PCP Working Group M. Boucadair
Internet-Draft France Telecom
Intended status: Standards Track May 22, 2013
Expires: November 23, 2013

Learn NAT64 PREFIX64s using PCP
draft-ietf-pcp-nat64-prefix64-01

Abstract

This document defines a new PCP extension to learn the IPv6 prefix(es) used by a PCP-controlled NAT64 device to build IPv4-embedded IPv6 addresses. This extension is needed for successful communications when IPv4 addresses are used in referrals.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

This document defines a new PCP extension [RFC6887] to inform PCP clients about the Pref64::/n [RFC6052] used by a PCP-controlled NAT64 device [RFC6146]. It does so by defining a new PREFIX64 option.

This extension is required to help establishing communications between IPv6-only hosts and remote IPv4-only hosts.

Some illustration examples are provided in Section 5. Detailed experiment results are available at [I-D.boucadair-pcp-nat64-experiments].

The use of this PCP extension for NAT64 load balancing purposes ([I-D.zhang-behave-nat64-load-balancing]) is out of scope.

2. Requirements Language

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].

3. Problem Statement

3.1. Issues

This document proposes a deterministic solution to solve the following issues:

- o Learn the Pref64::/n used by an upstream NAT64 function. This is needed to help:
  - * distinguishing between IPv4-converted IPv6 addresses and native IPv6 addresses.
  - * implementing IPv6 address synthesis for applications not relying on DNS.
- o Avoid stale Pref64::/n.
- o Discover multiple Pref64::/n when multiple prefixes in a network.
- o Use DNSSEC in the presence of NAT64.

Section 3.2 lists some applications which encounter the issues listed above.

3.2. Use Cases

3.2.1. AAAA Synthesis by Stub-resolver

The extension defined in this document can be used for hosts with DNS64 capability [RFC6147], added to the host's stub-resolver.

The stub resolver on the host will try to obtain (native) AAAA records and if it they are not found, the DNS64 function on the host will query for A records and then synthesizes AAAA records. Using the PREFIX64 PCP extension, the host's stub-resolver can learn the prefix used for IPv6/IPv4 translator and synthesize AAAA records accordingly.

Learning the Pref64::/n used to construct IPv4-converted IPv6 addresses [RFC6052] allows to make use of DNSSEC.

3.2.2. Applications Referrals

As discussed in [I-D.carpenter-behave-referral-object], a frequently occurring situation is that one entity A connected to the Internet (or to some private network) needs to inform another entity B how to reach either A itself or some third-party entity C. This is known as address referral.

In the particular context of NAT64 [RFC6146], applications relying on address referral will fail because an IPv6-only client won't be able to make use of an IPv4 address received in a referral. A non-exhaustive list of applications is provided below:

- BitTorrent is a distributed file sharing infrastructure which is based on P2P techniques for exchanging files between connected users. In order to download a given file, a BitTorrent client needs to obtain the corresponding torrent file. Then, it connects to the tracker to retrieve a list of lechers (clients which are currently downloading the file but do not yet detain all the portions of the file) and seeders (clients which detain all the portions of the file and are uploading them to other requesting clients). The client connects to those machines and downloads the available portions of the requested file. In the presence of an address sharing function, some encountered issues are solved if PCP is enabled (see [I-D.boucadair-pcp-bittorrent]). Nevertheless, an IPv6-only client can not connect to a remote IPv4-only machine even if base PCP is enabled. An extension is needed to solve this concern.
- In SIP environments [RFC3261], the SDP part of exchanged SIP messages includes required information for establishment of RTP sessions (particularly IP address and port number). When a NAT64 is involved in the path, an IPv6-only SIP UA (User Agent) which receives an SDP offer/answer containing an IPv4 address, cannot send media streams to the remote endpoint.
- An IPv6-only WebRTC (Web Real-Time communication, [I-D.ietf-rtcweb-overview]) can not make use of an IPv4 address received in referrals to establish a successful session with a remote IPv4-only WebRTC agent.

4. PREFIX64 Option

4.1. Format

The format of the PREFIX64 option is depicted in Figure 1. This option follows the guidelines specified in Section 7.3 of [RFC6887].

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Option Code  |  Reserved     |   Option Length               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Prefix64 (Variable)                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 1: Prefix64 PCP Option

The description of the fields is as follows:

- Option Code: To be assigned by IANA.
- Option Length: Indicates in octets the length of the Pref64::/n. Allowed values are 4, 5, 6, 7, 8, or 12 [RFC6052].

o Prefix64: This field identifies the IPv6 unicast prefix to be used for constructing an IPv4-embedded IPv6 address from an IPv4 address. The address synthesize MUST follow the guidelines documented in [RFC6052].

Option Name: PREFIX64
Number: <to be assigned in the optional-to-process range>
Purpose: Learn the prefix used by the NAT64 to build IPv4-embedded IPv6 addresses. This is used by a host for local address synthesis (e.g., when IPv4 address is present in referrals).
Valid for Opcodes: MAP, ANNOUNCE
Length: Variable
May appear in: request, response.
Maximum occurrences: 1 for a request. As many as fit within maximum PCP message size for a response.

4.2. Behavior

The PCP client includes a PREFIX64 option in a MAP or ANNOUNCE request to learn the IPv6 prefix used by an upstream PCP-controlled NAT64 device. When enclosed in a PCP request, Prefix64 MUST be set to ::/96. The PREFIX64 option can be inserted in a MAP request used to learn the external IP address as detailed in Section 11.6 of [RFC6887].

The PCP server controlling a NAT64 SHOULD be configured to return to requesting PCP clients the value of the Pref64::/n used to build IPv4-embedded IPv6 addresses. When enabled, the PREFIX64 option conveys the value of Pref64::/n.

The PCP server controlling a NAT64 MAY be configured to include a PREFIX64 option in all MAP responses even if the PREFIX64 option is not listed in the associated request. The PCP server controlling a NAT64 MAY be configured to include a PREFIX64 option in its ANNOUNCE messages.

When multiple prefixes are configured in a network, the PCP server MAY be configured to return multiple PREFIX64 options in the same message to the PCP client. The PCP server includes in the first PREFIX64 option, which appears in the PCP message it sends to the PCP client, the prefix to perform local IPv6 address synthesis [RFC6052]. Remaining PREFIX64 options convey other Pref64::/n configured in the network. Returning these prefixes allows an end host to avoid any NAT64 deployed in the network.

The host embedding the PCP client uses the prefix included in the first PREFIX64 option for local address synthesize. Remaining prefixes are used by the host to avoid any NAT64 deployed in the network. How the content of the PREFIX64 option(s) is passed to the OS is implementation-specific.

The PCP client MUST be prepared to receive multiple Pref64::/n (e.g., if several PCP servers are deployed; each of them is configured with a distinct Pref64::/n). The PCP client SHOULD associate each received Pref64::/n with the PCP server from which the Pref64::/n information was retrieved. If the PCP client fails to contact a given PCP server, the PCP client SHOULD clear the prefix(es) it learned from that PCP server.

If a distinct Pref64::/n is configured to the PCP-controlled NAT64 device, the PCP server SHOULD issue an unsolicited PCP message to inform the PCP client about the new Pref64::/n. Upon receipt of this message, the PCP client replaces the old prefix received from the same PCP server with the new Pref64::/n included in the PREFIX64 option.

5. Flow Examples

This section provides a non-normative description of use cases relying on the PREFIX64 option.

5.1. TCP Session Initiated from an IPv6-only Host

The usage shown in Figure 2 depicts a typical usage of the PREFIX64 option when a DNS64 capability is embedded in the host.

In the example shown in Figure 2, once the IPv6-only client discovered the IPv4 address of the remote IPv4-only server, it retrieves the Pref64::/n (i.e., 2001:db8:122:300::/56) to be used to build an IPv4-embedded IPv6 address for that server. This is achieved using the PREFIX64 option (Steps (a) and (b)). The client uses 2001:db8:122:300::/56 to construct an IPv6 address and then initiates a TCP connection (Steps (1) to (4)).

```
      +---------+              +-----+              +---------+
      |IPv6-only|              |NAT64|              |IPv4-only|
      | Client  |              |     |              |  Server |
      +---------+              +-----+              +---------+
           |                      |                      |
           | (a) PCP MAP Request  |                      |
           |      PREFIX64        |                      |
           |=====================>|                      |
           | (b) PCP MAP Response |                      |
```

```
             |      PREFIX64 =       |                      |
             | 2001:db8:122:300::/56 |                      |
             |<======================|                      |
             |    (1) TCP SYN        |    (2) TCP SYN       |
             |======================>|=====================>|
             |   (4) TCP SYN/ACK     |   (3) TCP SYN/ACK    |
             |<======================|<=====================|
             |    (5) TCP ACK        |    (6) TCP ACK       |
             |======================>|=====================>|
             |                       |                      |
```

Figure 2: Example of TCP session initiated from an IPv6-only host

5.2. SIP Flow Example

Figure 3 shows an example of the use of the option defined in Section 4 in a SIP context. In order for RTP/RTCP flows to be exchanged between an IPv6-only SIP UA and an IPv4-only UA without requiring any ALG (Application Level Gateway) at the NAT64 nor any particular function at the IPv4-only SIP Proxy Server (e.g., Hosted NAT traversal), the PORT_SET option [I-D.ietf-pcp-port-set] is used in addition to the PREFIX64 option.

In Steps (a) and (b), the IPv6-only SIP UA retrieves a pair of ports to be used for RTP/RTCP sessions, the external IPv4 address and the Pref64::/n to build IPv4-embedded IPv6 addresses. This is achieved by issuing a MAP request which includes a PREFIX64 option and a PORT_SET option. A pair of ports (i.e., port_X/port_X+1) and an external IPv4 address are then returned by the PCP server to the requesting PCP client together with a Pref64::/n (i.e., 2001:db8:122::/48).

The returned external IPv4 address and external port numbers are used by the IPv6-only SIP UA to build its SDP offer which contains exclusively IPv4 addresses (especially in the "c=" line, the port indicated for media port is the external port assigned by the PCP server). The INVITE request including the SDP offer is then forwarded by the NAT64 to the Proxy Server which will relay it to the called party (i.e., IPv4-only SIP UA) (Steps (1) to (3)).

The remote IPv4-only SIP UA accepts the offer and sends back its SDP answer in a "200 OK" message which is relayed by the SIP Proxy Server and NAT64 until being delivered to IPv6-only SIP UA (Steps (4) to (6)).

Pref64::/n (2001:db8:122::/48) is used by the IPv6-only SIP UA to construct a corresponding IPv6 address of the IPv4 address enclosed in the SDP answer made by the IPv4-only SIP UA (Step 6).

IPv6-only SIP UA and IPv4-only SIP UA are then able to exchange RTP/RTCP flows without requiring any ALG at the NAT64 nor any particular function at the IPv4-only SIP Proxy Server.

```
   +---------+              +-----+       +------------+     +---------+
   |IPv6-only|              |NAT64|       |  IPv4 SIP  |     |IPv4-only|
   | SIP UA  |              |     |       |Proxy Server|     | SIP UA  |
   +---------+              +-----+       +------------+     +---------+
       | (a) PCP MAP Request   |                |                 |
       |        PORT_SET       |                |                 |
       |        PREFIX64       |                |                 |
       |======================>|                |                 |
       | (b) PCP MAP Response  |                |                 |
       |        PORT_SET       |                |                 |
       |        PREFIX64:      |                |                 |
       |     2001:db8:122::/48 |                |                 |
       |<======================|                |                 |
       |  (1) SIP INVITE       | (2) SIP INVITE |  (3) SIP INVITE |
       |======================>|===============>|================>|
       |   (6) SIP 200 OK      | (5) SIP 200 OK |  (4) SIP 200 OK |
       |<======================|<===============|<================|
       |     (7) SIP ACK       |  (8) SIP ACK   |    (9) SIP ACK  |
       |======================>|===============>|================>|
       |                       |                |                 |
       |src port:     dst port:|src port:                dst port:|
       |port_A           port_B|port_X                      port_B|
       |<======IPv6 RTP=======>|<============IPv4 RTP============>|
       |<===== IPv6 RTCP======>|<============IPv4 RTCP===========>|
       |src port:     dst port:|src port:                dst port:|
       |port_A+1       port_B+1|port_X+1                  port_B+1|
       |                       |                                  |
```

Figure 3: Example of IPv6 to IPv4 SIP initiated Session

When the session is initiated from the IPv4-only SIP UA (see Figure 4), the IPv6-only SIP UA retrieves a pair of ports to be used for RTP/RTCP session, the external IPv4 address and the Pref64::/n to build IPv4-embedded IPv6 addresses (Steps (a) and (b)). These two steps can be delayed until receiving the INVITE message (Step 3).

The retrieved IPv4 address and port numbers are used to build the SDP answer in Step (4) while Pref64::/n is used to construct a corresponding IPv6 address of the IPv4 address enclosed in the SDP offer made by the IPv4-only SIP UA (Step 3). RTP/RTCP flows are

exchanged between an IPv6-only SIP UA and an IPv4-only UA without requiring any ALG at the NAT64 nor any function at the IPv4-only SIP Proxy Server.

```
+---------+             +-----+       +------------+     +---------+
|IPv6-only|             |NAT64|       |  IPv4 SIP  |     |IPv4-only|
| SIP UA  |             |     |       |Proxy Server|     | SIP UA  |
+---------+             +-----+       +------------+     +---------+
    | (a) PCP MAP Request   |                |                 |
    |        PORT_SET       |                |                 |
    |        PREFIX64       |                |                 |
    |======================>|                |                 |
    | (b) PCP MAP Response  |                |                 |
    |        PORT_SET       |                |                 |
    |        PREFIX64:      |                |                 |
    |     2001:db8:122::/48 |                |                 |
    |<======================|                |                 |
    |  (3) SIP INVITE       | (2) SIP INVITE |  (1) SIP INVITE |
    |<======================|<===============|<================|
    |   (4) SIP 200 OK      | (5) SIP 200 OK |  (6) SIP 200 OK |
    |======================>|===============>|================>|
    |     (9) SIP ACK       |  (8) SIP ACK   |    (7) SIP ACK  |
    |<======================|<===============|<================|
    |                       |                |                 |
    |src port:     dst port:|src port:                dst port:|
    |port_a           port_b|port_Y                      port_b|
    |<======IPv6 RTP=======>|<============IPv4 RTP============>|
    |<===== IPv6 RTCP======>|<============IPv4 RTCP===========>|
    |src port:     dst port:|src port:                dst port:|
    |port_a+1       port_b+1|port_Y+1                  port_b+1|
    |                       |                                  |
```

Figure 4: Example of IPv4 to IPv6 SIP initiated Session

6. IANA Considerations

The following PCP Option Code is to be allocated in the optional-to-process range (the registry is maintained in http://www.iana.org/assignments/pcp-parameters/pcp-parameters.xml#option-rules):

PREFIX64

7. Security Considerations

PCP-related security considerations are discussed in [RFC6887].

As discussed in [RFC6147], an attacker can manage to change the Pref64::/n used by the DNS64, the traffic generated by the host that

receives the synthetic reply will be delivered to the altered Pref64. This can result in either a denial- of-service (DoS) attack, a flooding attack, or an eavesdropping attack. This attack can be achieved by altering PCP messages issued by a legitimate PCP server or a fake PCP server is used.

Means to defend against attackers who can modify between the PCP server and the PCP client, or who can inject spoofed packets that appear to come from a legitimate PCP server SHOULD be enabled. For example, access control lists (ACLs) can be installed on the PCP client, PCP server, and the network between them, so those ACLs allow only communications from a trusted PCP server to the PCP client.

PCP server discovery is out of scope of this document. It is the responsibility of PCP server discovery document(s) to elaborate on the security considerations to discover a legitimate PCP server.

8. Acknowledgements

Many thanks to S. Perreault , R. Tirumaleswar, T. Tsou, D. Wing, J. Zhao, R. Penno and I. Van Beijnum for the comments and suggestions.

9. References

9.1. Normative References

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

[RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, June 2002.

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

[RFC6146] Bagnulo, M., Matthews, P., and I. van Beijnum, "Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers", RFC 6146, April 2011.

[RFC6147] Bagnulo, M., Sullivan, A., Matthews, P., and I. van Beijnum, "DNS64: DNS Extensions for Network Address Translation from IPv6 Clients to IPv4 Servers", RFC 6147, April 2011.

[RFC6887] Wing, D., Cheshire, S., Boucadair, M., Penno, R., and P. Selkirk, "Port Control Protocol (PCP)", RFC 6887, April 2013.

9.2. Informative References

[I-D.boucadair-pcp-bittorrent]
Boucadair, M., Zheng, T., Deng, X., and J. Queiroz, "Behavior of BitTorrent service in PCP-enabled networks with Address Sharing", draft-boucadair-pcp-bittorrent-00 (work in progress), May 2012.

[I-D.boucadair-pcp-nat64-experiments]
Abdesselam, M., Boucadair, M., Hasnaoui, A., and J. Queiroz, "PCP NAT64 Experiments", draft-boucadair-pcp-nat64-experiments-00 (work in progress), September 2012.

[I-D.carpenter-behave-referral-object]
Carpenter, B., Boucadair, M., Halpern, J., Jiang, S., and K. Moore, "A Generic Referral Object for Internet Entities", draft-carpenter-behave-referral-object-01 (work in progress), October 2009.

[I-D.ietf-pcp-port-set]
Sun, Q., Boucadair, M., Sivakumar, S., Zhou, C., Tsou, T., and S. Perreault, "Port Control Protocol (PCP) Extension for Port Set Allocation", draft-ietf-pcp-port-set-01 (work in progress), May 2013.

[I-D.ietf-rtcweb-overview]
Alvestrand, H., "Overview: Real Time Protocols for Brower-based Applications", draft-ietf-rtcweb-overview-06 (work in progress), February 2013.

[I-D.zhang-behave-nat64-load-balancing]
Zhang, D., Xu, X., and M. Boucadair, "Considerations on NAT64 Load-Balancing", draft-zhang-behave-nat64-load-balancing-03 (work in progress), July 2011.

Author's Address

Mohamed Boucadair
France Telecom
Rennes 35000
France

Email: mohamed.boucadair@orange.com