTR APP	MEDIUM	NO	NO	IMPLICIT	YES	ONCE
DHCPv6	NEW OPTION	NO	HIGH	YES	YES	IMPLICIT	YES	OPTIONALLY MANY (DNS)
RA	YES	NO	HIGH	YES	YES	IMPLICIT	YES	OPTIONALLY MANY (DNS)
A64 RR	NO	YES	LOW	YES	YES, NEW DNS RR	EXPLICIT	NO	NO



Summary

- 7 different solution proposals for the same problem hints there is a problem to be solved.
- Let's evaluate impact of each alternative and settle on the compromises we want to make.
- A candidate work for the upcoming new charter?

Feedback?

