Internet Engineering Task Force (IETF)
Request for Comments: 7225
Category: Standards Track
ISSN: 2070-1721

M. Boucadair
France Telecom

May 2014

Discovering NAT64 IPv6 Prefixes Using the Port Control Protocol (PCP)

Abstract

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

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 5741.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at http://www.rfc-editor.org/info/rfc7225.

Table of Contents

1. Introduction

According to [RFC6146], NAT64 uses Pref64::/n to construct IPv4-converted IPv6 addresses as defined in [RFC6052].

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

This PCP option is a deterministic solution to help establish communications between IPv6-only hosts and remote IPv4-only hosts. Unlike [RFC7050], this option solves all the issues identified in [RFC7051].

Some illustrative examples are provided in Section 5. Detailed experiments conducted to assess the applicability of the PREFIX64 option for services (e.g., accessing a video server, establishing SIP-based sessions, etc.) in NAT64 environments are available in [EXPERIMENTS].

The use of this PCP option for NAT64 load-balancing purposes is out of scope.

2. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "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:

- Learn the Pref64::/n used by an upstream NAT64 function. This is needed to help:
  - distinguish between IPv4-converted IPv6 addresses [RFC6052] and native IPv6 addresses.
  - implement IPv6 address synthesis for applications not relying on DNS (where DNS64 [RFC6147] would provide the synthesis).
- Avoid stale Pref64::/n values.
- Discover multiple Pref64::/n values when multiple prefixes exist in a network.
- Use DNSSEC ([RFC4033], [RFC4034], [RFC4035]) in the presence of NAT64.
- Discover the suffix used by a NAT64 function when non-null suffixes are in use (e.g., checksum neutral suffix).
- Support destination-based Pref64::/n (e.g., Section 5.1 of [RFC7050]).
- Associate a Pref64::/n with a given NAT64 when distinct prefixes are configured for each NAT64 enabled in a network.

A more extensive discussion can be found at [RFC7051].

3.2. Use Cases

This section provides some use cases to illustrate the problem space. More details can be found at Section 4 of [RFC7051].

3.2.1. AAAA Synthesis by the DNS Stub-Resolver

The option 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 they are not found, the DNS64 function on the host will query for A records and then synthesize AAAA records. Using the PREFIX64 PCP extension, the host's stub-resolver can learn the prefix used for IPv6/IPv4 translation and synthesize AAAA records accordingly.

Because synthetic AAAA records cannot be successfully validated in a host, learning the Pref64::/n used to construct IPv4-converted IPv6 addresses allows the use of DNSSEC. As discussed in Section 5.5 of [RFC6147], a security-aware and validating host has to perform the DNS64 function locally.

3.2.2. Application Referrals

As discussed in [REF-OBJECT], a frequently occurring situation is that one entity A connected to a 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 such applications is provided below:

- In SIP environments [RFC3261], the SDP part ([RFC4566]) of exchanged SIP messages includes information required for establishing RTP sessions (namely, IP address and port number). When a NAT64 is involved in the path, an IPv6-only SIP User Agent (UA) that 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 [WebRTC]) agent cannot make use of an IPv4 address received in referrals to establish a successful session with a remote IPv4-only WebRTC agent.

- BitTorrent is a distributed file-sharing infrastructure that is based on peer-to-peer (P2P) techniques for exchanging files between connected users. To download a given file, a BitTorrent client needs to obtain the corresponding torrent file. Then, it connects to a tracker to retrieve a list of "leechers" (clients that are currently downloading the file but do not yet possess all

portions of the file) and "seeders" (clients that possess all 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 (see Appendix A of [RFC6269]), some encountered issues are solved if PCP is enabled (see [PCP-BITTORRENT]). Nevertheless, an IPv6-only client cannot connect to a remote IPv4-only machine even if the base PCP protocol is used.

Learning the Pref64::/n solves the issues listed above.

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].

This option allows the mapping of specific IPv4 address ranges (contained in the IPv4 Prefix List) to separate Pref64::/n prefixes as discussed in [RFC6147].

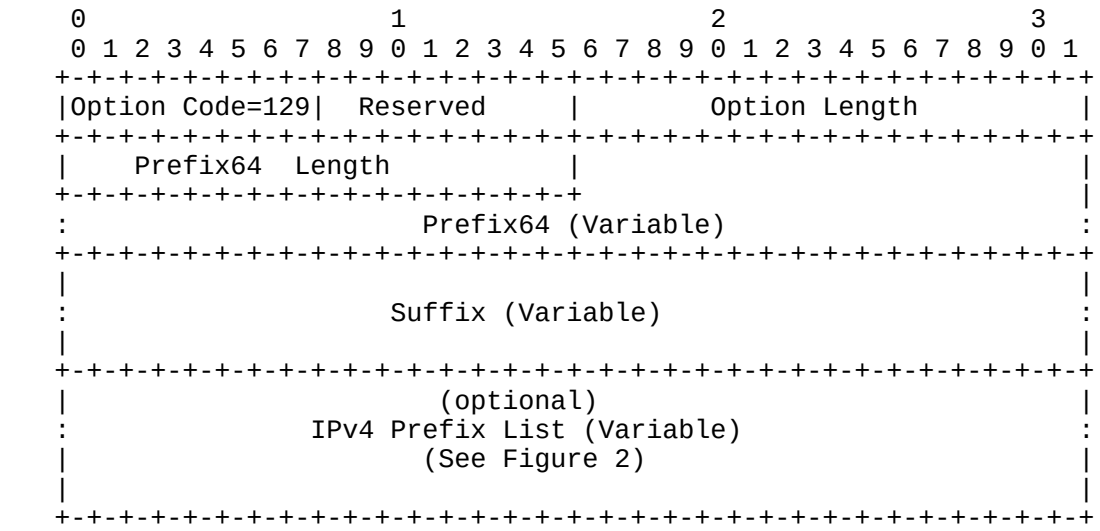


Figure 1: Prefix64 PCP Option

The description of the fields is as follows:

- Option Code: 129
- Reserved: This field is initialized as specified in Section 7.3 of [RFC6887].
- Option Length: Indicates in octets the length of the enclosed data.
- Prefix64 Length: Indicates in octets the length of the Pref64::/n. The allowed values are specified in [RFC6052] (i.e., 4, 5, 6, 7, 8, or 12).
- Prefix64: This field identifies the IPv6 unicast prefix to be used for constructing an IPv4-converted IPv6 address from an IPv4 address as specified in Section 2.2 of [RFC6052]. This prefix can be the Well-Known Prefix (i.e., 64:ff9b::/96) or a Network-Specific Prefix. The address synthesis MUST follow the guidelines documented in [RFC6052].
- Suffix: The length of this field is (12 - Prefix64 Length) octets. This field identifies the suffix to be used for constructing an IPv4-converted IPv6 address from an IPv4 address as specified in Section 2.2 of [RFC6052]. No suffix is included if a /96 Prefix64 is conveyed in the option.
- IPv4 Prefix List: This is an optional field. The format of the IPv4 Prefix List field is shown in Figure 2. This field may be included by a PCP server to solve the destination-dependent Pref64::/n discovery problem discussed in Section 5.1 of [RFC7050].
  * IPv4 Prefix Count: indicates the number of IPv4 prefixes included in the option. "IPv4 Prefix Count" field MUST be set to 0 in a request and MUST be set to the number of included IPv4 subnets in a response.
  * An IPv4 prefix is represented as "IPv4 Address/IPv4 Prefix Length" [RFC4632]. For example, to encode 192.0.2.0/24, "IPv4 Prefix Length" field is set to 24 and "IPv4 Address" field is set to 192.0.2.0. If a Pref64::/n is configured for all IPv4 addresses, a wildcard IPv4 prefix (i.e., 0.0.0.0/0) may be returned in the response together with the configured Pref64::/n. If no IPv4 Prefix List is returned in a PREFIX64 option, the PCP client assumes the prefix is valid for any destination IPv4 address. Valid IPv4 prefixes are listed in Section 3.1 of [RFC4632].

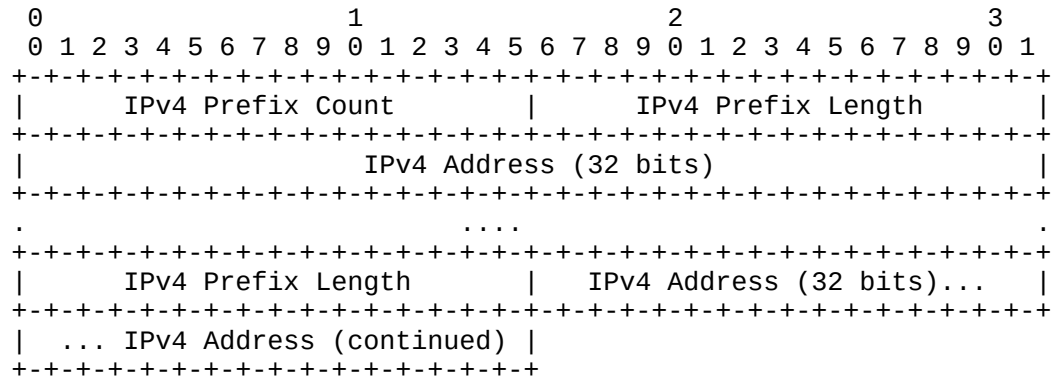


Figure 2: Format of IPv4 Prefix List field

Option Name: PREFIX64

Value: 129

Purpose: Learn the prefix used by the NAT64 to build IPv4-converted IPv6 addresses. This is used by a host for local address synthesis (e.g., when an 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 the maximum PCP message size for a response.

4.2. Server's Behavior

The PCP server controlling a NAT64 SHOULD be configured to return to requesting PCP clients the value of the Pref64::/n and suffix used to build IPv4-converted IPv6 addresses. When enabled, the PREFIX64 option conveys the value of the Pref64::/n and configured suffix. If no suffix is explicitly configured to the PCP server, the null suffix is used as the default value (see Section 2.2 of [RFC6052]).

If the PCP server is configured to honor the PREFIX64 option but no Pref64::/n is explicitly configured, the PCP server MUST NOT include any PREFIX64 option in its PCP messages.

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.

The PCP server MAY be configured with a list of destination IPv4 prefixes associated with a Pref64::/n. This list is then included by the PCP server in a PREFIX64 option sent to PCP clients.

The PCP server MAY be configured to return multiple PREFIX64 options in the same message to the PCP client. In such case, the server does the following:

- If no destination IPv4 prefix list is configured, the PCP server includes in the first PREFIX64 option, which appears in the PCP message it sends to the PCP client, the prefix and suffix to perform local IPv6 address synthesis [RFC6052]. Additional PREFIX64 options convey any other Pref64::/n values configured. Returning these prefixes allows an end host to identify all synthesized IPv6 addresses in a network; the host can prefer IPv4 or another network interface instead in order to avoid any NAT64 deployed in the network. The PCP server is required to disambiguate prefixes used for IPv6 address synthesis and other prefixes used to avoid any NAT64 deployed in the network. The PCP server can be configured with a customized IPv6 prefix list (i.e., specific to a PCP client or a group of PCP clients) or system-wide IPv6 prefix list (i.e., the same list is returned for any PCP client). Note, it is NOT RECOMMENDED to include PREFIX64 options in ANNOUNCE messages if a customized IPv6 prefix list is configured to the PCP server.

- If IPv4 prefix lists are configured, the PCP server includes in the first PREFIX64 options the Pref64::/n and suffix that are associated with an IPv4 prefix list (i.e., each of these PREFIX64 options conveys a distinct Pref64::/n together with an IPv4 prefix list). Additional PREFIX64 options convey any other Pref64::/n values configured (i.e., the remaining Pref64::/n values not mapped to any IPv4 prefix list).

If a distinct Pref64::/n or suffix is configured to the PCP-controlled NAT64 device, the PCP server SHOULD issue an unsolicited PCP ANNOUNCE message to inform the PCP client about the new Pref64::/n and/or suffix.

4.3. Client's Behavior

The PCP client includes a PREFIX64 option in a MAP or ANNOUNCE request to learn the IPv6 prefix and suffix used by an upstream PCP-controlled NAT64 device. When enclosed in a PCP request, the 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 client MUST be prepared to receive multiple prefixes (e.g., if several PCP servers are deployed and each of them is configured with a distinct Pref64::/n). The PCP client MUST associate each received Pref64::/n and suffix with the PCP server from which the Pref64::/n and suffix information was retrieved.

If the PCP client fails to contact a given PCP server, the PCP client SHOULD clear the prefix(es) and suffix(es) it learned from that PCP server. For example, a PCP client may fail to contact a PCP server if the host embedding the PCP client moves to a new network or if that PCP server is out of service. The use of these stale prefixes is not recommended to build an IPv4-converted IPv6 address because failures are likely to be encountered (see [RFC7051], Section 3, Issue #4).

If the PCP client receives a PREFIX64 option that includes an invalid IPv4 prefix, the PCP client ignores that IPv4 prefix. If one or more valid IPv4 prefixes and/or IPv6 prefixes and suffixes are present, the PCP client uses them.

Upon receipt of the message from the PCP server, the PCP client replaces any old prefix(es)/suffix(es) received from the same PCP server with the new one(s) included in the PREFIX64 option(s). If no PREFIX64 option includes a destination IPv4 prefix list, the host embedding the PCP client uses the prefix/suffix included in the first PREFIX64 option for local address synthesis. Other prefixes learned can be used by the host to avoid any NAT64 deployed in the network. If one or multiple received PREFIX64 options contain a destination IPv4 prefix list, the PCP client MUST associate the included IPv4 prefixes with the Pref64::/n and the suffix indicated in the same PREFIX64 option. In such case, the host embedding the PCP client MUST enforce a destination-based prefix Pref64::/n selection for local address synthesis purposes. How the content of the PREFIX64 option(s) is passed to the OS is implementation specific.

Upon receipt of an unsolicited PCP ANNOUNCE message, the PCP client replaces the old prefix/suffix received from the same PCP server with the new Pref64::/n and suffix included in the PREFIX64 option.

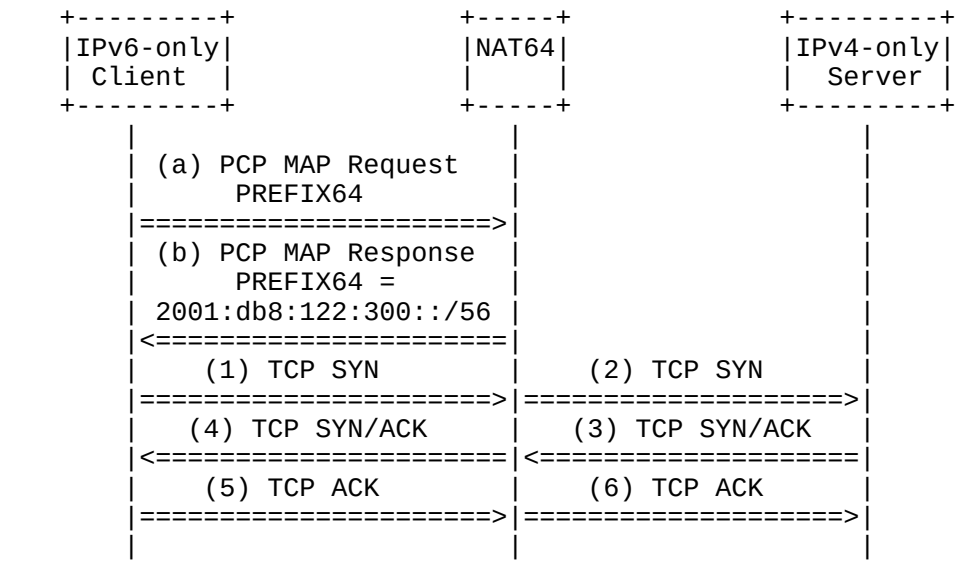
5. Flow Examples

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 3 depicts a typical usage of the PREFIX64 option when a DNS64 capability is embedded in the host.

In the example shown in Figure 3, once the IPv6-only client discovers the IPv4 address of the remote IPv4-only server (e.g., using DNS), it retrieves the Pref64::/n (i.e., 2001:db8:122:300::/56) to be used to build an IPv4-converted IPv6 address for that server. This retrieval is achieved using the PREFIX64 option (Steps (a) and (b)). The client then uses 2001:db8:122:300::/56 to construct an IPv6 address and then initiates a TCP connection (Steps (1) to (4)).



Note: The DNS exchange to retrieve the IPv4 address of the IPv4-only Server is not shown in the figure.

Figure 3: Example of a TCP Session Initiated from an IPv6-Only Host

5.2. SIP Flow Example

Figure 4 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 or any particular function at the IPv4-only SIP Proxy Server (e.g., hosted NAT traversal [LATCHING]), the PORT_SET option [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 that 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 (together with a Pref64::/n, i.e., 2001:db8:122::/48) are then returned by the PCP server to the requesting PCP client.

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 the 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., the 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 that is relayed by the SIP Proxy Server and NAT64 until being delivered to the IPv6-only SIP UA (Steps (4) to (6)).

The 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)).

The IPv6-only SIP UA and IPv4-only SIP UA are then able to exchange RTP/RTCP flows without requiring any ALG at the NAT64 or any special function at the IPv4-only SIP Proxy Server.

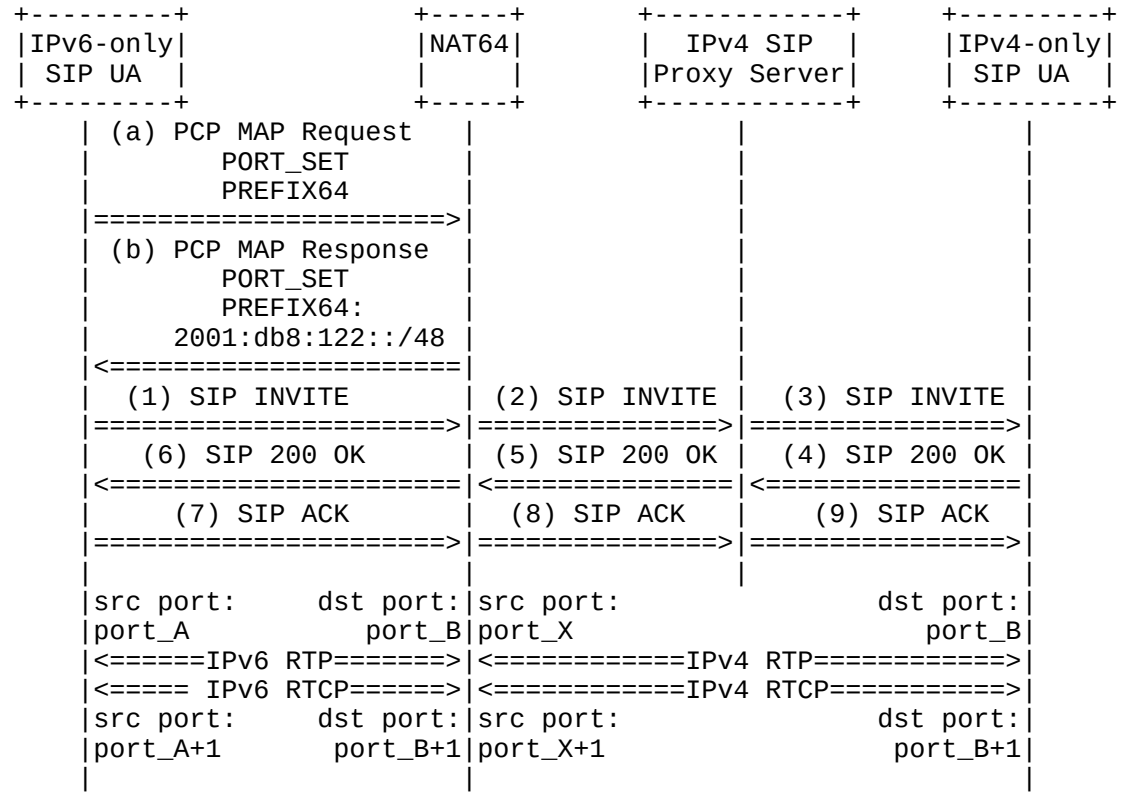


Figure 4: Example of IPv6 to IPv4 SIP-Initiated Session

When the session is initiated from the IPv4-only SIP UA (see Figure 5), the IPv6-only SIP UA retrieves a pair of ports to be used for the RTP/RTCP session, the external IPv4 address and the Pref64::/n to build IPv4-converted IPv6 addresses (Steps (a) and (b)). These two steps could instead be delayed until the INVITE message is received (Step (3)).

The retrieved IPv4 address and port numbers are used to build the SDP answer in Step (4), while the Pref64::/n is used to construct an IPv6 address corresponding to the IPv4 address enclosed in the SDP offer made by the IPv4-only SIP UA (Step (3)). RTP/RTCP flows are then exchanged between the IPv6-only SIP UA and the IPv4-only UA without requiring any ALG at the NAT64 or any special function at the IPv4-only SIP Proxy Server.

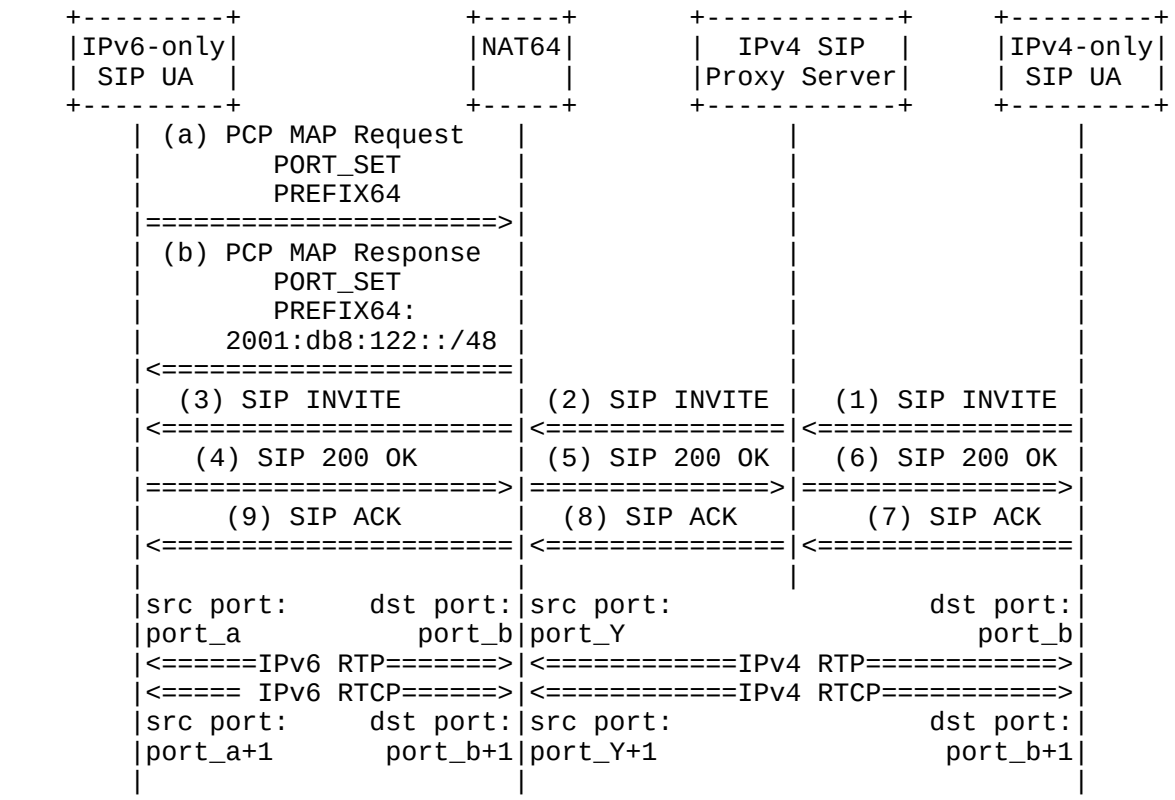


Figure 5: Example of IPv4 to IPv6 SIP-Initiated Session

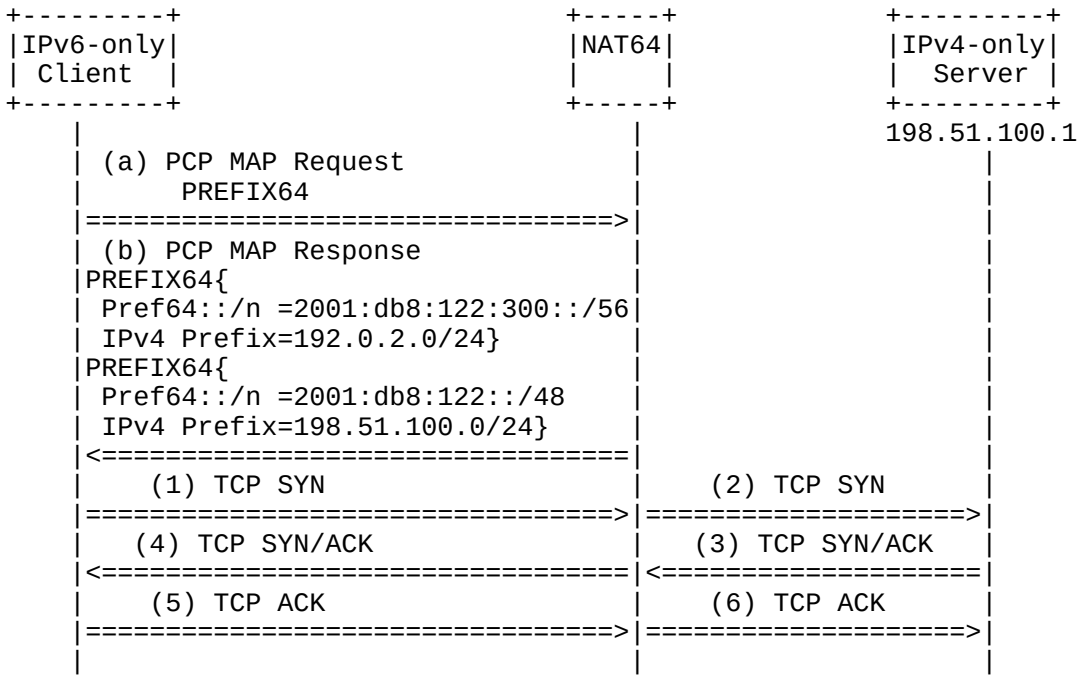
5.3. Mapping of IPv4 Address Ranges to IPv6 Prefixes

Figure 6 shows an example of a NAT64 configured with two Pref64::/n values; each of these Pref64::/n values is associated with a distinct IPv4 address range:

- 192.0.2.0/24 is mapped to 2001:db8:122:300::/56.
- 198.51.100.0/24 is mapped to 2001:db8:122::/48.

Once the IPv6-only client discovers the IPv4 address of the remote IPv4-only server (i.e., 198.51.100.1), it retrieves two IPv6 prefixes to be used to build an IPv4-converted IPv6 addresses. This retrieval is achieved using two PREFIX64 options (Step (b)).

Because 198.51.100.1 matches the destination prefix 198.51.100.0/24, the client uses the associated Pref64::/n (i.e., 2001:db8:122::/48) to construct an IPv6 address for that IPv4-only server, and then it initiates a TCP connection (Steps (1) to (6)).



Note: The DNS exchange to retrieve the IPv4 address of the IPv4-only Server is not shown in the figure.

Figure 6: Mapping of IPv4 Address Ranges to IPv6 Prefixes

A similar behavior is to be experienced if these Pref64::/n values and associated IPv4 prefix lists are configured to distinct NAT64 devices.

6. IANA Considerations

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

PREFIX64 set to 129 (see Section 4.1)

7. Security Considerations

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

As discussed in [RFC6147], if an attacker can manage to change the Pref64::/n used by the DNS64 function, 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 a man-in-the-middle (MITM) attack. This attack could be achieved either by altering PCP messages issued by a legitimate PCP server or by using a fake PCP server.

Means to defend against attackers who can modify packets 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. In some deployments, 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 for this document. It is the responsibility of documents about PCP server discovery to elaborate on the security considerations to discover a legitimate PCP server.

Learning a Pref64::/n via PCP allows using DNSSEC in the presence of NAT64. As such, NAT64 with DNSSEC and PCP is better than no DNSSEC at all, but it is less safe than DNSSEC without DNS64/NAT64 and PCP. The best mitigation action against Pref64::/n discovery attacks is thus to add IPv6 support in all endpoints and hence reduce the need to perform IPv6 address synthesis.

8. Acknowledgements

Many thanks to S. Perreault, R. Tirumaleswar, T. Tsou, D. Wing, J. Zhao, R. Penno, I. van Beijnum, T. Savolainen, S. Savikumar, D. Thaler, T. Lemon, S. Hanna, P. Resnick, R. Sparks, S. Farrell, and W. Cui for their 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.

[RFC4632] Fuller, V. and T. Li, "Classless Inter-domain Routing (CIDR): The Internet Address Assignment and Aggregation Plan", BCP 122, RFC 4632, August 2006.

[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

[PCP-BITTORRENT]
Boucadair, M., Zheng, T., Deng, X., and J. Queiroz, "Behavior of BitTorrent service in PCP-enabled networks with Address Sharing", Work in Progress, May 2012.

[EXPERIMENTS]
Abdesselam, M., Boucadair, M., Hasnaoui, A., and J. Queiroz, "PCP NAT64 Experiments", Work in Progress, September 2012.

[REF-OBJECT]
Carpenter, B., Jiang, S., and Z. Cao, "Problem Statement for Referral", Work in Progress, February 2011.

[LATCHING] Ivov, E., Kaplan, H., and D. Wing, "Latching: Hosted NAT Traversal (HNT) for Media in Real-Time Communication", Work in Progress, May 2014.

[PORT-SET] Qiong, Q., Boucadair, M., Sivakumar, S., Zhou, C., Tsou, T., and S. Perreault, "Port Control Protocol (PCP) Extension for Port Set Allocation", Work in Progress, November 2013.

[WebRTC] Alvestrand, H., "Overview: Real Time Protocols for Brower-based Applications", Work in Progress, February 2014.

[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.

[RFC4033] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "DNS Security Introduction and Requirements", RFC 4033, March 2005.

[RFC4034] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Resource Records for the DNS Security Extensions", RFC 4034, March 2005.

[RFC4035] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Protocol Modifications for the DNS Security Extensions", RFC 4035, March 2005.

[RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", RFC 4566, July 2006.

[RFC6269] Ford, M., Boucadair, M., Durand, A., Levis, P., and P. Roberts, "Issues with IP Address Sharing", RFC 6269, June 2011.

[RFC7050] Savolainen, T., Korhonen, J., and D. Wing, "Discovery of the IPv6 Prefix Used for IPv6 Address Synthesis", RFC 7050, November 2013.

[RFC7051] Korhonen, J. and T. Savolainen, "Analysis of Solution Proposals for Hosts to Learn NAT64 Prefix", RFC 7051, November 2013.

Author’s Address

Mohamed Boucadair
France Telecom
Rennes 35000
France

EMail: mohamed.boucadair@orange.com