Behave WG
Internet-Draft
Intended status: Informational
Expires: August 19, 2011

T. Savolainen
Nokia
J. Korhonen
Nokia Siemens Networks

February 15, 2011

Discovery of a Network-Specific NAT64 Prefix using a Well-Known Name
draft-savolainen-heuristic-nat64-discovery-01.txt

Abstract

This document describes a method for detecting presence of DNS64 and for learning IPv6 prefix used for protocol translation on an access network without explicit support from the access network. The method depends on existence of a known IPv4-only domain name. The information learned enables applications and hosts to perform local IPv6 address synthesis and on dual-stack accesses avoid traversal through NAT64.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on August 19, 2011.

Table of Contents

1. Introduction

As part of the transition to IPv6 NAT64 [I-D.ietf-behave-v6v4-xlate-stateful] and DNS64 [I-D.ietf-behave-dns64] technologies will be utilized by some access networks to provide IPv4 connectivity for IPv6-only hosts. The DNS64 utilizes IPv6 address synthesis to create local IPv6 presentations of peers having only IPv4 addresses, hence allowing DNS-using IPv6-only hosts to communicate with IPv4-only peers.

However, DNS64 cannot serve applications not using DNS, such as those receiving IPv4 address literals as referrals. Such applications could nevertheless be able to work through NAT64, provided they are able to create locally valid IPv6 presentations of peers' IPv4 addresses.

Additionally, DNS64 is not able to do IPv6 address synthesis for hosts running validating DNSSEC enabled resolvers, but instead the synthetization must be done by the hosts. In order to perform IPv6 synthesis hosts have to learn the IPv6 prefix(es) used on the access network for protocol translation.

This document describes a best effort method for advanced applications and hosts to learn the information required to perform local IPv6 address synthesis. An example application is a browser encountering an IPv4 address literal in an IPv6-only access network. Another example is a host running validating security aware DNS resolver.

The knowledge of IPv6 address synthetization taking place may also be useful if DNS64 and NAT64 are present in dual-stack enabled access network. In such cases hosts may choose to prefer IPv4 in order to avoid traversal through protocol translators.

The described method is intented for the scenarios where network assisted NAT64 and prefix discovery solutions are not available.

2. Requirements and Terminology

2.1. Requirements

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

2.2. Terminology

Known Name: a fully qualified domain name known by the implementation to have only an A record. Implementation knows it by hard-coding or e.g. via some provisioning technology. The name is not known by everybody.

Well-Known IPv4-only Name: a fully qualified domain name well-known to have only A record.

3. Host behavior

A host requiring information about presence of NAT64 and the IPv6 prefix used for protocol translation shall send a DNS query for AAAA records of a known IPv4-only fully qualified domain name. This may happen, for example, at the moment the host is configured an IPv6 address of a DNS server. This may also happen at the time when first DNS query for AAAA record is initiated. The host may perform this check in both IPv6-only and dual-stack access networks.

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 hence does not reveal the IPv6 prefix(es) used for protocol translation.

A DNS reply with one or more non-empty AAAA records indicates that the access network is utilizing IPv6 address synthesis. A host MUST look through all of the received AAAA records to collect all available prefixes. The prefixes may include Well-Known Prefix or one or more Network-Specific Prefixes. In the case of NSPs the host SHALL search for the IPv4 address inside of the received IPv6 addresses to determine used address format.

An IPv4 address inside synthesized IPv6 address should be found at some of the locations described in [RFC6052]. If the searched IPv4 address is not found on any of the standard locations the network must be using different formatting. In such case the host may try to find out the IPv4 address at some other location.

The host should ensure a 32-bit IPv4 address value is present only once in an IPv6 address. In case another instance of the value is found inside the IPv6, the host shall repeat the search with another IPv4 address.

In the case only one IPv6 prefix was present in the DNS response: a host shall use that IPv6 prefix for both local synthetization and for detecting synthesis done by the DNS64 entity on the network.

In the case multiple IPv6 prefixes were present in the DNS response: a host SHOULD use all received prefixes when determining whether other received IPv6 addresses are synthetic. However, for selecting prefix for the local IPv6 address synthesis host MUST use the following prioritization order, of which purpose is to avoid use of prefixes containing suffixes reserved for the future [RFC6052]:

1. Use NSP having /96 prefix

2. Use WKP prefix

3. Use longest available NSP prefix

In the case of NXDOMAIN or empty AAAA reply: the DNS64 is not available on the access network, network filtered the well-known AAAA query on purpose, or something went wrong in the DNS resolution. All unsuccesful cases result in unavailability of a host to perform local IPv6 address synthesis. The host MAY periodically resend AAAA query to check if DNS64 has become available or temporary problem cleared. The host MAY also continue monitoring DNS replies with IPv6 addresses constructed from WKP, in which case the host MAY use the WKP as if it were learned during the query for well-known name.

3.1. Connectivity test

After the host has obtained a candidate prefix and format for the IPv6 address synthesis it may locally synthesize an IPv6 address, by using a publicly routable IPv4 address, and test connectivity with the resulting IPv6 address. The connectivity test may be conducted e.g. with ICMPv6 or with a transport layer protocol. The used public IPv4 address may be learned via separate A query.

This connectivity test ensures local address synthetization results in functional and protocol translatable IPv6 addresses.

3.2. IPv4 addresses of the known name

The IPv4 addresses of the known name should be such that they are unlikely to appear more than once within an IPv6 address and also as easy as possible to find from within the synthetic IPv6 address. Good addresses might be 127.127.127.127 as a primary and 192.168.127.254 as a secondary. The secondary address is needed in the case multiple instaces of primary address are present in a synthetic IPv6 address. The IPv4 addresses can, however, be publicly routable especially if also used for the connectivity test.

3.3. Non-standard IPv6 address formats

A node may need to perform more complex heuristics to cope with networks possibly using non-standard IPv6 address formats. Non-standard approaches might include for example:

1. Non-standard location: IPv4 address in one piece at non-standard location. Can be found by pattern matching.

2. Fragmented: IPv4 address in multiple pieces around the IPv6 address. May be found by pattern matching.

3. Obfuscated address: IPv4 address is obfuscated, for example xorred. May potentially be found especially if standard addess format is used, but as this is an indication of access network's unwillingness to support host based synthetization the host should not try to decipher the IPv6 prefix.

4. Hosting of an IPv4-only name(s)

The required IPv4-only name has to be hosted by someone. While IANA(?) might host one (?), it may be safest for device, operating system, and/or application vendors to host IPv4-only names for their own uses. The name should have two A records in order to manage in situations where the first IPv4 address appears more than once within synthetic IPv6 address. Another name may be needed for connectivity test purposes.

5. Required IPv4 addresses

A prefix detection without connectivity test does not require any routable IPv4 addresses. The connectivity test requires a routable IPv4 address on the server side.

6. Security Considerations

No security considerations have been identified.

7. IANA Considerations

IANA(?) should define a name and an IPv4 address for a Well-Known IPv4-only Name.

8. Acknowledgements

Authors would like to thank Dan Wing, Washam Fan, Cameron Byrne, and Christian Huitema for improvement ideas and comments.

9. Normative References

[I-D.ietf-behave-dns64]
Bagnulo, M., Sullivan, A., Matthews, P., and I. Beijnum, "DNS64: DNS extensions for Network Address Translation from IPv6 Clients to IPv4 Servers", draft-ietf-behave-dns64-11 (work in progress), October 2010.

[I-D.ietf-behave-v6v4-xlate-stateful]
Bagnulo, M., Matthews, P., and I. Beijnum, "Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers", draft-ietf-behave-v6v4-xlate-stateful-12 (work in progress), July 2010.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC6052] Bao, C., Huitema, C., Bagnulo, M., Boucadair, M., and X. Li, "IPv6 Addressing of IPv4/IPv6 Translators", RFC 6052, October 2010.

Authors' Addresses

Teemu Savolainen
Nokia
Hermiankatu 12 D
FI-33720 Tampere
Finland

Email: teemu.savolainen@nokia.com

Jouni Korhonen
Nokia Siemens Networks
Linnoitustie 6
FI-02600 Espoo
Finland

Email: jouni.nospam@gmail.com