PCP Working Group
Internet-Draft
Intended status: Standards Track
Expires: May 29, 2014

M. Boucadair
France Telecom
T. Reddy
Cisco

November 25, 2013

Retrieving the Capabilities of a PCP-controlled Device
draft-boucadair-pcp-capability-03

Abstract

This document extends Port Control Protocol (PCP) with the ability to retrieve the capabilities of PCP-controlled device: CAPABILITY Option.

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 RFC 2119 [RFC2119].

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 May 29, 2014.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents

carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction 2
2. CAPABILITY 2
3. PCP Client/Server Behavior 3
4. Option Usage 4
5. Security Considerations 5
6. IANA Considerations 5
7. References 6
7.1. Normative References 6
7.2. Informative References 6
Authors' Addresses 7

1. Introduction

This document extends the base PCP [RFC6887] with a new feature to discover the capabilities of a PCP-controlled device. Retrieving the capabilities of a PCP-controlled device would allow to avoid error, provide a hint why some applications fails, help select the OpCode to issue, etc.

This option can be elected to be defined as a new OpCode.

2. CAPABILITY

The CAPABILITY option (Code: TBA, Figure 1) is used by a PCP Server to indicate to a requesting PCP Client the capabilities it supports with regards to port forwarding operations.

One single Capability option is conveyed in the same PCP response message even if several functions are co-located in the same PCP-controlled device (e.g., NAT44 and NAT64, NAT44 and ports set assignment capability, etc.).

This option, when received from a PCP Server, is used by a PCP Client to constraint the content of its requests and therefore avoid errors.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| CAPABILITY    |  Reserved     |            Length=16          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
|A|                           Capability                          |
+-+                                                               |
:                                                                 :
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

This Option:

Option Name: PCP Capabilities Option (CAPABILITY)
Number: TBA (IANA)
Purpose: Retrieve the capabilities of a PCP-controlled device
Valid for Opcodes: ANNOUNCE, MAP, PEER
Length: 16
May appear in: both request and response
Maximum occurrences: None

Figure 1: Capability option

A-bit when set (i.e., 1) indicates the PCP Server supports authentication. If this bit is set to 0, is indicates plain PCP is supported.

The Capability Field is encoded in 127 bits. Each bit in the Capability bit mask is used to represent the PCP-controlled device capability. Several bits can be set if several functions are co-located in the same device. The following values for the Capability field are:

Bit #: Description
1: NAT44
2: Stateless NAT64 [RFC6145].
4: Stateful NAT64 [RFC6146].
8: A+P Port Range Router [RFC6346]
9: Supports PORT_SET option [I-D.ietf-pcp-port-set].
16: IPv4 firewall.
32: IPv6 Firewall [RFC6092].
64: NPTv6 [RFC6296].
125: DSCP re-marking function.
126: FLOWDATA-aware function ([I-D.wing-pcp-flowdata]).
127: ILNP Translator [RFC6740].

3. PCP Client/Server Behavior

This section specifies the behavior of the PCP Client and the PCP Server to handle the CAPABILITY Option.

The PCP Server MAY be configured to return the CAPABILITY Option even if it is not included in the request.

Once the PCP Client is configured with its PCP Server(s), it MAY issue an ANNOUNCE OpCode which enclose a CAPABILITY Option. Sending the ANNOUNCE OpCode and the CAPABILITY Option allows the PCP Client to determine whether the PCP Server is alive and also to retrieve its capabilities. Based on the received capabilities, the PCP Client may decide to tune its requests (e.g., Section 4) and decide whether all PCP Servers need to be contacted in parallel or only a subset of them should be contacted.

Upon receipt of a PCP request from a PCP Client requiring the PCP Server to enforce an operation beyond its capabilities, the PCP Server MAY return an error code together with the CAPABILITY option.

When a new PCP Server joins the network then it MAY send an ANNOUNCE OpCode with its capabilities (i.e., CAPABILITY Option).

4. Option Usage

Below are provided examples of the CAPABILITY Option usage:

o In an IPv6 network with NPTv6 [RFC6296], Firewalls implementing the PCP Server are on different devices: the PCP Client learns of the available PCP Servers by using DHCP [I-D.ietf-pcp-dhcp] or any other PCP Server discovery technique defined in future specifications. PCP Client learns the PCP Server capabilities using CAPABILITY Option. The PCP Client sends MAP PCP request to PCP-controlled NPTv6 device with Internal Port=0 and Protocol=0 (which means 'all ports for all protocols') to find the external IP address. This PCP request has to be sent only once since NPTv6 is stateless and provides a 1:1 relationship between addresses in the "inside" and "outside" prefixes. The PCP Client will send PCP re-request to NTPv6 only before the Assigned Lifetime of the MAP response expires or when the host embedding the PCP Client acquires a new IPv6 address using "inside" prefix. However PCP Client will send MAP/PEER requests to Firewall to create/delete dynamic outbound mapping or use PCP instead of its default application keep-alives to maintain the Firewall state alive.

```
     PCP
    Client                                  __________
+-----------+   +------+   +------+    /          \   +-----------+
|Application|___| NPTv6|___| FW   |____| Internet |___|Application|
|  Client   |   |      |   |      |    |          |   |   Server  |
+-----------+   +------+   +------+    \__________/   +-----------+
```

Figure 2: NPTv6 and FW not collocated with PCP server Capability

o In a network with NAT64 [RFC6146], Firewall implementing PCP servers are on different devices: IPv6-only PCP Client can send

PREFIX64 PCP Option [I-D.ietf-pcp-nat64-prefix64] only to the PCP-controlled NAT64 device to learn the Prefix64(s) used to build IPv4-embedded IPv6 addresses.

- o Multiple PCP-controlled devices: See Figure 3 the example of a network deploying several techniques to ensure interconnection with IPv4, provide IPv6-only connectivity, etc. Of course, one can argue this configuration is no realistic.

```
                                       +-----+
                                 ______|NPTv6|___________
                                /      +-----+           \
                                |                        |
                                |                     +-----+
   +-----------+   +------+     |                     | PRR |
   |Application|___| IPv6 |_____|     SP Network      +-----+
   |PCP  Client|   |  FW  |     |                        |
   +-----------+   +------+     |                     +------+
                                |                     | NAT64|
   +-----------+   +-------+    |                     |   +  |
   |PCP Client |___|A+P NAT|____|                     |  FW  |
   +-----------+   +-------+    |      +-----+        +------+
                                \______|NPTv6|___________/
                                       +-----+
```

Figure 3: Multiple PCP-controlled devoce

- o In a IPv6 network with ILNP translator [RFC6740], Firewall implementing PCP servers are on different devices. PCP client needs to send PCP request only to the PCP-controlled ILNP translator to find Global Locators associated with Internal Locators.
- o When the PCP-controlled device is a PRR, the PCP Client should use PORT_SET [I-D.ietf-pcp-port-set] option.

5. Security Considerations

Security considerations discussed in [RFC6887] must be considered.

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#options):

CAPABILITY

A sub-registry is required to track the set of capabilities of PCP-controlled devices.

7. References

7.1. Normative References

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

[RFC6145] Li, X., Bao, C., and F. Baker, "IP/ICMP Translation Algorithm", RFC 6145, April 2011.

[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.

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

7.2. Informative References

[I-D.ietf-opsawg-firewalls]
Baker, F. and P. Hoffman, "On Firewalls in Internet Security", draft-ietf-opsawg-firewalls-01 (work in progress), October 2012.

[I-D.ietf-pcp-dhcp]
Boucadair, M., Penno, R., and D. Wing, "DHCP Options for the Port Control Protocol (PCP)", draft-ietf-pcp-dhcp-09 (work in progress), November 2013.

[I-D.ietf-pcp-nat64-prefix64]
Boucadair, M., "Learning NAT64 PREFIX64s using PCP", draft-ietf-pcp-nat64-prefix64-04 (work in progress), July 2013.

[I-D.ietf-pcp-port-set]
Qiong, 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-04 (work in progress), November 2013.

[I-D.wing-pcp-flowdata]
Wing, D., Penno, R., and T. Reddy, "PCP Flowdata Option", draft-wing-pcp-flowdata-00 (work in progress), July 2013.

[RFC6092] Woodyatt, J., "Recommended Simple Security Capabilities in Customer Premises Equipment (CPE) for Providing Residential IPv6 Internet Service", RFC 6092, January 2011.

[RFC6296] Wasserman, M. and F. Baker, "IPv6-to-IPv6 Network Prefix Translation", RFC 6296, June 2011.

[RFC6346] Bush, R., "The Address plus Port (A+P) Approach to the IPv4 Address Shortage", RFC 6346, August 2011.

[RFC6740] Atkinson,, RJ., "Identifier-Locator Network Protocol (ILNP) Architectural Description", RFC 6740, November 2012.

Authors' Addresses

Mohamed Boucadair
France Telecom
Rennes 35000
France

Email: mohamed.boucadair@orange.com

Tirumaleswar Reddy
Cisco Systems, Inc.
Cessna Business Park, Varthur Hobli
Sarjapur Marathalli Outer Ring Road
Bangalore, Karnataka 560103
India

Email: tireddy@cisco.com

PCP Working Group
Internet-Draft
Intended status: Standards Track
Expires: March 22, 2013

M. Boucadair
France Telecom
R. Penno
D. Wing
Cisco

September 18, 2012

PCP Description Option
draft-boucadair-pcp-description-option-01

Abstract

This document extends Port Control Protocol (PCP) with the ability to associate a description with a PCP-instantiated mapping: DESCRIPTION Option.

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 RFC 2119 [RFC2119].

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 March 22, 2013.

Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of

publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction 3
2. Format 3
3. Behaviour 4
4. Security Considerations 4
5. IANA Considerations 5
6. References 5
  6.1. Normative References 5
  6.2. Informative References 5
Authors' Addresses 5

1. Introduction

This document extends the base PCP [I-D.ietf-pcp-base] with the ability to associate a description with a PCP-instantiated mapping: DESCRIPTION Option.

This option can be used in the context of [I-D.ietf-pcp-upnp-igd-interworking].

This option has been defined first in [I-D.boucadair-pcp-extensions].

2. Format

This option can be used by a user (or an application) to indicate a description associated with a given mapping such as "FTP server", "My remote access to my CP router", "Camera", "Network attached storage serve", etc.

Issues related to the usage of this field for troubleshooting or for any further usage are out of scope of this document.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| DESCRIPTION   |  Reserved     |           Length              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           Description                         |
:                                                               :
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

This Option:

Option Name: Description Option (DESCRIPTION)
Number: TBA in the optional-to-process range (IANA)
Purpose: Used to associate a text description with a mapping
Valid for Opcodes: MAP, PEER
Length: Variable
May appear in: both request and response
Maximum occurrences: 1

Figure 1: Description Option

Description field carries the description text.

3. Behaviour

DESCRIPTION Option is optional to be supported by PCP Servers and PCP Clients.

This option (Code TBA, Figure 1) MAY be included in a PCP MAP/PEER request to associate a description with the requested mapping.

The PCP Server MAY be configurable to accept the DESCRIPTION Option. If the PCP Server does not support the DESCRIPTION Option or it is configured to reject it, received DESCRIPTION Options MUST be ignored by the PCP Server and no DESCRIPTION Option MUST be included in the response. The PCP Server MUST store the content of DESCRIPTION Option only if it supports the DESCRIPTION Option and if it is configured to accept handling DESCRIPTION Options it receives.

If the PCP Client does not receive a DESCRIPTION Option in a response to a request enclosing a DESCRIPTION Option, this means the PCP Server does not support that Option. The PCP Client SHOULD avoid including the DESCRIPTION Option in any subsequent request to that PCP Server.

If the DESCRIPTION Option is not included in the request, the PCP Server MUST NOT include the DESCRIPTION Option in the associated response.

The maximum length SHOULD be configurable in the PCP Server. If a PCP Client includes a DESCRIPTION PCP Option with a length exceeding the maximum length supported by the PCP Server, only the portion of the Description field fitting that maximum length is stored by the PCP Server and returned to the PCP Client in the response. If the PCP Server receives a DESCRIPTION option having a length which does not exceed the maximum value configured, the PCP Server MUST record the complete sequence of the description text and MUST send back to the PCP Client the same DESCRIPTION Option as the one included in the request.

The PCP Client MUST NOT include empty DESCRIPTION Option (i.e., Length set to 0) in a request. Empty DESCRIPTION Options MUST be ignored by the PCP Server.

4. Security Considerations

Security considerations discussed in [I-D.ietf-pcp-base] must be considered.

5. IANA Considerations

The following PCP Option Codes are to be allocated in the optional-to-process range:

DESCRIPTION

6. References

6.1. Normative References

[I-D.ietf-pcp-base]
Wing, D., Cheshire, S., Boucadair, M., Penno, R., and P. Selkirk, "Port Control Protocol (PCP)", draft-ietf-pcp-base-26 (work in progress), June 2012.

[I-D.ietf-pcp-upnp-igd-interworking]
Boucadair, M., Dupont, F., Penno, R., and D. Wing, "Universal Plug and Play (UPnP) Internet Gateway Device (IGD)-Port Control Protocol (PCP) Interworking Function", draft-ietf-pcp-upnp-igd-interworking-03 (work in progress), September 2012.

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

6.2. Informative References

[I-D.boucadair-pcp-extensions]
Boucadair, M., Penno, R., and D. Wing, "Some Extensions to Port Control Protocol (PCP)", draft-boucadair-pcp-extensions-03 (work in progress), April 2012.

Authors' Addresses

Mohamed Boucadair
France Telecom
Rennes, 35000
France

Email: mohamed.boucadair@orange.com

Reinaldo Penno
Cisco
USA

Email: repenno@cisco.com

Dan Wing
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, California 95134
USA

Email: dwing@cisco.com

PCP Working Group
Internet-Draft
Intended status: Informational
Expires: March 28, 2013

M. Ait Abdesselam
M. Boucadair
A. Hasnaoui
J. Queiroz
France Telecom

September 24, 2012

PCP NAT64 Experiments
draft-boucadair-pcp-nat64-experiments-00

Abstract

This memo documents a set of PCP experiments conducted in NAT64 environment. Two services are detailed in the document: access to a video server behind NAT64 and SIP-based sessions. Both 3G and Wi-Fi IPv6-only connectivity have been used.

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 March 28, 2013.

Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as

described in the Simplified BSD License.

Table of Contents

1. Introduction 3
2. Software Modules & Modifications 3
2.1. PCP Server 3
2.2. NAT64 4
2.3. PCP Packet Generator 4
2.4. RA Daemon 5
2.5. DNS64 5
2.6. Wirshark PCP Dissector 5
2.7. SIP Proxy Server 5
2.8. SIP UA 5
2.9. PCP Server Discovery 6
2.9.1. DHCP 6
2.9.2. RA-based approach 7
3. Testbed Setup & Configuration 7
3.1. Handsets 7
3.2. IPv6-only APN for 3G 8
3.3. Wi-Fi Connectivity 8
3.4. Network Topology 9
4. Tested Services 10
4.1. HTTP Webcam Server Behind NAT64 11
4.2. SIP Use Case 12
4.2.1. Media Sessions 15
4.2.2. IPv6-only to IPv4-only 15
4.2.3. IPv4-only to IPv6-only 19
4.2.4. IPv6-only to IPv6-only 20
5. IANA Considerations 21
6. Security Considerations 21
7. Acknowledgements 22
8. Normative References 22
Authors' Addresses 23

1. Introduction

This document describes a set of PCP [I-D.ietf-pcp-base] experiments conducted in the context of NAT64 [RFC6146]. Both Wi-Fi and 3G configurations have been tested.

The main goals of these experiments are:
- o Port a NAT64 implementation to be controlled using PCP.
- o Integrate a PCP Client in an Android device.
- o Validate the PCP chain in the NAT64 context.
- o Assess the use of PCP for NAT64 traversal and delivery of services behind NAT64.
- o Evaluate the complexity to update applications to invoke PCP service or embed a PCP Client.

Two services are detailed in the document: access to video server behind NAT64 (Section 4.1) and SIP-based sessions (Section 4.2).

2. Software Modules & Modifications

The following sub-sections provide more details on the software modules used for the experiments.

2.1. PCP Server

The PCP server used for NAT64 experiments is based on the DS-Lite compliant daemon implementation from ISC. The base functionalities of this PCP Server are listed below:
- o Configurable port range to be used for the external port mapping for both TCP and UDP.
- o Support of MAP and PEER OpCodes.
- o Support of THIRD_PARTY and PREFER_FAILURE Options.

The code has been updated as follows:

- o Add an interactive shell interface with basic commands to: view active mappings, list users, delete a specific user and reset a user’s epoch time, etc.
- o Adapt the behavior to be compatible with a NAT64 environment.
- o Support of DESCRIPTION PCP option [I-D.boucadair-pcp-description-option].
- o Support of PREFIX64 option [I-D.boucadair-pcp-nat64-prefix64-option].
- o Support of PORT_RESERVATION option [I-D.boucadair-pcp-rtp-rtcp].
- o Establish and maintain a communication channel to control the NAT64 module.

The PCP Server software architecture is shown in Figure 1.

```
 +-------------------------------------------------------------------+
 |  +---------------------------------+       +----------------+  |
 |  | _____Core module "main.c"______ |       |    "Com.c"     |  |
 |  | main_loop(), users_db, maps_db, |       |    Network     |  |
 |  | new_client(), new_mapping(),    <=======> communication  |  |
 |  | new_dynamic(), check_expire(),  |       |  ___module___  |  |
 |  | check_client(), PCPD_flags, ... |       | PCP net_socket <===>
 |  |                                 |       | Receive_PCP()  |  |
 |  +------------/\---||--------------+       | Send_PCP()     |  |
 |    cgn_init() ||   ||                      | ...            |  |
 |               ||   || lib_init()           | NAT64 net_sock <===>
 |  +------------||---\/--------------+       | Ask_nat64(),   |  |
 |  | __PCP Server library "pcpd.c"__ |       | Ask_prefix(),  |  |
 |  |  Receive_request(),             |       | Add_binding(), |  |
 |  |  Send_response(),               |       | Add_session(), |  |
 |  |  add_mapping(), del_mapping(),  |       | Ask_client(),  |  |
 |  |  ...                            |       | ...            |  |
 |  +---------------------------------+       +----------------+  |
 +-------------------------------------------------------------------+
```

Figure 1: PCP server software architecture.

2.2. NAT64

The NAT64 module is based on the Viagenie’s Ecdysis open source implementation for Linux.

The compilation environment is Debian Squeeze 6.0 / Linux Kernel 2.6.32-5-686.

The main modifications we incorporated into Ecdysis module are listed below:

- o Add a management interface to: view mappings, delete mapping, add a new mappings, etc.
- o Add a listening TCP interface for the PCP server.
- o Instantiate/delete mappings when a command is received from the PCP Server module.
- o Packets matching the explicit mapping are handled appropriately.

2.3. PCP Packet Generator

A basic Android software, denoted as "PCP Packet Generator", has been developed to generate customized PCP requests to be sent to a PCP Server.

This tool allows to set any values for the PCP fields and Options to be used. Received responses are handled, parsed and validated. The content of received PCP requests are shown in a human readable format.

2.4. RA Daemon

The radvd v1.8.6 (Router Advertisement Daemon) is used to send RA messages for a stateless configuration of IPv6 mobile devices.

2.5. DNS64

Bind9 v9.9.0 is used as DNS64 server. This PREFIX64 is configured to NAT64 and DNS64: 2001:688:1f94:300a::/96.

A DNS record is created for "mysip.fr", which is used to contact the SIP Proxy Server. DNS64 returns IPv4-embedded IPv6 addresses when resolving "mysip.fr" is: [PREIFX64+sip_serv_ipv4].

AAAA DNS record is created for "mypcp.fr" used to contact the PCP Server. DNS64 reruns the IPv6 address of the NAT64 [2001:688:1f94:3000::2].

2.6. Wirshark PCP Dissector

The used wireshark version is 1.8.0 running on Linux 2.6.32-5-686.

For PCP, an extension "pcp dissector" has been used to parse PCP packets (with the port 5351 as destination or source port). The recognized opcodes are MAP and PEER. Recognized options are all those conforming to version 18 of [I-D.ietf-pcp-base].

2.7. SIP Proxy Server

The used SIP server is Asterisk v1.2 running on a Debian Squeeze 6.0 with Kernel image: 2.6.32-5-686.

The default configuration is used. No extra feature to assist NAT traversal nor IPv6 support were activated.

2.8. SIP UA

The selected SIP UA for mobile devices is Linphone 1.3.2 for Android. Linphone is based on the eXosip2 C library.

For our experiments, Linphone module has been updated with the main modifications listed below:

- o Add to the GUI a configuration option to set the domain name of the PCP server to be used. Leaving the option field blank disables PCP.
- o If PCP is enabled, a PCP request is sent to instantiate a mapping for the port used for SIP signaling messages (random port is used). The retrieved external IP and port number will be used in the CONTACT and VIA fields of all SIP messages headers. The same request is used to retrieve the PREFIX64 used by the NAT64. The returned PREFIX64 is stored by the UA.
- o If PCP is enabled, for any incoming or outgoing session, two PCP requests are sent to create four bindings for the audio and video RTP and RTCP flows. The allocated external IP and ports are returned in the session description offer/answer.
- o For an incoming call (from IPv4-only network), the IP address included in the INVITE headers and SDP offer is the IPv4 address representing the (IPv6) calling host. The PREFIX64 of the NAT64, returned in PCP, is used to synthesize an IPv6 address based on the IPv4 address contained in the SDP offer [RFC6052].
- o For an outgoing call, the same problem occurs to send the "200 OK" message. The same PREFIX64 is used to construct the IPv4-converted IPv6 address representing the IPv4-only UA.
- o Other minor GUI modifications.

Linphone has been also patched to support ALTC attribute [I-D.boucadair-mmusic-altc] (see Section 4.2.4).

2.9. PCP Server Discovery

PCP Client needs to implement a method to discover a PCP server no located in the first hop. PCP_SERVER is added automatically to "host" file owing to two methods detailed below:

1. [I-D.ietf-pcp-dhcp]: DHCPv6 PCP option is used to discover a PCP server name. The DHCPv6 server, when configured to do so, provides the requested PCP server information by including one or more PCP server names option in its response.
2. I-D.boucadair-pcp-nodhcp-discovery specifies Router Advertisement option to learn the PCP Server.

Note these discovery methods are not integrated in Android but are tested using Linux Fedora 15.

2.9.1. DHCP

2.9.1.1. DHCP Server

DHCPv6 server from ISC is used to integrate the PCP DHCPv6 option. We modified the configuration file: "inetc/dhcp/dhcpd6.conf" to provision a PCP server name to clients.

2.9.1.2. DHCP Client

Setting up the clients is relatively easy. There are several implementations available but we used the DHCPv6 client embedded in Fedora 15.

We modified the configuration file: "etc/dhcp/dhclient6.conf" to request a specific option (PCP OPTION). The same code is used in the client and server sides:

```
#pcp server option option dhcp6.OPTION_PCPSERVER code 156 = string;
```

The client is updated with a script for analyzing, extracting and storing the content of received PCP option.

2.9.2. RA-based approach

As an alternative to DHCP, we also implemented an RA-based approach to learn the PCP Name of PCP Server(s). This option (called PCPS) contains one or more PCP domain names sharing the Lifetime value.

The router advertisement daemon (radvd-1.8.6) is run by Linux systems acting as IPv6 routers.

An IPv6 host can configure the PCP server of one or more PCPSERVER via RA messages periodically sent by a router or solicited by a Router Solicitation (RS).

3. Testbed Setup & Configuration

3.1. Handsets

The used handsets model is:

Samsung Galaxy SII GT-I9100.

The mobile devices have been upgraded to Android ICS 4.0.3 with:

- o ROM version: IML74K.XXLPQ.
- o Kernel version: 3.0.15-9100XXLPQ-CL223505.
- o Base band version: I9100XXLPQ.

The latest Android version is used to avoid some well known IPv6 issues.

ICS 4 version does not support DHCPv6, and the IPv6 addresses are autoconfigured using RA.

For advanced network utilities, the smartphones have been rooted to unlock access functionalities as setting of DNS server, IP addressing, easiest development/debugging, custom application install, etc.

Busybox is also installed to add more configuration tools.

The "IP WebCam" is a software that turns a mobile Android device into a wireless webcam with multiple viewing options such as a VideoPlayer or web browser by creating an HTTP server that broadcasts video and audio flows by converting it to JavaScript.

URL: https://play.google.com/store/apps/details?id=com.pas.webcam

3.2. IPv6-only APN for 3G

An IPv6-only APN from Orange France has been used to assess the PCP behavior over a 3G network.

3.3. Wi-Fi Connectivity

The Wi-Fi IPv6-only environment is set using a 45 Mbps Wireless Access Point Netgear-WG602-v4.

```
+-----------------------------ISSUE---------------------------------+
|The Android handsets can access to a Wi-Fi IPv6-only network by    |
|configuring at first a static IPv4 address to be used with SSID    |
|network in the Android Wi-Fi configuration menus.  Once the device |
|connected to the network and the wlan0 interface got an IPv6 global|
|address (by RA), the IPv4 address can be deleted.  This avoids the |
|device to ask automatically for a DHCPv4 server, and allows to     |
|connect to IPv6-only networks.  This is a problem due to lack of   |
|global support of IPv6 in Android.                                 |
+-------------------------------------------------------------------+
```

DHCPv6 is also not supported. The DNS server must be set manually

using the shell command:

```
#setprop net.dns1 dns_serv_address
```

3.4. Network Topology

Two network topologies are used for the tests. For both configurations, the same NAT64, DNS64 and PCP server are used. NAT64/PCP Server is configured with

- o An IPv4 address pool
- o An IPv6 prefix (/64)
- o Only the handsets change location from 3G network to Wi-Fi local IPv6-only network. /64 is allocated to the handset.
- o A route to the IPv4 default gateway.
- o A route to the IPv6 default gateway.

The network topology is shown in Figure 2.

```
   +-------------------------------------------------------------------+
   |     +--------+              +---------------+         ************|
   |     | Remote |--------------|    Internet   |         *   IPv4   *|
   |     |  host  |              |       v4      |         *  Network *|
   |     +--------+              +---------------+         ************|
   |    /________/                          |                          |
   |                                        |                          |
   |                                        |                          |
   | +------------------------+           +---------------+            |
   | |   161.105.194.14/29    |-----------|   Gateway v4  |            |
   | |   _________________    |           | 161.105.194.9 |            |
   | |                        |           +---------------+            |
   +-|      NAT64+DNS64       |----------------------------------------+
   |         +              |
   +-|      PCP Server        |----------------------------------------+
   | |   _________________    |     +-------------------------+        |
   | |2001:688:1f94:3000::2/64|--+--|       Gateway v6        |        |
   | +------------------------+  |  |2001:688:1f94:3000::1/64|        |
   |                             |  +-------------------------+        |
   |         +---------+         |              |                      |
   |         |  WiFi   |---------+   +------------------------+        |
   |         |  access |             |      Internet v6       |        |
   |         |  point  |             +------------------------+        |
   |         +---------+                         |                     |
   |            ^                  +------------------------+          |
   |            ^                  |    IPv6 only 3G APN    |          |
   |            ^                  +------------------------+          |
   |          ||__                               ^                     |
   |          |   | Handset on WiFi             ^                     |
   |          |   | Global assigned IPv6:       ^                     |
   |          |___| 2001:688:1f94:3000::XXXX/64 ^                     |
   |                                             ^                     |
   |                                             ^                     |
   |   *****************                     ||__                      |
   |   *  IPv6 Network  *                     |   | Handset on 3G       |
   |   *****************                     |   | Global assigned IPv6|
   |                                         |___| 2a01:cd01:XXXX/128  |
   +-------------------------------------------------------------------+
```

Figure 2: Global testbed topology.

4. Tested Services

4.1. HTTP Webcam Server Behind NAT64

The first tested service is an HTTP server running on a mobile device connected to a IPv6-enabled 3G network, that shows the video flows of the device cam.

The PCP Packet Generator is used to send a MAP request to the PCP server containing the following fields:

```
Client IP: [2a01:cd01:XXXX]
Request Opcode: MAP
Requested internal port: 8080
Suggested external address: [::ffff:0]
Suggested external port: 8080
Lifetime: 3000 sec
Transport protocol: TCP
Description: "HTTP Webcam server service"
```

The PCP response returns the external IPv4 address and the external assigned port to be used to access the HTTP Webcam server.

This example is shown in Figure 3.

```
   +-------------------------------------------------------------------+
   |   +--------+src port:XXXX    +---------------+        ************|
   |   | Remote |================>|    Internet   |        *   IPv4   *|
   |   |  host  | HTTP request    |       v4      |        *  Network *|
   |   +--------+ to              +---------------+        ************|
   |  /_________/ 161.105.194.14:8080       |                          |
   |                                        |                          |
   |                                        |                          |
   | +------------------------+           +---------------+            |
   | |   161.105.194.14/29    |<==========|   Gateway v4  |            |
   | |   _________________    |port:      | 161.105.194.9 |            |
   | | NAT64+DNS64+PCP server |  8080     +---------------+            |
   +-|   <BINDING CREATED>    |----------------------------------------+
     | [161.105.194.14:YY]    |
   +-|  <>[2a01:cd01::.8080]  |----------------------------------------+
   | |   _________________    |     +------------------------+         |
   | |2001:688:1f94:3000::2/64|====>|       Gateway v6       |         |
   | +------------------------+ src |2001:688:1f94:3000::1/64|         |
   |                        port:YY +------------------------+         |
   |                                            |                      |
   |                                            |                      |
   |                                            |                      |
   |                      ||__      +------------------------+         |
   |       Handset on 3G |   |~~~~~|                        |         |
   | Global assigned IPv6|   |     |    IPv6 only 3G APN    |         |
   |   2a01:cd01:XXXX/128|___|     |                        |         |
   |  HTTP Webcam server running   +------------------------+         |
   |           on port 8080                                            |
   |                                                 *****************|
   |                                                 *  IPv6 Network  *|
   |                                                 *****************|
   +-------------------------------------------------------------------+
```

Figure 3: Access to IPv4 server behind NAT64

4.2. SIP Use Case

The registration call flow for the IPv6-only SIP UA is depicted in Figure 4.

In the following examples, port 5070 is used instead of the default SIP port (5060).

At bootstrapping of the SIP UA, it retrieves the PREFIX64 used by the NAT64 and installs a mapping used for SIP registration.

```
          +---------+              +-------+       +------------+
          | Client4 |              | NAT64 |       |  IPv4 SIP  |
          |IPv6_only|              |+ PCP  |       |Proxy Server|
          | SIP UA  |              |Server |       | "Mysip.fr" |
          +---------+              +-------+       +------------+
               | (a) PCP MAP REQUEST   |                |
               |  Request SIP port MAP |                |
               |     PREFIX64_OPTION   |                |
               |======================>|                |
               | (b) PCP MAP RESPONSE  |                |
               | Assigned external:    |                |
               |              X.X.X.X:X|                |
               |   PREFIX64_OPTION     |                |
               |<======================|                |
               |   (1)SIP REGISTER     |(2)SIP REGISTER |
               |======================>|===============>|
               |  (4) SIP 200 OK       | (3) SIP 200 OK |
               |<======================|<===============|
               |                       |                |
```

Figure 4: SIP REGISTER

(a) Below is shown the content of the PCP MAP Request issued by Client4 towards the PCP Server:

```
Source: 2001:688:1f94:3000:289f:db7:e8ae:2988 port: 12345
Destination: 2001:688:1f94:3000::2.5351

 PCP Request:
  Version: 1
  R bit: Request (0)
  Opcode: MAP (0x01)
  Requested Lifetime: 36000 sec
  PCP Client's IP Address: 2001:688:1f94:3000:289f:db7:e8ae:2988
      (2001:688:1f94:3000:289f:db7:e8ae:2988)
  MAP Request: Protocol: UDP (17)
  Internal Port: 3938
  Suggested External Port: 3938
  Suggested External IP Address: ::ffff:0.0.0.0
  Option Code: Unknown (0x7f) Option Length: 12 bytes Data:
      000000000000000000000000
```

(b) The PCP MAP Response received from the PCP Server is shown below:

```
     Source: 2001:688:1f94:3000::2.5153
     Destination: 2001:688:1f94:3000:289f:db7:e8ae:2988.12345

      PCP Response:
       Version: 1
       R bit: Response (1)
       Opcode: Unknown (0x81)
       Result Code: 0
       Lifetime: 36000 sec
       Epoch Time: 1
       MAP Response Protocol: UDP (17)
       Internal Port: 3938
       Assigned External Port: 3938
       Assigned External IP Address: ::ffff:161.105.194.14 (::ffff:
           161.105.194.14)
       Option Code: PREFIX64 (0x7f) Reserved: 0 Option Length: 12 bytes
           Data: 200106881f94300a00000000
```

(1) Then, the UA uses the retrieved external IP address and port to generate the following SIP REGISTER message:

```
     Source: 2001:688:1f94:3000:289f:db7:e8ae:2988 port: 3938
     Destination: 2001:688:1f94:3000::a169:c20d port:5070

      SIP Message:
       REGISTER sip:mysip.fr SIP/2.0
       Via: SIP/2.0/UDP 161.105.194.14:3938;branch=z9hG4bK1572043597
       From: <sip:client4@mysip.fr:5070>;tag=893886783
       To: <sip:client4@mysip.fr:5070>
       Call-ID: 1271173454
       CSeq: 2 REGISTER
       Contact: <sip:client4@161.105.194.14:3938;line=b3433a7df33282d>
           Authorization: Digest username="client4", realm="asterisk",
           nonce="09f75e47", uri="sip:mysip.fr",
           response="826fcff4c6e84ee45fbfa52c351e6316", algorithm=MD5
       Max-Forwards: 70
       User-Agent: Linphone/3.4.0 (eXosip2/unknown)
       Expires: 3600
```

(2) SIP REGISTER is translated by the NAT64 using the PCP-instantiated mapping. This message is then forwarded to the SIP Proxy Server:

```
Source: 161.105.194.14:3938 (NAT64)
Destination: 161.105.194.13:5070 (SIP Proxy)
 Same SIP Message as (1).
```

(3) A positive response is generated by the SIP Proxy Server as shown

below:

```
     Source: 161.105.194.13:5070 (SIP Proxy)
     Source: 161.105.194.14:3938 (NAT64)

      SIP/2.0 200 OK
      Via: SIP/2.0/UDP 161.105.194.14:
           3938;branch=z9hG4bK1572043597;received=161.105.194.14
      From: <sip:client4@mysip.fr:5070>;tag=893886783
      To: <sip:client4@mysip.fr:5070>;tag=as0b92321f
      Call-ID: 1271173454
      CSeq: 2 REGISTER
      Server: Asterisk PBX 1.6.2.9-2+squeeze6
      Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER, SUBSCRIBE,
           NOTIFY, INFO
      Supported: replaces, timer
      Expires: 3600
      Contact: <sip:client4@
           161.105.194.14:3938;line=b3433a7df33282d>;expires=3600
```

(4) 200 OK message is translated by the NAT64 using the PCP-instantiated mapping:

```
       Source: 2001:688:1f94:3000::a169:c20d.5070
       Destination: 2001:688:1f94:3000:289f:db7:e8ae:2988.3938
        Same SIP message as (3).
```

At the end of this procedure, IPv6-only SIP UA is able to place and receive session requests.

PREFIX64 retrieved during this phase is used to build IPv4-embedded IPv6 addresses when receiving an IPv4 address in an SDP offer/answer.

4.2.1. Media Sessions

Both audio and video sessions are supported. The audio codecs used for these experiments are: speex 16 KHz, speex 8Khz, and gsm. The used video codecs are H264 and MPEG4.

4.2.2. IPv6-only to IPv4-only

Figure 5 and Figure 6 illustrate the exchanges which occur when initiating a SIP session from an IPv6-only UA to an IPv4-only SIP UA.

PCP exchanges take place at the bootstrapping of the SIP UA to reserve one or two pair of ports (one for audio and another one for video).

```
   +---------+            +-------+      +------------+     +---------+
   | Client4 |            | NAT64 |      |  IPv4 SIP  |     |IPv4-only|
   |IPv6_only|            |+ PCP  |      |Proxy Server|     | SIP UA  |
   | SIP UA  |            |Server |      | "Mysip.fr" |     | Client1 |
   +---------+            +-------+      +------------+     +---------+
        | (a) PCP MAP REQUEST   |                |                 |
        |Requested port:        |                |                 |
        |              port_A   |                |                 |
        |PORT_RESERVATION_OPTION|                |                 |
        |======================>|                |                 |
        | (b) PCP MAP RESPONSE  |                |                 |
        |Assigned port:         |                |                 |
        |               port_X  |                |                 |
        |PORT_RESERVATION_OPTION|                |                 |
```

Figure 5: Use PCP to reserve a pair of ports

(a) The following PCP MAP Request is issued from Client4 towards the PCP Server:

```
   Source: 2001:688:1f94:3000:289f:db7:e8ae:2988.12345
   Destination: 2001:688:1f94:3000::2.5351

    PCP Request:
     Version: 1
     R bit: Request (0)
     Opcode: MAP (0x01)
     Requested Lifetime: 36000 sec
     PCP Client’s IP Address: 2001:688:1f94:3000:289f:db7:e8ae:2988
         (2001:688:1f94:3000:289f:db7:e8ae:2988)
     MAP Request: Protocol: UDP (17)
     Internal Port: 7076
     Suggested External Port: 7076
     Suggested External IP Address: ::ffff:0.0.0.0
     Option Code: RTP (0x84) Option Length: 0 bytes Data: (NULL)
```

This request aims to reserve a pair or ports preserving parity and contiguity.

(b) PCP MAP Response from PCP Server to Client4:

```
      Destination: 2001:688:1f94:3000:289f:db7:e8ae:2988.12345
      Source: 2001:688:1f94:3000::2.5153
       PCP Response:
        Version: 1
        R bit: Response (1)
        Opcode: Unknown (0x81)
        Result Code: 0
        Lifetime: 36000 sec
        Epoch Time: 1
        MAP Response Protocol: UDP (17)
        Internal Port: 7076
        Assigned External Port: 7076
        Assigned External IP Address: ::ffff:161.105.194.14 (::ffff:
            161.105.194.14)
        Option Code: RTP (0x84) Option Length: 0 bytes Data: (NULL)
```

At the end of this procedure, two external ports are reserved in the NAT64: 7076 and 7077.

In this example, the PCP Server honors the requested external port. If the requested port was in use, an alternative pair of ports would be assigned.

Figure 6 illustrates the messages exchanged to establish a session between an IPv6-only UA and a remote IPv4-only UA.

```
+---------+            +-------+      +------------+     +---------+
| Client4 |            | NAT64 |      |  IPv4 SIP  |     |IPv4-only|
|IPv6_only|            |+ PCP  |      |Proxy Server|     | SIP UA  |
| SIP UA  |            |Server |      | "Mysip.fr" |     | Client1 |
+---------+            +-------+      +------------+     +---------+
     |  (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  |
     |======================>|===============>|================>|
     |                       |                |                 |
     |port_A           port_B|port_X                     port_B|
     |<======IPv6 RTP=======>|<============IPv4 RTP============>|
     |<===== IPv6 RTCP======>|<============IPv4 RTCP===========>|
     |port_A+1       port_B+1|port_X+1                  port_B+1|
```

Figure 6: IPv6 to IPv4 SIP Session

(1, 2, 3) Below is shown the content of the SIP INVITE message sent by Client4. This message uses the external IP address and port in SIP headers and SDP lines. This message is translated by the NAT64

without altering the SIP/SDP content.

```
INVITE sip:13@mysip.fr:5070 SIP/2.0
Via: SIP/2.0/UDP 161.105.194.14:56252;branch=z9hG4bK1876803184
From: <sip:client4@mysip.fr:5070>;tag=631384602
To: <sip:13@mysip.fr:5070> Call-ID: 1377792765 CSeq: 21 INVITE
Contact: <sip:client4@161.105.194.14:56252>
Authorization: Digest username="client4", realm="asterisk",
 nonce="3358d80b", uri="sip:13@mysip.fr:5070",
 response="41442e94f6610e6f383a355a1bdf3e48", algorithm=MD5
Content-Type: application/sdp Allow: INVITE, ACK, CANCEL, OPTIONS,
 BYE, REFER, NOTIFY, MESSAGE, SUBSCRIBE, INFO
Max-Forwards: 70
User-Agent: Linphone/3.4.0 (eXosip2/unknown)
Subject: Phone call Content-Length: 443

v=0
o=client4 2487 2487 IN IP4 161.105.194.14
s=Talk c=IN IP4 161.105.194.14
b=AS:256
t=0 0
m=audio 7076 RTP/AVP 111 110 3 101
a=rtpmap:111 speex/16000
a=fmtp:111 vbr=on a=rtpmap:110 speex/8000
a=fmtp:110 vbr=on a=rtpmap:3 GSM/8000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-11
m=video 9076 RTP/AVP 102 99
a=rtpmap:102 H264/90000
a=fmtp:102 profile-level-id=428014
a=rtpmap:99 MP4V-ES/90000
a=fmtp:99
profile-level-id=3
```

(4, 5, 6) The content of the 200 OK message is shown below:

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP 161.105.194.14:
   56252;branch=z9hG4bK1876803184;received=161.105.194.14
From: <sip:client4@mysip.fr:5070>;tag=631384602
To: <sip:13@mysip.fr:5070>;tag=as3d61114e
Call-ID: 1377792765 CSeq: 21 INVITE
Server: Asterisk PBX 1.6.2.9-2+squeeze6 Allow: INVITE, ACK, CANCEL,
  OPTIONS, BYE, REFER, SUBSCRIBE, NOTIFY, INFO Supported: replaces,
  timer
Contact: <sip:13@161.105.194.13>
Content-Type: application/sdp Content-Length: 414

v=0
o=root 1210300728 1210300728 IN IP4 161.105.194.13
c=IN IP4 161.105.194.13 b=CT:384
t=0 0
m=audio 13238 RTP/AVP 3 110 111 101
a=rtpmap:3 GSM/8000
a=rtpmap:110 speex/8000
a=rtpmap:111 G726-32/8000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-16 a=ptime:20
a=sendrecv
m=video 14466 RTP/AVP 102 99
a=rtpmap:102 H264/90000
a=rtpmap:99 MP4V-ES/90000
a=sendrecv
```

When this message is received by the IPv6-only UA, the IPv6-only UA uses PREFIX64 to build the IPv4-embedded IPv6 address corresponding to the IPv4 address included in the SDP response. RTP/RTCP flows are sent to that IPv6 address.

4.2.3. IPv4-only to IPv6-only

Figure 7 shows the messages exchanged to establish a SIP session initiated from an IPv4-only UA.

In this scenario, PREFIX64 is used to handle the SDP offer received by the IPv6-only UA from the IPv4-only UA. The IPv6-only UA uses PREFIX64 to build the IPv4-embedded IPv6 address corresponding to the IPv4 address included in the SDP offer. RTP/RTCP flows are sent to that IPv6 address.

```
   +---------+            +-------+      +------------+     +---------+
   | Client4 |            | NAT64 |      |  IPv4 SIP  |     |IPv4-only|
   |IPv6_only|            |+ PCP  |      |Proxy Server|     | SIP UA  |
   | SIP UA  |            |Server |      | "Mysip.fr" |     | Client1 |
   +---------+            +-------+      +------------+     +---------+
        |  (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  |
        |<======================|<===============|<================|
        |                       |                |                 |
        |port_A           port_B|port_X                      port_B|
        |<======IPv6 RTP=======>|<============IPv4 RTP============>|
        |<===== IPv6 RTCP======>|<============IPv4 RTCP===========>|
        |port_A+1       port_B+1|port_X+1                  port_B+1|
```

Figure 7: IPv4 to IPv6 SIP Session

4.2.4. IPv6-only to IPv6-only

Two scenarios have been tested:

1. The behavior of the IPv6-only UA is similar to the one described in Section 4.2.2: in this scenario, NAT64 is involved in the RTP exchanges between IPv6-only UAs.

2. In order to remove the NAT64 from the path, the Linphone module was patched to support ALTC attribute [I-D.boucadair-mmusic-altc]. Figure 8 shows an example of INVITE message generated by the IPv6-only SIP UA. The remote IPv6-only UA will use the IPv6 altc line to generate its response. As a consequence, IPv6 will be used to exchange RTP flows.

```
INVITE sip:13@mysip.fr:5070 SIP/2.0
Via: SIP/2.0/UDP 161.105.194.14:35011;branch=z9hG4bK702695557
From: <sip:client4@mysip.fr:5070>;tag=641336337
To: <sip:13@mysip.fr:5070>
Call-ID: 1532307201
CSeq: 20 INVITE
Contact: <sip:client4@161.105.194.14:35011>
Content-Type: application/sdp
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER, NOTIFY, MESSAGE, SUBSCRIBE, I
NFO
Max-Forwards: 70
User-Agent: Linphone/3.4.0 (eXosip2/unknown)
Subject: Phone call
Content-Length:   538

v=0
o=client4 3867 3867 IN IP4 161.105.194.14
s=Talk
c=IN IP4 161.105.194.14
b=AS:256
t=0 0
m=audio 7056 RTP/AVP 111 110 3 101
a=rtpmap:111 speex/16000
a=fmtp:111 vbr=on
a=rtpmap:110 speex/8000
a=fmtp:110 vbr=on
a=rtpmap:3 GSM/8000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-11
m=video 9056 RTP/AVP 102 99
a=rtpmap:102 H264/90000
a=fmtp:102 profile-level-id=428014
a=rtpmap:99 MP4V-ES/90000
a=fmtp:99 profile-level-id=3
a=altc: IP6 2001:688:1f94:3000:6c73:ea54:cef:2730 45678
a=altc: IP4 161.105.194.14 7056
```

Figure 8: PCP+ALTC Attribute

5. IANA Considerations

No request is made to IANA.

6. Security Considerations

This document does not introduce any security issue in addition to

what is discussed in [I-D.ietf-pcp-base].

7. Acknowledgements

Special thanks to X. Deng for porting Linphone to support ALTC attribute.

Many thanks to the authors of PCP Server (ISC) and NAT64 (Viagenie) modules.

8. Normative References

[I-D.boucadair-mmusic-altc]
Boucadair, M., Kaplan, H., Gilman, R., and S. Veikkolainen, "Session Description Protocol (SDP) Alternate Connectivity (ALTC) Attribute", draft-boucadair-mmusic-altc-05 (work in progress), April 2012.

[I-D.boucadair-pcp-description-option]
Boucadair, M., Penno, R., and D. Wing, "PCP Description Option", draft-boucadair-pcp-description-option-01 (work in progress), September 2012.

[I-D.boucadair-pcp-nat64-prefix64-option]
Boucadair, M., "Learn NAT64 PREFIX64s using PCP", draft-boucadair-pcp-nat64-prefix64-option-02 (work in progress), September 2012.

[I-D.boucadair-pcp-rtp-rtcp]
Boucadair, M. and S. Sivakumar, "Reserving N and N+1 Ports with PCP", draft-boucadair-pcp-rtp-rtcp-04 (work in progress), April 2012.

[I-D.ietf-pcp-base]
Wing, D., Cheshire, S., Boucadair, M., Penno, R., and P. Selkirk, "Port Control Protocol (PCP)", draft-ietf-pcp-base-27 (work in progress), September 2012.

[I-D.ietf-pcp-dhcp]
Boucadair, M., Penno, R., and D. Wing, "DHCP Options for the Port Control Protocol (PCP)", draft-ietf-pcp-dhcp-05 (work in progress), September 2012.

[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.

Authors' Addresses

Mehdi Ait Abdesselam
France Telecom
Issy Les Moulineaux
France

Email: mehdi.aitabdesselam@orange.com

Mohamed Boucadair
France Telecom
Rennes, 35000
France

Email: mohamed.boucadair@orange.com

Amina Hasnaoui
France Telecom
Issy Les Moulineaux,
France

Email: amina.hasnaoui@orange.com

Jaqueline Queiroz
France Telecom
Issy Les Moulineaux
France

Phone:
Email: jaqueline.queiroz@orange.com

PCP Working Group
Internet-Draft
Intended status: Standards Track
Expires: March 16, 2013

M. Boucadair
France Telecom

September 12, 2012

Learn NAT64 PREFIX64s using PCP
draft-boucadair-pcp-nat64-prefix64-option-02

Abstract

This document defines a new PCP Option/OpCode to learn the Prefix64(s) used by a PCP-controlled NAT64 device to build IPv4-embedded IPv6 addresses. This Option/OpCode is needed for successful communications when IPv4 addresses are used in referrals (e.g., SIP).

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 March 16, 2013.

Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction 3
2. Use Cases 4
3. Requirements Language 4
4. PREFIX64 Option 4
4.1. Format 4
4.2. Behaviour 5
5. GET_PREFIX64 OpCode 6
6. Flow Examples 7
6.1. Examples with PREFIX64 PCP Option 7
6.2. Examples with GET_PREFIX64 OpCode 9
7. IANA Considerations 11
8. Security Considerations 11
9. Acknowledgements 11
10. References 12
10.1. Normative References 12
10.2. Informative References 12
Author’s Address 12

1. Introduction

This document defines a new PCP Option/OpCode [I-D.ietf-pcp-base] to inform PCP Clients about the Prefix64 [RFC6052] used by a PCP-controlled NAT64 device [RFC6146].

This Option is required to help establishing communications between IPv6-only hosts and remote IPv4-hosts. An illustration example is shown in Figure 1. In this example, NAT64 is co-located with a PCP server while IPv6-only SIP UA interacts with a PCP Client.

In Figure 1, the PCP Client issues a PCP MAP request with PORT_RESERVATION_OPTION to reserve a pair of ports preserving parity and contiguity [I-D.boucadair-pcp-rtp-rtcp]. A pair of ports and an external IPv4 address are then returned by the PCP server to the requesting PCP Client. This information is 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)). 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)). At the end of this process, IPv4-only SIP UA can send media streams to the IPv4 address/port as indicated in the SDP offer while IPv6-only SIP UA can not send media streams as only IPv4 addresses are present in the SDP answer.

```
+---------+              +-----+       +------------+     +---------+
|IPv6-only|              |NAT64|       |  IPv4 SIP  |     |IPv4-only|
| SIP UA  |              |     |       |Proxy Server|     | SIP UA  |
+---------+              +-----+       +------------+     +---------+
    | (a) PCP MAP REQUEST   |                |                |
    |PORT_RESERVATION_OPTION|                |                |
    |======================>|                |                |
    | (b) PCP MAP RESPONSE  |                |                |
    |PORT_RESERVATION_OPTION|                |                |
    |<======================|                |                |
    |                       |                |                |
    |  (1) SIP INVITE       | (2) SIP INVITE |  (3) SIP INVITE |
    |======================>|===============>|================>|
    |   (6) SIP 200 OK      | (5) SIP 200 OK |  (4) SIP 200 OK |
    |<======================|<===============|<================|
    |                       |                |                |
```

Figure 1

A solution is proposed in Section 4 and Section 5.

2. Use Cases

This issue is not specific to SIP but it is valid for all applications using IP addresses in referrals. The option/OpCode defined in this document can be used in various schemes as listed below (the list is not exhaustive):

- For hosts with DNS64 capability, 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 Option, the host's stub-resolver can learn the prefix used for IPv6/IPv4 translator and synthesize AAAA records accordingly.

- As Peer-to-Peer (P2P) communications for real-time communication is becoming popular with RTCWEB (e.g., P2P for Media, data channels for file transfer etc), this option can be used to help for NAT64 traversal. SIP is only one example among those protocols.

- Can be used for any application using referrals.

3. 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].

4. PREFIX64 Option

4.1. Format

The format of PREFIX64 PCP Option is depicted in Figure 2.

```
 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 2: Prefix64 PCP Option

The description of the fields is as follows:

- o Option Code: To be assigned by IANA.

- o Option Length: Indicates in octets the length of the Prefix64. 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 by IANA.
  Purpose: Learn the prefix used by the NAT64 to build IPv4-embedded IPv6 addresses. This is be used by a host for local address synthesis (e.g., when IPv4 address is present in referrals).
  Valid for Opcodes: MAP
  Length: Variable
  May appear in: request, response.
  Maximum occurrences: 1

4.2. Behaviour

A PCP Client MAY include a PREFIX64 PCP Option in a MAP request to learn the IPv6 prefix used by an upstream PCP-controlled NAT64 device. When enclosed in a MAP request, PREFIX64 MUST be set to ::/96.

A PCP Server controlling a NAT64 SHOULD be configured to return the value of the Prefix64 used to build IPv4-embedded IPv6 addresses to requesting PCP Clients. When allowed, PREFIX64 PCP Option conveys the value of Prefix64.

A PCP Server controlling a NAT64 SHOULD inject a PREFIX64 PCP Option in MAP responses even if the option is not listed in the associated request.

Upon receipt of the PREFIX64 PCP Option, the host embedding the PCP Client uses Prefix64 for local address synthesize [RFC6052].

A PCP Client SHOULD associate each received Prefix64 with the PCP Server from which the Prefix64 information was retrieved.

5. GET_PREFIX64 OpCode

Discussion: Both PREFIX64 option and OpCode are maintained in this version of the document. Based on the WG inputs, both or only one of them will be maintained.

This OpCode allows to retrieve a list of Prefix64s configured on the PCP-controlled NAT64 (see Figure 3). "Prefix64/IPv4 Prefix Count" indicates the number of Prefix64 prefixes included in the response. Each Prefix64 is associated with an IPv4 prefix. "Prefix64/IPv4 Prefix Count" field MUST be set to 0 in a request and MUST be set to the number of included {Prefix64, IPv4 subnet} in a response.

This allows to return in the same response the list of configured PREFIX64s per IPv4 prefix range.

An IPv4 prefix is represented as "IPv4 Address/IPv4 Prefix Length". IPv6 Prefix Length field indicates in bits the length of the Prefix64; allowed values are 32, 40, 48, 56, 64 and 96. Prefix64 field MUST be set to ::/96 in a request and MUST be set to the value of the Prefix64 used to construct IPv4-embedded IPv6 addresses for a given IPv4 subnet. a wildcard "IPv4 Address/IPv4 Prefix Length" means the associated Prefix64 is valid for any IPv4 address.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Prefix64/IPv4 Prefix Count   |         Reserved (16bits)     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   IPv4 Prefix Length          |   IPv6 Prefix Length          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       IPv4 Address (32 bits)                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
:                       Prefix64 (Variable)                     :
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 3: GET_PREFIX64 Opcode

A server MUST be configured to accept or ignore this OpCode. At receipt of a request, if it is configured to accept this OpCode, the PCP Server controlling a NAT64 MUST return the list of configured Prefix64s for each IPv4 subnet. If a single Prefix64 is configured for all IPv4 addresses, a wildcard IPv4 prefix MUST be returned in the response together with the configured Prefix64.

Retrieved Prefix64s are used locally to construct IPv4-embedded IPv6 addresses. If several Prefix64s are discovered, if the destination IPv4 address matches an IPv4 prefix in the list, the associated Prefix64 is used to construct the corresponding IPv6 address.

6. Flow Examples

6.1. Examples with PREFIX64 PCP Option

Figure 4 shows an example of the use of the option defined in Section 4.

```
+---------+              +-----+       +------------+     +---------+
|IPv6-only|              |NAT64|       |  IPv4 SIP  |     |IPv4-only|
| SIP UA  |              |     |       |Proxy Server|     | SIP UA  |
+---------+              +-----+       +------------+     +---------+
    | (a) PCP MAP REQUEST   |               |                 |
    |PORT_RESERVATION_OPTION|               |                 |
    |     PREFIX64_OPTION   |               |                 |
    |======================>|               |                 |
    | (b) PCP MAP RESPONSE  |               |                 |
    |PORT_RESERVATION_OPTION|               |                 |
    |     PREFIX64_OPTION   |               |                 |
    |<======================|               |                 |
    |  (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  |
    |======================>|===============>|================>|
    |                       |               |                 |
    |<======IPv6 RTP=======>|<============IPv4 RTP============>|
    |<===== IPv6 RTCP======>|<============IPv4 RTCP===========>|
    |                       |                                 |
```

Figure 4: Example of IPv6 to IPv4 SIP initiated Session

In Steps (a) and (b), the IPv6-only SIP UA retrieves a pair of ports

to be used for RTP/RTCP, the external IPv4 address and the Prefix64 to be used to build IPv4-embedded IPv6 addresses. The retrieved IPv4 address and port numbers are used to build the SDP offer in Step (1) while Prefix64 is used to construct a corresponding IPv6 address of the IPv4 address enclosed in the SDP answer made by the IPv4-only SIP UA (Step 6). RTP/RTCP flows are exchanged between an IPv6-only SIP UA and an IPv4-only UA without requiring any ALG at the NAT64 and no particular function to be supported by the IPv4-only SIP Proxy Server to help establishing the session (e.g., Hosted NAT traversal).

Now when the session is initiated from IPv4 SIP UA (see Figure 5): Steps (a) and (b), the IPv6-only SIP UA retrieves a pair of ports to be used for RTP/RTCP, the external IPv4 address and the Prefix64 to be used to build IPv4-embedded IPv6 addresses. These two steps can be delayed until receiving the INVITE message (Step 3).

It is recommended to pre-reserve a pair of port to optimize the required session establishment delay.

The retrieved IPv4 address and port numbers are used to build the SDP answer in Step (4) while Prefix64 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 and no particular function to be supported by the IPv4-only SIP Proxy Server to help establishing the session (e.g., Hosted NAT traversal).

```
   +---------+             +-----+       +------------+     +---------+
   |IPv6-only|             |NAT64|       |  IPv4 SIP  |     |IPv4-only|
   | SIP UA  |             |     |       |Proxy Server|     | SIP UA  |
   +---------+             +-----+       +------------+     +---------+
       | (a) PCP MAP REQUEST   |               |                 |
       |PORT_RESERVATION_OPTION|               |                 |
       |     PREFIX64_OPTION   |               |                 |
       |======================>|               |                 |
       | (b) PCP MAP RESPONSE  |               |                 |
       |PORT_RESERVATION_OPTION|               |                 |
       |     PREFIX64_OPTION   |               |                 |
       |<======================|               |                 |
       |  (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  |
       |<======================|<===============|<================|
       |                       |               |                 |
       |<======IPv6 RTP=======>|<============IPv4 RTP============>|
       |<===== IPv6 RTCP======>|<============IPv4 RTCP===========>|
       |                       |                                 |
```

Figure 5: Example of IPv4 to IPv6 SIP initiated Session

6.2. Examples with GET_PREFIX64 OpCode

Figure 6 shows an example of the use of the OpCode defined in Section 5.

Unlike previous examples, two requests are needed to place this session: Steps (a) and (b) are used to retrieve the list of {IPv4 subnet, Prefix64::/n} while Steps (c) and (d) are used to reserve a pair of port and learn the assigned IPv4 address. The remaining steps are similar to Figure 4.

The order of sending the requests is shown for illustration purposes. Another order to issue the request may be adopted, e.g.,

1. GET_PREFIX64 and MAP requests can be sent simultaneously.

2. GET_PREFIX64 request can be sent after MAP returned an IPv4 external IP address.

3. GET_PREFIX64 request can be issued at the bootstrap of the application. No need to issue the request for each new session.

```
   +---------+             +-----+       +------------+     +---------+
   |IPv6-only|             |NAT64|       |  IPv4 SIP  |     |IPv4-only|
   | SIP UA  |             |     |       |Proxy Server|     | SIP UA  |
   +---------+             +-----+       +------------+     +---------+
       |(a)PCP GET_PREFIX64 REQ|               |                 |
       |======================>|               |                 |
       |(b)PCP GET_PREFIX64 REP|               |                 |
       |List of {IPv4 subnet,  |               |                 |
       |         Prefix64::/n} |               |                 |
       |<======================|               |                 |
       | (c) PCP MAP REQUEST   |               |                 |
       |PORT_RESERVATION_OPTION|               |                 |
       |======================>|               |                 |
       | (d) PCP MAP RESPONSE  |               |                 |
       |PORT_RESERVATION_OPTION|               |                 |
       |<======================|               |                 |
       |  (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  |
       |======================>|===============>|================>|
       |                       |                |                 |
       |<======IPv6 RTP=======>|<============IPv4 RTP============>|
       |<===== IPv6 RTCP======>|<============IPv4 RTCP===========>|
       |                       |                                  |
```

Figure 6: Example of SIP Initiated Session

In the example shown in Figure 7, once the IPv6-only Client discovered the IPv4 address of the remote IPv4-only server, it retrieves the PREFIX64 to be used to build an IPv4-embedded IPv6 address for that server. This is achieve using GET_PREFIX64 PCP OpCode (Steps (a) and (b)). The client uses PREFIX64 to construct an IPv6 address and then initiates a TCP connection (Steps (1) to (4)).

The usage shown in Figure 7 depicts a typical usage of GET_PREFIX64 PCP OpCode when a DNS64 capability is embedded in the host.

```
          +---------+             +-----+            +---------+
          |IPv6-only|             |NAT64|            |IPv4-only|
          | Client  |             |     |            |  Server |
          +---------+             +-----+            +---------+
              |                     |                      |
              |(a)PCP GET_PREFIX64 REQ|                    |
              |======================>|                    |
              |(b)PCP GET_PREFIX64 REP|                    |
              |                     |                      |
              |    (1) TCP SYN      |    (2) TCP SYN       |
              |======================>|======================>|
              |   (4) TCP SYN/ACK   |   (3) TCP SYN/ACK    |
              |<======================|<======================|
              |    (5) TCP ACK      |    (6) TCP ACK       |
              |======================>|======================>|
              |                     |                      |
```

Figure 7: Example of TCP Session

7. IANA Considerations

This document request a new PCP OpCode:

GET_PREFIX64

This document request a new PCP option:

PREFIX64

8. Security Considerations

This document does not introduce any security issue in addition to what is taken into account in [I-D.ietf-pcp-base].

9. Acknowledgements

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

10. References

10.1. Normative References

[I-D.ietf-pcp-base]
Wing, D., Cheshire, S., Boucadair, M., Penno, R., and P. Selkirk, "Port Control Protocol (PCP)", draft-ietf-pcp-base-26 (work in progress), June 2012.

[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.

[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.

10.2. Informative References

[I-D.boucadair-pcp-rtp-rtcp]
Boucadair, M. and S. Sivakumar, "Reserving N and N+1 Ports with PCP", draft-boucadair-pcp-rtp-rtcp-04 (work in progress), April 2012.

Author’s Address

Mohamed Boucadair
France Telecom
Rennes, 35000
France

Email: mohamed.boucadair@orange.com

PCP WG
Internet-Draft
Intended status: Standards Track
Expires: April 18, 2013

M. Boucadair
France Telecom
S. Sivakumar
Cisco

October 15, 2012

Reserving N and N+1 Ports with PCP
draft-boucadair-pcp-rtp-rtcp-05

Abstract

This document defines a new PCP Option to reserve a pair of ports (N and N+1) by a PCP-controlled device while preserving the parity and contiguity. This PCP Option eases the NAT traversal for applications having requirements on the port parity and contiguity (e.g., RTP/RTCP).

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 RFC 2119 [RFC2119].

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 April 18, 2013.

Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents

(http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction 3
2. Why N/N+1 Option is Needed? 3
3. Definition of the Port Reservation Option 4
3.1. Requirements 4
3.2. Rationale 4
3.3. PCP Port Reservation Option 5
4. Client Behaviour 5
5. Server Behaviour 6
6. Illustration Examples 7
6.1. Port Reservation Option Not Supported by The PCP Server . 7
6.2. Port Reservation Option Is Supported by The PCP Server . . 8
6.3. Delete the Mappings 10
7. IANA Considerations 12
8. Security Considerations 12
9. Acknowledgments 12
10. References 13
10.1. Normative References 13
10.2. Informative References 13
Authors' Addresses 13

1. Introduction

This document defines a new PCP Option [I-D.ietf-pcp-base] which aims to ease the traversal of RTP/RTCP based applications [RFC3550] when a NAT is involved in the path.

The main advantage of using PCP is it does not need any further feature to be supported by the outbound proxy to assist the remote endpoint to successfully establish media sessions. In particular, ALGs are not required in the NAT for this purpose and no dedicated functions at the media gateway are needed.

The base PCP specification allows to retrieve the external IP address and external port to be conveyed in the SIP signaling messages [RFC3261]. Therefore SIP Proxy Servers do not need to support means to ease the NAT traversal of SIP messages (e.g., [RFC5626], [RFC6223], etc.). Another advantage of using the external IP address and port is this provides a hint to the proxy server there is no need to return a small expire timer (e.g., 60s).

This option has been implemented as reported in [I-D.boucadair-pcp-nat64-experiments]; no issue has been reported in that document.

2. Why N/N+1 Option is Needed?

Traditionally the voice/video applications that use RTP and RTCP would specify only the RTP port that the application would use for streaming the RTP data. The inherent assumption is that the RTCP traffic will be sent on the next higher port. Below is provided an excerpt from [RFC3550]:

> "RTP relies on the underlying protocol(s) to provide de-multiplexing of RTP data and RTCP control streams. For UDP and similar protocols, RTP SHOULD use an even destination port number and the corresponding RTCP stream SHOULD use the next higher (odd) destination port number. For applications that take a single port number as a parameter and derive the RTP and RTCP port pair from that number, if an odd number is supplied then the application SHOULD replace that number with the next lower (even) number to use as the base of the port pair. For applications in which the RTP and RTCP destination port numbers are specified via explicit, separate parameters (using a signaling protocol or other means), the application MAY disregard the restrictions that the port numbers be even/odd and consecutive although the use of an even/odd port pair is still encouraged."

[RFC3605] defines an explicit "a=RTCP" SDP attribute for some applications using a distinct port than RTP+1. Even though [RFC3605] defines a new attribute for explicitly specifying the RTCP attribute for the SDP based applications, but since it is not a MUST to use this attribute, there are still applications that are not compliant with this RFC. There are also non-SDP based applications that use RTP/RTCP like H323, that make the assumption that RTCP streaming will happen on RTP+1 port.

In order for these applications to work across NAT, the NAT device must have an application layer gateway, that would allocate two consecutive ports. In a PCP context, a similar functionality need to be provided for the PCP Client to request two consecutive ports and the PCP Server to allocate and respond with the information of the allocated port.

This document describes the mechanism to request a pair of consecutive ports for a PCP-controlled device and the corresponding mechanism for the PCP Server to allocate and respond to the port allocation request.

It is acknowledged that modern applications adopt new approaches (e.g., use the same port for both RTP and RTCP) which does not encounter the problem raised above. This document do not target those applications but "legacy" ones.

3. Definition of the Port Reservation Option

3.1. Requirements

The PCP Option used to reserve a port pair should meet the following requirements:

1. Preserve the port parity as discussed in Section 4.2.2 of [RFC4787].

2. Preserve port contiguity as discussed in Section 4.2.3 of [RFC4787] (i.e., RTCP = RTP+1).

3.2. Rationale

Since PCP does not support a mechanism to include multiple port numbers in the same request/response, only the RTP port is explicitly signaled in PCP messages. The companion port (i.e., RTCP port) is reserved too by the PCP Server.

3.3. PCP Port Reservation Option

The format of the PCP Port Reservation Option is defined in Figure 1.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|PORT_RESRV_OPT |  Reserved     |           0..0                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

This Option:

Option Name: Port Reservation Option (PORT_RESRV_OPT)
Number: TBA (IANA)
Purpose: Used to retrieve a pair of ports
Valid for Opcodes: MAP
Length: 0
May appear in: both request and response
Maximum occurrences: 1

Figure 1: Port Reservation Option (a.k.a., N/N+1 port)

4. Client Behaviour

To retrieve a pair of ports following the requirements listed in Section 3.1, the PCP Client adds the Port Reservation Option to its PCP MAP request. The PCP Client MAY indicate its preferred external port. This port number is likely to be equal to the internal port indicated in the PCP request.

Once a response is received from the PCP Server, the PCP Client checks whether the Port Reservation Option is supported by the peer PCP Server following the procedure defined in Section 7.3 of [I-D.ietf-pcp-base].

If the answer is positive, the PCP Client retrieves the mapping returned by the PCP Server; in particular the external port number should be even. For the RTP case, this port is indicated to the remote peer as the port number used for RTP flows; RTCP is assumed to use the returned external port number + 1.

If the Port Reservation Option is not supported by the PCP Server, and according to the port quota, only the RTP port can be signaled to the remote endpoint (e.g., SDP offer/answer [RFC4566]). RTCP flows are likely to fail if no mechanism to assist the traversal

of RTCP flows is supported (e.g., "a=RTCP" attribute).

When a pair of ports is retrieved from the PCP Server, two mappings are instantiated in both the PCP Server and PCP Client. For explicit deletion of these mappings, the PCP Client and PCP Server follow the procedure defined in Section 11.5 of [I-D.ietf-pcp-base] for each port mapping.

To reduce the delay to establish media sessions, the PCP Client MAY reserve a pair of ports once the (SIP) registration phase has been successfully completed. These pair of ports will be included in SDP offers/answsers for instance.

5. Server Behaviour

Upon receiving the Port Reservation Option in a PCP request, the PCP Server validates the request for the supported OpCode values. If an unrecognized value is received a Invalid request error is returned to the PCP Client (e.g., using MALFORMED_REQUEST error). The reason for rejecting the request could be an invalid internal IP address, invalid Internal port, etc.

For a valid request, the PCP Server collects the Internal port and the hinted external port and verify against any administrative rules to allow or disallow the PCP Client from making this request. An example of an administrative rule will be by fulfilling the request it would put the client over its administratively allowed limits. In those cases, the PCP Server will treat this as an error and this is handled the same way as described in [I-D.ietf-pcp-base] for the denial of honoring the request with the appropriate Opcode.

To handle the PCP Reservation Option by the PCP Server, the procedure defined in Section 7.3 of [I-D.ietf-pcp-base] should be followed. When PCP Reservation Option is not supported, the PCP Server MUST treat the request as any PCP request to create an individual mapping. If port parity preservation is supported by the PCP Server, an even port is likely to be returned to the PCP Client. Otherwise, a port is returned if the port quota is not reached.

The following describes the behavior of the PCP Server when the PCP Reservation Option is supported.

The PCP Server should request the controlling NAT device to allocate a pair of consecutive ports. If there is a hinted external port present in the request, the server MAY try to honor the request. The PCP Server MUST honor the parity by requesting the allocation of ports that match the parity. However, there is no guarantee that the

hinted external ports are available or be allocated. Two mappings are therefore instantiated by the PCP Server with the same lifetime value. These mappings are treated as any individual mapping.

If a mapping already exists and the PCP Reservation Option can be honored, the PCP Server instantiate the companion mapping and sends back a positive answer to the requesting PCP Client.

If the port allocation failed either because of the unavailability of ports or the port parity could not be honored, the PCP Server SHOULD reserve only one external port. The PCP Server SHOULD indicate in the response that the PCP Reservation Option has not been honored as specified in Section 6.3 of [I-D.ietf-pcp-base].

If the request contains the PREFER_FAILURE option and one or both hinted external ports (i.e., the hinted external port number and hinted external port number + 1) cannot be allocated, the PCP Server MUST reply with result code CANNOT_PROVIDE_EXTERNAL_PORT.

6. Illustration Examples

This section provides a list of examples to illustrate the usage of PCP Port Reservation Option.

6.1. Port Reservation Option Not Supported by The PCP Server

Figure 2 shows an example of the flow exchange which is observed when the PORT_RESERVATION_OPTION is not supported by the PCP Server.

```
                        +------+                          +------+
                        | PCP  |                          | PCP  |
                        |Client|                          |Server|
                        +------+                          +------+
                           |       (1) PCP MAP Request         |
                           |           protocol= UDP           |
                           | internal-ip-address= 198.51.100.1 |
                           |       internal-port= 6000         |
                           |      PORT_RESERVATION_OPTION      |
                           |---------------------------------->|
                           |                                   |
                           |      (2) PCP MAP Response         |
                           |           protocol= UDP           |
                           | internal-ip-address= 198.51.100.1 |
                           |       internal-port= 6000         |
                           |   external-ip-address= 192.0.2.1  |
                           |        external-port= 15659       |
                           |       assigned-lifetime= 3600     |
                           |   UNSUPP_OPTION(PORT_RESRV_OPT)   |
                           |<----------------------------------|
                           |                                   |
```

Figure 2: Flow Example of a PCP Server which does not support the Port Reservation Option

6.2. Port Reservation Option Is Supported by The PCP Server

Figure 3 and Figure 4 illustrate two examples of the flow exchanges which are observed when the PORT_RESERVATION_OPTION is supported by the PCP Server. Figure 3 shows an example of a PCP Server supporting the option and honoring the requested external port number. Figure 4 shows an example of a PCP Server supporting the option but not honoring the requested external port number.

```
                      +------+                          +------+
                      | PCP  |                          | PCP  |
                      |Client|                          |Server|
                      +------+                          +------+
                         |       (1) PCP MAP Request         |
                         |           protocol= UDP           |
                         | internal-ip-address= 198.51.100.1 |
                         |       internal-port= 6000         |
                         |      PORT_RESERVATION_OPTION      |
                         |---------------------------------->|
                         |                                   |
                         |      (2) PCP MAP Response         |
                         |           protocol= UDP           |
                         | internal-ip-address= 198.51.100.1 |
                         |       internal-port= 6000         |
                         |   external-ip-address= 192.0.2.1  |
                         |        external-port= 6000        |
                         |       assigned-lifetime= 3600     |
                         |      PORT_RESERVATION_OPTION      |
                         |<----------------------------------|
                         |                                   |
```

Figure 3: Flow Example of a PCP Server supporting the option and honoring the hinted external port

```
                        +------+                          +------+
                        | PCP  |                          | PCP  |
                        |Client|                          |Server|
                        +------+                          +------+
                           |       (1) PCP MAP Request         |
                           |           protocol= UDP           |
                           | internal-ip-address= 198.51.100.1 |
                           |       internal-port= 6000         |
                           |      PORT_RESERVATION_OPTION      |
                           |---------------------------------->|
                           |                                   |
                           |      (2) PCP MAP Response         |
                           |           protocol= UDP           |
                           | internal-ip-address= 198.51.100.1 |
                           |       internal-port= 6000         |
                           |   external-ip-address= 192.0.2.1  |
                           |        external-port= 12000       |
                           |       assigned-lifetime= 3600     |
                           |      PORT_RESERVATION_OPTION      |
                           |<----------------------------------|
                           |                                   |
```

Figure 4: Flow Example of a PCP Server supporting the option but not honoring the hinted external port

6.3. Delete the Mappings

Figure 5 and Figure 6 shows the exchanges that occur to delete the created mappings.

```
                        +------+                           +------+
                        | PCP  |                           | PCP  |
                        |Client|                           |Server|
                        +------+                           +------+
                           |          PCP MAP Request          |
                           |           protocol= UDP           |
                           | internal-ip-address= 198.51.100.1 |
                           |       internal-port= 6000         |
                           |   external-ip-address= 192.0.2.1  |
                           |        external-port= 6000        |
                           |       requested-lifetime= 0       |
                           |---------------------------------->|
                           |                                   |
                           |          PCP MAP Request          |
                           |           protocol= UDP           |
                           | internal-ip-address= 198.51.100.1 |
                           |       internal-port= 6001         |
                           |   external-ip-address= 192.0.2.1  |
                           |        external-port= 6001        |
                           |       requested-lifetime= 0       |
                           |---------------------------------->|
                           |                                   |
```

Figure 5: Flow example to delete the mappings

```
                        +------+                          +------+
                        | PCP  |                          | PCP  |
                        |Client|                          |Server|
                        +------+                          +------+
                           |                                 |
                           |          PCP MAP Request        |
                           |           protocol= UDP         |
                           | internal-ip-address= 198.51.100.1 |
                           |       internal-port= 6000       |
                           |   external-ip-address= 192.0.2.1  |
                           |        external-port= 12000     |
                           |       requested-lifetime= 0     |
                           |--------------------------------->|
                           |                                 |
                           |          PCP MAP Request        |
                           |           protocol= UDP         |
                           | internal-ip-address= 198.51.100.1 |
                           |       internal-port= 6001       |
                           |   external-ip-address= 192.0.2.1  |
                           |        external-port= 12001     |
                           |       requested-lifetime= 0     |
                           |--------------------------------->|
                           |                                 |
```

Figure 6: Flow example to delete the mappings (2)

7. IANA Considerations

This document requests the assignment of a new PCP Option code:

Option Name	Value
PORT_RESERVATION_OPTION	TBA

8. Security Considerations

This document does not introduce any security issue in addition to what is taken into account in [I-D.ietf-pcp-base].

9. Acknowledgments

Many thanks to S. Perrault for his comments.

10. References

10.1. Normative References

[I-D.ietf-pcp-base]
Wing, D., Cheshire, S., Boucadair, M., Penno, R., and P. Selkirk, "Port Control Protocol (PCP)", draft-ietf-pcp-base-28 (work in progress), October 2012.

[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.

[RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, July 2003.

[RFC3605] Huitema, C., "Real Time Control Protocol (RTCP) attribute in Session Description Protocol (SDP)", RFC 3605, October 2003.

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

[RFC4787] Audet, F. and C. Jennings, "Network Address Translation (NAT) Behavioral Requirements for Unicast UDP", BCP 127, RFC 4787, January 2007.

10.2. Informative References

[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.

[RFC5626] Jennings, C., Mahy, R., and F. Audet, "Managing Client-Initiated Connections in the Session Initiation Protocol (SIP)", RFC 5626, October 2009.

[RFC6223] Holmberg, C., "Indication of Support for Keep-Alive", RFC 6223, April 2011.

Authors' Addresses

Mohamed Boucadair
France Telecom
Rennes, 35000
France

Email: mohamed.boucadair@orange.com

Senthil Sivakumar
Cisco
7100 Kit Creek Road
Research Triangle Park, North Carolina 27709
USA

Email: ssenthil@cisco.com

PCP Working Group
Internet-Draft
Intended status: Standards Track
Expires: March 17, 2013

M. Boucadair
France Telecom
R. Penno
D. Wing
Cisco

September 13, 2012

PCP Server Selection
draft-boucadair-pcp-server-selection-00

Abstract

This document specifies the behavior to be followed by the PCP Client to contact its PCP Server(s) when one or several PCP Names are configured. Multiple Names may be configured to a PCP Client in some deployment contexts such as multi-homing.

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 RFC 2119 [RFC2119].

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 March 17, 2013.

Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents

(http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction 3
2. Terminology 3
3. Name Resolution 3
4. IP Address Selection 4
  4.1. Serial Queries 4
5. Examples 4
  5.1. Example 1 5
  5.2. Example 2 5
  5.3. Example 3 5
6. Security Considerations 6
7. IANA Considerations 6
8. Acknowledgements 6
9. References 6
  9.1. Normative References 6
  9.2. Informative References 6
Authors' Addresses 7

1. Introduction

This document specifies the behavior to be followed by the PCP Client [I-D.ietf-pcp-base] to contact its PCP Server(s) [I-D.ietf-pcp-base] when receiving one or several PCP Names (e.g., DHCP [I-D.ietf-pcp-dhcp]). This document is not specific to DHCP; it is applicable to any mechanism that configures server names.

Multiple Names may be configured to a PCP Client in some deployment contexts such as multi-homing. It is out of scope of this document to enumerate all deployment scenarios which require multiple Names to be configured.

This document assumes appropriate name resolution means (e.g., Section 6.1.1 of [RFC1123]) are available on the host client.

2. Terminology

This document makes use of the following terms:

- o PCP Server denotes a functional element which receives and processes PCP requests from a PCP Client. A PCP Server can be co-located with or be separated from the function (e.g., NAT, Firewall) it controls. Refer to [I-D.ietf-pcp-base].
- o PCP Client denotes a PCP software instance responsible for issuing PCP requests to a PCP Server. Refer to [I-D.ietf-pcp-base].
- o Name is a domain name that contains one or more labels. In particular, a PCP name may be structured as DNS qualified name or be composed of strings such as can be passed to getaddrinfo (Section 6.1 of [RFC3493]), including address literals, etc.

3. Name Resolution

Each configured Name is passed to the name resolution library (e.g., Section 6.1.1 of [RFC1123] or [RFC6055]) to retrieve the corresponding IP address(es) (IPv4 or IPv6). Then, the PCP Client MUST follow the procedure specified in Section 4 to contact its PCP Server(s).

A host may have multiple network interfaces (e.g, 3G, WiFi, etc.); each configured differently. Each PCP Server learned MUST be associated with the interface via which it was learned.

4. IP Address Selection

This section specifies the behavior to be followed by the PCP Client to contact its PCP Server(s) when receiving one or several PCP Names:

1. If only one PCP Name is configured: if a list of IP addresses is returned as a result of resolving the PCP Server Name, the PCP Client follows the procedure specified in Section 4.1.
2. If several PCP Names are configured: each Name is treated as a separate PCP Server. Moreover, each Name may be resolved into one IP address or a list of IP addresses. The PCP Client contacts in parallel the first IP address of each Name and follows the procedure specified in Section 4.1 for the list of IP addresses returned for each Name. Section 5 provides some examples to illustrate this procedure.

The discovery procedure may result in a PCP Client instantiating multiple mappings maintained by distinct PCP Servers. The decision to use all these mappings or delete some of them is deployment-specific. Only the client can decide whether all the mappings are needed or only a subset of them.

4.1. Serial Queries

The PCP Client initializes its retransmission timer, RETRY_TIMER, to 2 seconds. The PCP Client sends its PCP message to the PCP Server and waits 2 seconds for a response. If no response is received, it doubles the value of RETRY_TIMER, sends another (identical) PCP message and waits 2*RETRY_TIMER. This procedure is repeated three (3) times, doubling the value of RETRY_TIMER each time. If no response is received after four (4) attempts, the PCP Client tries with the next IP address in its list of PCP Server addresses. If it has exhausted its list, the procedure is repeated every fifteen minutes until the PCP request is successfully answered. If, when sending PCP requests the PCP Client receives an ICMP error (e.g., port unreachable, network unreachable) it SHOULD immediately try the next IP address in the list. Once the PCP Client has successfully received a response from a PCP Server address on that interface, it sends subsequent PCP requests to that same server address until that PCP Server becomes non-responsive, which causes the PCP client to attempt to re-iterate the procedure starting with the first PCP Server address on its list.

5. Examples

The following sub-sections provide three examples to illustrate the procedure.

For all these examples, let’s suppose pcpserver-x, pcpserver-y and pcpserver-z are configured as PCP Names.

5.1. Example 1

Let’s also suppose:

* IPx1 and IPx2 are returned for pcpserver-x; IPx1 is not reachable.
* IPy1 and IPy2 are returned for pcpserver-y; IPy1 is reachable
* IPz1 and IPz2 are returned for pcpserver-z; IPz1 is reachable

The procedure to contact the PCP Servers is as follows:

* Send PCP requests to all servers: IPx1, IPy1 and IPz1
* Responses are received from IPy1 and IPz1 but not from IPx1
  - The request is re-sent to IPx1
  - If no response is received after four attempts, the request is sent to IPx2

5.2. Example 2

Now, if the following conditions are made:

* IPx1 and IPx2 are returned for pcpserver-x; IPx1 is not reachable.
* IPy1 and IPy2 are returned for pcpserver-y; IPy1 is reachable
* IPz1 and IPz2 are returned for pcpserver-z; IPz1 is not reachable

The procedure to contact the PCP Servers lead to the following:

* Send PCP requests to all servers: IPx1, IPy1 and IPz1
* A response is received from IPy1 but not from IPx1 and IPz1
  - the requests are re-sent to IPx1 and IPz1
  - If no response is received after four attempts, the request is then sent to IPx2 and IPz2

5.3. Example 3

Let’s suppose now that:

* IPx1 and IPx2 are returned for pcpserver-x; IPx1 is not reachable.
* IPy1 and IPy2 are returned for pcpserver-y; IPy1 is not reachable
* IPz1 and IPz2 are returned for pcpserver-z; IPz1 is not reachable

The procedure to contact the PCP Servers is as follows:

```
* Send PCP requests to all servers: IPx1, IPy1 and IPz1
* No answer is received for all requests
  - the requests are re-sent to IPx1, IPy1 and IPz1
  - If no response is received after four attempts, the request
    is then sent to IPx2, IPy2 and IPz2
```

6. Security Considerations

The security considerations in [I-D.ietf-pcp-base] are to be considered.

7. IANA Considerations

This document does not request any action from IANA.

8. Acknowledgements

TBC.

9. References

9.1. Normative References

[I-D.ietf-pcp-base]
Wing, D., Cheshire, S., Boucadair, M., Penno, R., and P. Selkirk, "Port Control Protocol (PCP)", draft-ietf-pcp-base-26 (work in progress), June 2012.

[I-D.ietf-pcp-dhcp]
Boucadair, M., Penno, R., and D. Wing, "DHCP Options for the Port Control Protocol (PCP)", draft-ietf-pcp-dhcp-04 (work in progress), August 2012.

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

9.2. Informative References

[RFC1123] Braden, R., "Requirements for Internet Hosts - Application and Support", STD 3, RFC 1123, October 1989.

[RFC3493] Gilligan, R., Thomson, S., Bound, J., McCann, J., and W. Stevens, "Basic Socket Interface Extensions for IPv6", RFC 3493, February 2003.

[RFC6055] Thaler, D., Klensin, J., and S. Cheshire, "IAB Thoughts on Encodings for Internationalized Domain Names", RFC 6055, February 2011.

Authors' Addresses

Mohamed Boucadair
France Telecom
Rennes, 35000
France

Email: mohamed.boucadair@orange.com

Reinaldo Penno
Cisco
USA

Email: repenno@cisco.com

Dan Wing
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, California 95134
USA

Email: dwing@cisco.com

Network Working Group
Internet-Draft
Intended status: Informational
Expires: January 15, 2014

G. Chen
Z. Cao
China Mobile
M. Boucadair
France Telecom
A. Vizdal
Deutsche Telekom AG
L. Thiebaut
Alcatel-Lucent
July 14, 2013

Analysis of Port Control Protocol in Mobile Network
draft-chen-pcp-mobile-deployment-04

Abstract

This memo provides a motivation description for the Port Control Protocol (PCP) deployment in a 3GPP mobile network environment. The document focuses on a mobile network specific issues (e.g. cell phone battery power consumption, keep-alive traffic reduction), PCP applicability to these issues is further studied and analyzed.

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 January 15, 2014.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of

publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction 2
2. Benefits of Introducing PCP in Mobile Networks 3
2.1. Restoring Internet Reachability 3
2.2. Radio Resource Optimization 3
2.3. Energy Saving 4
3. Overviews of PCP Deployment in Mobile Network 4
4. PCP Server Discovery 5
5. MN and multi-homing 6
6. Retransmission Consideration 6
7. Unsolicited Messages Delivery 7
8. SIPTO Architecture 8
9. Authentication Consideration 9
10. Conclusion 9
11. Security Considerations 9
12. IANA Considerations 9
13. Acknowledgements 9
14. References 9
14.1. Normative References 9
14.2. Informative References 10
Authors' Addresses 12

1. Introduction

The Port Control Protocol[RFC6887] allows an IPv6 or IPv4 host to control how incoming IPv6 or IPv4 packets are translated and forwarded by a network address translator (NAT) or simple firewall(FW), and also allows a host to optimize its outgoing NAT keepalive messages. A 3rd Generation Partnership Project (3GPP) network can benefit from the use of the PCP service. Traffic in a mobile network is becoming a complex mix of various protocols, different applications and user behaviors. Mobile networks are currently facing several issues such as a frequent keepalive message, terminal battery consumption and etc. In order to mitigate these issues, PCP could be used to improve terminal behavior by managing how incoming packets are forwarded by upstream devices such as NAT64, NAT44 translators and firewall devices.

It should be noticed that mobile networks have particular characteristics and therefore, there are several factors that should

be investigated before implementing PCP in a mobile context. Without the particular considerations , PCP may not provide desirable outcomes. Some default behaviors may even cause negative impacts or system failures in a mobile environment. Considering very particular environment of mobile networks , it’s needed to have a document describing specific concerns from mobile network side. That would also encourage PCP support in mobile network as well.

This memo covers PCP-related considerations in mobile networks. The intension of publishing this memo is to elaborate major issues during the deployment and share the thoughts for potential usages in mobile networks. Such considerations would provide a pointer to parties interested (e.g. mobile operators) to be included in their UE profile requirements. Some adaptation of PCP protocol might be derived from this document. Such a work would be documented in separated memo(s).

2. Benefits of Introducing PCP in Mobile Networks

2.1. Restoring Internet Reachability

Many Mobile networks are making use of a Firewall to protect their customers from an unwanted Internet originated traffic. The firewall is usually configured to reject all unknown inbound connections and only permit inbound traffic that belongs to a connection initiated from the Firewall or NAT/PAT device. The behavior is described as Category I in [I-D.ietf-opsawg-firewalls].There are applications that can be running on the mobile device that require to be reachable from the Internet or there could be services running behind the terminal that require reachability from the Internet. For example, mobile phones should be able to reachable for instant message or online game. PCP enabled applications / devices could request a port from the Firewall to ensure Internet reachability, and thus would lighten the traffic flow of keep-alive by reducing the number of sending packets. This would result in resource savings on the Firewall node whilst still keeping the customer protected from the unwanted traffic.

2.2. Radio Resource Optimization

3GPP network use different radio channels to transmit control messages(e.g. signaling) and data packages(e.g. voice packages or data flows). Always-on applications, e.g. IM(Instant Message), VoIP or P2P based applications always generate a fair amount of keepalive messages periodically. It’s observed that a number of trivial keepalive messages may occupy the data channel. For example, 16% of traffic caused by instant signaling message would consume 50%~70% radio resource in some area. It likely causes the air congestion with voice calls and service data transmission. PCP could help to

reduce the frequency of periodic messages aimed at refreshing a NAT/ FW binding by indicating to the mobile device the Life time of a binding.

2.3. Energy Saving

Devices with low battery resources exist widely in mobile environments, such as mobile terminals, advanced sensors, etc. mobile terminals often go to "sleep" (IDLE) mode to extend battery life and save air resources. Host initiated message needs to "wake-up" mobile terminals by changing the state to CONNECTED. That would cause more energy on such terminals. Testing reports show that energy consumption is dramatically reduced with prolonged sending interval of signaling messages [VTC2007_Energy_Consumption].

3. Overviews of PCP Deployment in Mobile Network

The Figure 1 shows the architecture of a mobile network. Radio access network would provide wireless connectivity to the MN. Packets are transmitted through Packet Switch(PS) domain heading to MGW. MGW bear the responsibilities of address allocation, routing and transfer. The connection between MN and MGW normally is a point-to- point link, on which MGW is the default router for MN. NAT/ Firewall could either be integrated with MGW or deployed behind MGW as standalone. The traffic is finally destined to application servers, which manage subscriber service.

```
   MN                                                        Internet
    |               +---------------+                     +----------------+
  +-+     +-----+   |            +--|--+     +-------+    |  +----------+  |
  | | /|/ | RAN |---| PS Network | MGW |---- |NAT/FW |----|  |APP Server|  |
  +-+     +-----+   |            +--|--+     +-------+    |  +----------+  |
                    +---------------+                     +----------------+
  MN:  Mobile Nodes
  RAN: Radio Access Network
  PS:  Packet Switch
  MGW:Mobile GateWay
  NAT/FW: Network Address Translator or Firewall
```

Figure 1: Mobile Networks Scenario

A PCP client could be located on MN to control the outbound and inbound traffic on PCP servers. The PCP server is hosted by the NAT/ FW respectively. Corresponding to the various behaviors of PCP client, MN would perform PCP operation using MAP, PEER or ANNOUNCE opcodes. A specific application programming interface may be provided to applications. More discussions and recommendations are presented in following sub-sections.

4. PCP Server Discovery

A straightforward solution seems that MN assume their default router as the PCP Server. However, NAT/FW normally is deployed in a different node than the MGW. Thus there is the need to ensure that MN get information allowing them to discover a PCP server.

[I-D.ietf-pcp-dhcp] specified name options in DHCPv4 and DHCPv6 to discover PCP server. It’s expected the same mechanism could be used in mobile network. 3GPP network allocates IP address and respective parameter during the PDP (Packet Data Protocol)/PDN(Packet Data Network) context activation phase (PDP and PDN represent terminology in 3G and LTE network respectively). On the UE, a PDP/PDN context has same meaning which is equivalent to a network interface.

It should be noted that the Stateful DHCPv6-based address configuration[RFC3315]is not supported by 3GPP specifications. 3GPP adopts IPv6 Stateless Address Auto-configuration (SLAAC) [RFC4861]to allocate IPv6 address. The UE uses stateless DHCPv6[RFC3736] for additional parameter configuration. The MGW acts as the DHCPv6 server. PCP servers discovery could leverage current process to perform the functionalities. The M-bit is set to zero and the O-bit may be set to one in the Router Advertisement (RA) sent to the UE. To carry out PCP sever discovery, a MN should thus send an Information-request message that includes an Option Request Option (ORO) requesting the DHCPv6 PCP Server Name option.

Regarding the IPv4 bearer, MN generally indicates that it prefers to obtain an IPv4 address as part of the PDP context activation procedure. In such a case, the MN relies on the network to provide IPv4 parameters as part of the PDP context activation/ PDN connection set-up procedure. The MN may nevertheless indicate that it prefers to obtain the IPv4 address and configuration parameter after the PDP Context activation by DHCPv4, but it is not available on a wide scale[RFC6459]. MN usually receive those configurations in PCO(Protocol Configuration Options) . PCP server name options in DHCPv4 would not help the PCP servers discovery in that case.

A specific method in 3GPP is to extend PCO [TS24.008]information element to transfer a request of PCP server name. However, additional specification efforts are required in 3GPP to make that happen.

[I-D.cheshire-pcp-anycast]and
[I-D.kiesel-pcp-ip-based-srv-disc]propose anycast-based solutions to discover the closest PCP server on the data path. It may be worth to consider the case when a subscriber roams to different areas, where anycast configurations may be unavailable or operators use other

provisioning method, for example [I-D.ietf-pcp-dhcp]. Asymmetric routing should also be considered in the anycast-based solution. Otherwise, the traffic would likely loses the mapping information for the inbound traffic.

5. MN and multi-homing

As a MN may activate multiple PDP context / PDN connection, it may be multi-homed (the UE receives at least an IP address / an IPv6 prefix per PDN connection). Different MGWs are likely to be associated with each of these PDP context / PDN connection and may thus advertise different PCP servers (using the mechanism described in the previous section). In that case, a MN has to be able to manage multiple PCP servers and to associate an IP flow with the PCP server corresponding to the PDP context / PDN connection used to carry that IP flow.

6. Retransmission Consideration

Mobile devices are usually powered with limited battery . Users would like to use such MN for several days without charging, even several weeks in sensor case. Many applications do not send or receive traffic constantly; instead, the network interface is idle most of the time. That could help to save energy unless there is data leading the link to be activated. Such state changes is based on network-specific timer values corresponding to a number of Radio Resource Control (RRC) states(see more at Section 8.2.2 3GPP[TS23.060]. In order to maximize battery life, it’s desirable that all activities on battery-powered devices needs to be coordinated and synchronized. It’s not specific to PCP. Whereas , those concerns also can be applied to PCP retransmission behavior.

PCP designed retransmission mechanisms on the client for reliable delivery of PCP request. If a PCP client fails to receive an expected response from a server, the client must retransmit its message. The retransmission method may cause unnecessary power consumption when a subscriber roams to a network, in which PCP is not deployed. Several timers are specified to control the retransmission behavior. Therefore, an appropriate implementation and configuration are desirable to help to alleviate the concern. For example, the time transiting to idle is normally less than default Maximum Retransmission Time (MRT), i.e. 1024 seconds. With "no maximum" setting of Maximum Retransmission Duration (MRD), it would cause devices activating their uplink radio in order to retransmit the request messages. Furthermore, the state transition and the transmission take some times, which causes significant power consumption. The MRD should be configured with an optimal time which in line with activated state duration on the device.

The power consumption problem is made complicated if several PCP clients residing on a MN. Several clients are potentially sending requests at random times and by so doing causing MN uplink radio into a significantly power consuming state for unnecessarily often. It’s necessary to perform a synchronization process for tidy up several PCP clients retransmission. A time-line observer maybe required to control different PCP clients resending requests in an optimal transmission window. If the uplink radio of MN is active at the time of sending retransmission from several clients, a proper MRD described as above should be set in a client. If the uplink radio of MN is in idle mode, the time-line observer should hold Initial Retransmission Time(IRT) for while to synchronize different retransmitted PCP requests into same optimal transmission window.

7. Unsolicited Messages Delivery

When the states on NAT/FW have been changed like reboot or changed configuration, PCP servers can send unsolicited messages (e.g. ANNOUNCE messages or unicast PCP MAP or PEER responses) to clients informing them of the new state of their mappings. This aims at achieving rapid detection of PCP failure, rapid PCP recovery or PCP mapping update. When those messages are delivered in a mobile environment, it should be noted multicast delivery may not be available in 3GPP network. PCP servers would use unicast delivery. More considerations are listed as the below.

- o This requires PCP servers to retain knowledge of the IP address(es) and port(s) of their clients, for example using redundancy design based on hot-standby, even though they have rebooted

- o Care should be taken not to generate floods of unicast messages, e.g. to multiple thousands of MN that were served by a PCP server that has rebooted. Such flood may have impacts on Mobile Networks as it may imply the simultaneous generation of Paging process(see more at Section 8.2.4 3GPP[TS23.060]) for very big numbers of MN.

- o Paging function is optionally supported at some particular nodes, e.g. Traffic Offload Function (TOF) in Selected IP Traffic Offload architecture (more discussions on this issues is described in Section 7). The delivery of unsolicited messages would fail in this case.

8. SIPTO Architecture

Since Release 10, 3GPP starts supporting of Selected IP Traffic Offload (SIPTO) function defined in [TS23.060], [TS23.401].The SIPTO function allows an operator to offload certain types of traffic at a network node close to the UE’s point of attachment to the access network. It can be achieved by selecting a set of MGWs that is geographically/topologically close to a UE’s point of attachment. Two variants of solutions has specified in 3GPP.

The mainstream standard deployment relies on selecting a MGW that is geographically/ topologically close to a UE’s point of attachment. This deployment may apply to both 3G and LTE. The MN may sometimes be requested to re-activate its PDP context / PDN connection, in which case it is allocated a new MGW and thus a new IP address and a new PCP server. In this case, host renumbering is inevitable. Some considerations have been described as Address Change Events at Section11.5 of [RFC6887]. The deletions of the mapping information on the old MGW is necessary in order to avoid traffic sending to the old IP address. In a mobile device context, PCP client may take the NAS(Non-Access Stratum) layer message (e.g. "reactivation request" or "detach request" message) as a notification to delete the old mapping information before the subscriber moved to new MGW. Afterwards, PCP clients install new mappings for its new IP address.

As an implementation option dedicated to 3G networks, it is also possible to carry out Selected IP Traffic Offload in a TOF(Traffic Offload Function) entity [TS23.060]located at the interface of the Radio Access Network, i.e. in the path between the radio stations and the Mobile Gateway. The TOF decides on which traffic to offload and enforces NAT for that traffic. The deployment of a TOF is totally transparent for user’s equipments that even cannot know which traffic is subject to TOF (NATed at the TOF) and which traffic is processed by the MGW. The PCP server advertised by the MGW does not take into account the NAT carried out by the TOF function. Therefore, PCP client doesn’t know which PCP servers should be selected to send the request. [I-D.rpcw-pcp-pmipv6-serv-discovery]provides a solution in the similar architecture, in which a PCP proxy with advanced functions[I-D.ietf-pcp-proxy] is required on the offloading point to dispatch requests to a right PCP server. Additional consideration will be given for determining the each traffic flow, since TOF inspects the NAS and RANAP(Radio Access Network Application Part) messages to build the local UE context and local session context. The traffic flow can’t be identified with 5 tuples. The offloaded IP flow is indicated with Radio Access Bearer Identifier (RAB-ID). PCP proxy must understand RAB-ID and map the identifier with each IP flow.

9. Authentication Consideration

The general authentication requirements have been analyzed in [I-D.reddy-pcp-auth-req]. In mobile networks, it is desirable to reuse the existing credentials on the UE for the pcp authentication between involved entities. This way makes the deployment of authentication easiler.

The [I-D.ietf-pcp-authentication] has provided solutions for PCP authentication, in which an EAP option is included in the PCP requests from the devices. In the EAP framework, the EAP authentication server could be the co-located with the PCP server or separated and located on a third-party entity. If the EAP authentication server is placed on the AAA/Radius server, there is a need of an interface between the PCP server and AAA. But per our investigation of 3GPP networks, most exisiting NAT devices do not have such an interface with AAA. So in practical deployment, this could be taken into consideration.

10. Conclusion

PCP mechanism could be potentially adopted in different usage contexts. The deployment in mobile network described applicability analysis, which could give mobile operators a explicit recommendation for PCP implementation. Operators would benefit from such particular considerations. The memo would take the role to document such considerations for PCP deployment in mobile network.

11. Security Considerations

TBD

12. IANA Considerations

This document makes no request of IANA.

13. Acknowledgements

The authors would like to thank Dan Wing, Stuart Cheshire, Ping Lin and Tao Sun for their discussion and comments.

Many thanks to Reinaldo Penno and Tirumaleswar Reddy for their detailed reviews.

14. References

14.1. Normative References

[I-D.ietf-pcp-authentication]
Wasserman, M., Hartman, S., and D. Zhang, "Port Control Protocol (PCP) Authentication Mechanism", draft-ietf-pcp-authentication-01 (work in progress), October 2012.

[I-D.ietf-pcp-dhcp]
Boucadair, M., Penno, R., and D. Wing, "DHCP Options for the Port Control Protocol (PCP)", draft-ietf-pcp-dhcp-07 (work in progress), March 2013.

[I-D.ietf-pcp-proxy]
Boucadair, M., Penno, R., and D. Wing, "Port Control Protocol (PCP) Proxy Function", draft-ietf-pcp-proxy-03 (work in progress), June 2013.

[I-D.reddy-pcp-auth-req]
Reddy, T., Patil, P., Wing, D., and R. Penno, "PCP Authentication Requirements", draft-reddy-pcp-auth-req-04 (work in progress), July 2013.

[RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, November 1987.

[RFC3704] Baker, F. and P. Savola, "Ingress Filtering for Multihomed Networks", BCP 84, RFC 3704, March 2004.

[RFC3736] Droms, R., "Stateless Dynamic Host Configuration Protocol (DHCP) Service for IPv6", RFC 3736, April 2004.

[RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, September 2007.

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

[TS23.060]
, "General Packet Radio Service (GPRS); Service description; Stage 2", June 2012.

[TS23.401]
, "General Packet Radio Service (GPRS) enhancements for Evolved Universal Terrestrial Radio Access Network (E-UTRAN) access", June 2012.

14.2. Informative References

[I-D.cheshire-pcp-anycast]
Cheshire, S., "PCP Anycast Address", draft-cheshire-pcp-anycast-01 (work in progress), March 2013.

[I-D.ietf-opsawg-firewalls]
Baker, F. and P. Hoffman, "On Firewalls in Internet Security", draft-ietf-opsawg-firewalls-01 (work in progress), October 2012.

[I-D.kiesel-pcp-ip-based-srv-disc]
Kiesel, S. and R. Penno, "PCP Server Discovery based on well-known IP Address", draft-kiesel-pcp-ip-based-srv-disc-00 (work in progress), February 2013.

[I-D.rpcw-pcp-pmipv6-serv-discovery]
Reddy, T., Patil, P., Chandrasekaran, R., and D. Wing, "PCP Server Discovery with IPv4 traffic offload for Proxy Mobile IPv6", draft-rpcw-pcp-pmipv6-serv-discovery-02 (work in progress), February 2013.

[RFC3315] Droms, R., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, July 2003.

[RFC6459] Korhonen, J., Soininen, J., Patil, B., Savolainen, T., Bajko, G., and K. Iisakkila, "IPv6 in 3rd Generation Partnership Project (3GPP) Evolved Packet System (EPS)", RFC 6459, January 2012.

[TS24.008]
, "Mobile radio interface Layer 3 specification; Core network protocols; Stage 3", 9.11.0 3GPP TS 24.008, June 2012.

[TS33.220]
, "Generic Authentication Architecture (GAA); Generic Bootstrapping Architecture (GBA)", 10.1.0 3GPP TS 33.220, March 2012.

[VTC2007_Energy_Consumption]
, "Energy Consumption of Always-On Applications in WCDMA Networks", 2007.

Authors' Addresses

Gang Chen
China Mobile
No.32 Xuanwumen West Street
Xicheng District
Beijing 100053
China

Email: phdgang@gmail.com

Zhen Cao
China Mobile
No.32 Xuanwumen West Street
Xicheng District
Beijing 100053
China

Email: caozhen@chinamobile.com

Mohamed Boucadair
France Telecom
No.32 Xuanwumen West Street
Rennes,
35000
France

Email: mohamed.boucadair@orange.com

Vizdal Ales
Deutsche Telekom AG
Tomickova 2144/1
Prague 4, 149 00
Czech Republic

Email: ales.vizdal@t-mobile.cz

Laurent Thiebaut
Alcatel-Lucent

Email: laurent.thiebaut@alcatel-lucent.com

Network Working Group
Internet-Draft
Intended status: Informational
Expires: December 13, 2014

X. Deng
M. Boucadair
France Telecom
Q. Zhao
Beijing University of Posts and Telecommunications
J. Huang
C. Zhou
Huawei Technologies

June 11, 2014

Using Port Control Protocol (PCP) to update dynamic DNS
draft-deng-pcp-ddns-06

Abstract

This document focuses on the problems encountered when using dynamic DNS in address sharing contexts (e.g., DS-Lite, NAT64) during IPv6 transition. Both issues and possible solutions are documented in this memo.

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 December 13, 2014.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents

carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction 2
1.1. Problem Statement 2
1.2. Scope and Goals 3
2. Solution Space 4
2.1. Locate a Service Port 4
2.2. Create Explicit Mappings for Incoming Connections 5
2.3. Detect Changes 5
3. Some Deployment Solutions 6
3.1. Reference Topology 6
3.2. For Web Service 7
3.3. For Non-web Service 8
4. Security Considerations 10
5. IANA Considerations 10
6. Contributors 11
7. Acknowledgements 11
8. References 11
8.1. Normative References 11
8.2. Informative References 11
Authors' Addresses 12

1. Introduction

1.1. Problem Statement

Dynamic DNS (DDNS) is a widely deployed service to facilitate hosting servers (e.g., access to a webcam, HTTP server, FTP server, etc.) at customers' premises. There are a number of providers which offer a DDNS service, working in a client and server mode, which mostly use a web-form based communication. DDNS clients are generally implemented in the user's router or computer, which once detects changes to its assigned IP address it automatically sends an update message to the DDNS server. The communication between the DDNS client and the DDNS server is not standardized, varying from one provider to another, although a few standard web-based methods of updating emerged over time.

When the network architecture evolves towards an IPv4 sharing architecture during IPv6 transition, the DDNS client will have to not only inform the IP address updates if any, but also to notify the changes of external port on which the service is listening, because

well known port numbers, e.g., port 80 will no longer be available to every web server. It will also require the ability to configure corresponding port forwarding on CGN (Carrier Grade NAT, [RFC6888]) devices, so that incoming communications initiated from Internet can be routed to the appropriate server behind the CGN.

Issues encountered in address sharing are documented in [RFC6269]. This document focuses on the problems encountered when using dynamic DNS in address sharing contexts (e.g., DS-Lite [RFC6333], NAT64 [RFC6146]). Below are listed the main challenges:

Announce and Discover an alternate service port: The DDNS service must be able to maintain an alternative port number instead of the default port number.

Allow for incoming connections: Appropriate means to instantiate port mappings in the address sharing device must be supported.

Detect changes and trigger DDNS updates: DDNS client must be triggered by the change of the external IP address and the port number. Concretely, upon change of the external IP address (and/or external port number), the DDNS client must refresh the DNS records otherwise the server won’t be reachable from outside. This issue is exacerbated in the DS-Lite context because no public IPv4 address is assigned to the CPE.

1.2. Scope and Goals

This document describes some candidate solutions to resolve the aforementioned issues with a particular focus on DS-Lite. These solutions may also be valid for other address sharing schemes.

This document sketches deployment considerations based on the PCP (Port Control Protocol, [RFC6887]). Note DDNS may be considered as an implementation of the Rendezvous service mentioned in [RFC6887].

Indeed, after creating an explicit mapping for incoming connections using PCP, it is necessary to inform remote hosts about the IP address, protocol, and port number for the incoming connection to reach the services hosted behind a DS-Lite CGN. This is usually done in an application-specific manner. For example, a machine hosting a game server might use a rendezvous server specific to that game (or specific to that game developer), a SIP phone would use a SIP proxy, and a client using DNS-Based Service Discovery [RFC6763] would use DNS Update [RFC2136][RFC3007], etc. PCP does not provide this rendezvous function.

The rendezvous function may support IPv4, IPv6, or both. Depending on that support and the application's support of IPv4 or IPv6, the PCP client may need an IPv4 mapping, an IPv6 mapping, or both. An example illustrating how the DDNS server may implement such a service notification functionality if necessary is provided in Section 3.

This document does not specify any protocol extension, but instead it focuses on the elaboration of the problem space and illustrate how existing tools can be re-used to solve the problem for some deployment contexts. Particularly, this document requires no changes to PCP or dynamic updates in the standard domain name system [RFC2136], but is rather an operational document to make the current DDNS service providers be aware of the impacts and issues that the IPv6 transitioning and IPv4 address sharing will bring to them, and gives solutions address the forthcoming issues. The current DDNS service providers usually employs a web-based form to maintain DDNS service registration and updates.

Generic deployment considerations for DS-Lite, including B4 remote management and IPv4 connectivity check, can be found in [RFC6908]. This document complements [RFC6908] with deployment considerations related to Rendezvous service maintenance. Additional PCP-related deployment considerations are available at [I-D.boucadair-pcp-deployment-cases].

Solutions relying on DNS-based Service Discovery [RFC6763] or Apple's Back to My Mac (BTMM) Service [RFC6281] are not considered in this document. Moreover, this document does not assume that DDNS service relies on [RFC2136].

IPv4 addresses used in the examples are derived from the IPv4 block reserved for documentation in [RFC6890]. DNS name examples follow [RFC2606].

2. Solution Space

2.1. Locate a Service Port

As listed below, at least two solutions can be used to associate a port number with a service:

1. Use service URIs (e.g., FTP, SIP, HTTP) which embed an explicit port number. Indeed, Uniform Resource Identifier (URI) defined in [RFC3986] allows to carry port number in the syntax (e.g., mydomain.example:15687).

2. Use SRV records [RFC2782]. Unfortunately, the majority of browsers do not support this record type.

DDNS client and DDNS server are to be updated so that an alternate port number is signaled and stored by the DDNS server. Requesting remote hosts will be then notified with the IP address and port number to reach the server.

2.2. Create Explicit Mappings for Incoming Connections

PCP is used to install the appropriate mapping(s) in the CGN so that incoming packets can be delivered to the appropriate server.

2.3. Detect Changes

In a network described in Figure 1, DDNS client/ PCP client can either be running on a Customer Premise Equipment (CPE) or be running on the host that is hosting some services itself. There are several possible ways to address the problems stated in Section 1.1:

1. If the DDNS client is enabled, the host issues periodically (e.g., 60 minutes) PCP MAP requests (e.g., messages 1 and 2 in Figure 1) with short lifetime (e.g., 30s) for the purpose of enquiring external IP address and setting. If the purpose is to detect any change of external port, the host must issues a PCP mapping to install a mapping for the internal server. Upon change of the external IP address, the DDNS client updates the records accordingly (e.g., message 3 in Figure 1).

2. If the DDNS client is enabled, it checks the local mapping table maintained by the PCP client. This process is repeated periodically (e.g., 5 minutes, 30 minutes, 60 minutes). If there is no PCP mapping created by PCP client, it issues a PCP MAP request (e.g., messages 1 and 2 in Figure 1) for the purpose of enquiring external IP address and setting up port forwarding mappings for incoming connections. Upon change of the external IP address, the DDNS client updates the records in the DDNS server, e.g., message 3 in Figure 1.

```
                                        +-----------------+
                                        |  DDNS Server    |
                                        +-----------------+
                                                  ^
                                                  |
                                                  |3. DDNS updates
                                                  |  (if any)
                                                  |
 +---------------+                     +-----------------+
 |DDNS Client    |1. PCP MAP request  | CGN/PCP Server  |
 |PCP Client/IWF |------------------->| (PCP mapping for|80:8080+------+
 |on CPE or      |2. PCP MAP response | port forwarding)|<------|Client|
 |the host itself|<-------------------|                 |       +------+
 |               |3. DDNS updates     |                 |
 |               |     (if any)       |                 |
 |               |------------------->|                 |
 +---------------+                     +-----------------+
```

Figure 1: Flow Chart

3. Some Deployment Solutions

3.1. Reference Topology

Figure 2 illustrates the topology used for the deployment solutions elaborated in the following sub-sections.

```
+--------------+   +--------+    +---------+   +--------+   +-------+
| Service      |   |  DDNS  |    |  CGN&   |   | PCP    |   |Servers|
| User         |---|  Server|----|  PCP    |---| Client |---|       |
|              |   |        |    |  Server |   | /DDNS  |   |       |
|              |   |        |    |         |   | client |   |       |
+--------------+   +--------+    +---------+   +--------+   +-------+
A user DDNS Server AFTR B4(CPE) A host

From Internet                                                behind B4
```

Figure 2: Implementation Topology

Figure 2 involves of the following entities:

- o Servers: refer to the servers that are deployed in the DS-Lite network, or more generally, an IP address sharing environment. They are usually running on a host that has been assigned with a private IPv4 address. Having created a proper mapping via PCP in AFTR, these services have been made available to the Internet users. The services may provide Web, FTP, SIP and other services though these ones may not be able to been seen as using a well

known port from the outside anymore, in the IP address sharing context.

- o B4 (CPE): An endpoint of IPv4-in-v6 tunnel [RFC6333]. A PCP client together with a DDNS client are running on it. After PCP client establishes a mapping on the AFTR, an end user may register its domain name and its external IPv4 address plus port number to its DDNS service provider (DDNS server), manually or automatically by DDNS client. Later, likewise, end users may manually or let DDNS client on behalf of it, to automatically announce IP address and/or port changes to the DDNS server.

- o AFTR: Responsible for maintaining mappings between internal IPv4 Address plus port and external IPv4 address plus port [RFC6333].

- o DDNS server: Maintains a table that associates a registered domain name and a pair of registered host’s external IPv4 address plus port number. When being notified IP address and port number changes from DDNS client, DDNS server announces the updates to DNS servers on behalf of end user. [RFC2136] and [RFC3007] may be used by DDNS server to send updates to DNS servers. In many current practices, DDNS server provider usually announce its own IP address as the registered domain names of end users. When HTTP requests reach the DDNS server, they may employ URL Forwarding or HTTP 301 redirection to redirect the request to a proper registered end user by looking up the maintained link table.

- o Service users: refers to users who want to access services behind an IP address sharing network. They issue standard DNS requests to locate the services, which will lead them to a DDNS server, provided that the requested services have been registered to a DDNS service provider. The DDNS server will then handle the rest in the way as described before.

3.2. For Web Service

Current DDNS server implementations typically assume that the end servers host web server on the default 80 port. In the DS-Lite context, they will have to take into account that external port assigned by AFTR may be any number other than 80, in order to maintain proper mapping between domain names and external IP plus port. By doing such changes, the HTTP request would be redirected to the AFTR which servers the specific end host that are running servers.

Figure 3 depicts how messages are handled in order to be delivered to the right server.

```
   Web Visitor        DDNS server       AFTR      B4(CPE)     Web Server
                                                               behind B4
   | HTTP Get*             |              |          |               |
   |----------------------->|              |          |               |
   | ip_DDNS_server        |------------->|          |               |
   |                       | HTTP 301     |          |               |
   |                       |<-------------|          |               |
   | HTTP Get* ip_aftr:8001               |          |               |
   |------------------------------------->|                          |
   |                                      | HTTP Get* ip_websrv:8000 |
   |                                      |------------------------->|
   |                                      |                          |
   |                       HTTP response  | HTTP response            |
   |<-------------------------------------|--------------------------|
   |                                      |                          |
```

Figure 3: Http Service Messages

When a web user sends out a HTTP GET message to DDNS server after a standard DNS query, DDNS server redirects the request to a registered web server, in this case, by responding with a HTTP 301 message. Then, the HTTP GET message will be sent out to the AFTR, which will in turn finds the proper hosts behind it. For simplicity, messages among AFTR, B4 and web server behind B4 are not shown completely; for communications among those nodes, refer to [RFC6333].

3.3. For Non-web Service

For non-web services, as mentioned in Section 2, other means will be needed to inform the users about the service information.

[RFC6763] includes an example of DNS-based solution which allows an application running in the end user’s device to retrieve service-related information via DNS SRV/TXT records, and list available services. In a scenario where such application is not applicable, following provides another solution for a third party, e.g., DDNS service provider, to disclose services to the Internet users.

A web portal can be used to list available services. DDNS server maintains a web portal for each user FQDN (Fully Qualified Domain Name), which provides users service links. Figure 4 assumes "websrv.example.com" is a user’s FQDN provided by a DDNS service provider.

```
   +-------------+    +-------------+    +----------+ Internet +-------+
   |DDNS client /|    |DDNS server /|    |DNS server|          |Visitor|
   |  Web Server |    | web portal  |    |          |          |       |
   +-------------+    +-------------+    +----------+          +-------+
       |      register      |                 |                    |
       |<------------------>|                 |                    |
       | websrv.example.com |  update DNS     |                    |
       |   192.0.2.1:2000   | <-------------> |                    |
       |                    |websrv.example.com|                   |
       |                    |   portal's IP   |                    |
       |              +-------------+         |                    |
       |              |update portal|         |                    |
       |              +-------------+         |  DNS resolve for   |
       |                    |                 | <----------------> |
       |                    |                 | websrv.example.com |
       |                    |                 |  get portal's IP   |
       |                    |                 |                    |
       |                    |   visit portal of websrv.example.com |
       |                    | <----------------------------------> |
       |                    |                 |                    |
       |                 visit http://192.0.2.1:2000              |
       | <-------------------------------------------------------->|
       |                    |                 |                    |
```

Figure 4: Update Web Portal

The DDNS client registers the servers' information to the DDNS server, including public IP address and port obtained via PCP, user's FQDN and other necessary information. The DDNS server also behaves as portal server, it registers its IP address, port number, and user's FQDN to the DNS system, so that visitors can access the web portal.

DDNS server also maintains a web portal for each user's FQDN, update the portal according to registered information from DDNS client. When a visitor accesses "websrv.example.com", a DNS query will resolve to portal server's address, port number, and the visitor will see the portal and the available services.

```
+-------------------------------------------------------------+
|                                                             |
|              Portal: websrv.example.com                     |
|                                                             |
|    Service1: web server                                     |
|    Link:     http://192.0.2.1:2000                          |
|                                                             |
|    Service2: video                                          |
|    Link:     rtsp://192.0.2.1:8080/test.sdp                 |
|                                                             |
|    ......                                                   |
|                                                             |
+-------------------------------------------------------------+
```

Figure 5: An Example of Web Portal

As shown in Figure 5, the web portal shows the service URLs that are available to be accessed. Multiple services are accessible per user's FQDN.

Some applications which are not HTTP-based can also be delivered using this solution. When a user clicks on a link, the registered application in the client OS will be invoked to handle the link. How this can be achieved is out of the scope of this document.

4. Security Considerations

This document does not introduce a new protocol nor specify protocol extensions. Security-related considerations related to PCP [RFC6887] and DS-Lite [RFC6333] should be taken into account.

The protocol between the DDNS client and DDNS server is proprietary in most cases, some extensions may be necessary, which is up to DDNS operators. These operators should enforce security-related policies to avoid that illegitimate users alter records installed by legitimate users or install fake records that would lead to attract illegitimate traffic. Means to protect the DDNS server against DoS (Denial of Service) should be enabled. Note these considerations are not specific to address sharing contexts but are valid for DDNS service in general.

5. IANA Considerations

This document does not require any action from IANA.

6. Contributors

The following individuals contributed text to the document:

Xiaohong Huang

Beijing University of Posts and Telecommunications, China
Email: huangxh@bupt.edu.cn

Yan Ma

Beijing University of Posts and Telecommunications, China
Email: mayan@bupt.edu.cn

7. Acknowledgements

Thanks to Stuart Cheshire for bringing up DNS-Based Service Discovery and [RFC6281] where covers DNS-based SD scenario and gives an example of how the application means of solution to address dynamic DNS update, in this case, apple’ BTMM, can be achieved.

Many thanks to D. Wing, D. Thaler, and J. Abley for their comments.

8. References

8.1. Normative References

[RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, January 2005.

[RFC6333] Durand, A., Droms, R., Woodyatt, J., and Y. Lee, "Dual-Stack Lite Broadband Deployments Following IPv4 Exhaustion", RFC 6333, August 2011.

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

8.2. Informative References

[I-D.boucadair-pcp-deployment-cases] Boucadair, M., "Port Control Protocol (PCP) Deployment Models", draft-boucadair-pcp-deployment-cases-02 (work in progress), April 2014.

[RFC2136] Vixie, P., Thomson, S., Rekhter, Y., and J. Bound, "Dynamic Updates in the Domain Name System (DNS UPDATE)", RFC 2136, April 1997.

[RFC2606] Eastlake, D. and A. Panitz, "Reserved Top Level DNS Names", BCP 32, RFC 2606, June 1999.

[RFC2782] Gulbrandsen, A., Vixie, P., and L. Esibov, "A DNS RR for specifying the location of services (DNS SRV)", RFC 2782, February 2000.

[RFC3007] Wellington, B., "Secure Domain Name System (DNS) Dynamic Update", RFC 3007, November 2000.

[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.

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

[RFC6281] Cheshire, S., Zhu, Z., Wakikawa, R., and L. Zhang, "Understanding Apple’s Back to My Mac (BTMM) Service", RFC 6281, June 2011.

[RFC6763] Cheshire, S. and M. Krochmal, "DNS-Based Service Discovery", RFC 6763, February 2013.

[RFC6888] Perreault, S., Yamagata, I., Miyakawa, S., Nakagawa, A., and H. Ashida, "Common Requirements for Carrier-Grade NATs (CGNs)", BCP 127, RFC 6888, April 2013.

[RFC6890] Cotton, M., Vegoda, L., Bonica, R., and B. Haberman, "Special-Purpose IP Address Registries", BCP 153, RFC 6890, April 2013.

[RFC6908] Lee, Y., Maglione, R., Williams, C., Jacquenet, C., and M. Boucadair, "Deployment Considerations for Dual-Stack Lite", RFC 6908, March 2013.

Authors’ Addresses

Xiaohong Deng

Email: dxhbupt@gmail.com

Mohamed Boucadair
France Telecom
Rennes 35000
France

Email: mohamed.boucadair@orange.com

Qin Zhao
Beijing University of Posts and Telecommunications
China

Email: zhaoqin.bupt@gmail.com

James Huang
Huawei Technologies
China

Email: james.huang@huawei.com

Cathy Zhou
Huawei Technologies
China

Email: cathy.zhou@huawei.com

Network Working Group
Internet-Draft
Updates: 6887 (if approved)
Intended status: Standards Track
Expires: January 21, 2016

M. Wasserman
S. Hartman
Painless Security
D. Zhang
Huawei
T. Reddy
Cisco

July 20, 2015

Port Control Protocol (PCP) Authentication Mechanism
draft-ietf-pcp-authentication-14

Abstract

An IPv4 or IPv6 host can use the Port Control Protocol (PCP) to flexibly manage the IP address and port mapping information on Network Address Translators (NATs) or firewalls to facilitate communication with remote hosts. However, the un-controlled generation or deletion of IP address mappings on such network devices may cause security risks and should be avoided. In some cases the client may need to prove that it is authorized to modify, create or delete PCP mappings. This document describes an in-band authentication mechanism for PCP that can be used in those cases. The Extensible Authentication Protocol (EAP) is used to perform authentication between PCP devices.

This document updates RFC6887.

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 January 21, 2016.

Copyright Notice

Copyright (c) 2015 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction 3
2. Terminology 4
3. Protocol Details 5
3.1. Session Initiation 5
3.1.1. Authentication triggered by the client 6
3.1.2. Authentication triggered by the server 7
3.1.3. Authentication using EAP 7
3.2. Recovery from lost PA session 9
3.3. Session Termination 10
3.4. Session Re-Authentication 11
4. PA Security Association 12
5. Packet Format 13
5.1. Packet Format of PCP Auth Messages 13
5.2. Opcode-specific information of AUTHENTICATION Opcode . . 15
5.3. NONCE Option 16
5.4. AUTHENTICATION_TAG Option 16
5.5. PA_AUTHENTICATION_TAG option 18
5.6. EAP_PAYLOAD Option 19
5.7. PRF Option 19
5.8. MAC_ALGORITHM Option 20
5.9. SESSION_LIFETIME Option 20
5.10. RECEIVED_PAK Option 21
5.11. ID_INDICATOR Option 21
6. Processing Rules 22
6.1. Authentication Data Generation 22
6.2. Authentication Data Validation 23
6.3. Retransmission Policies for PA Messages 24
6.4. Sequence Numbers for PCP Auth Messages 24
6.5. Sequence Numbers for Common PCP Messages 25
6.6. MTU Considerations 26
7. IANA Considerations 27

7.1. NONCE 28
7.2. AUTHENTICATION_TAG 28
7.3. PA_AUTHENTICATION_TAG 28
7.4. EAP_PAYLOAD 29
7.5. PRF 29
7.6. MAC_ALGORITHM 29
7.7. SESSION_LIFETIME 30
7.8. RECEIVED_PAK 30
7.9. ID_INDICATOR 30
8. Security Considerations 31
9. Acknowledgements 31
10. Change Log 32
10.1. Changes from wasserman-pcp-authentication-02 to ietf-pcp-authentication-00 32
10.2. Changes from wasserman-pcp-authentication-01 to -02 . . 32
10.3. Changes from ietf-pcp-authentication-00 to -01 32
10.4. Changes from ietf-pcp-authentication-01 to -02 32
10.5. Changes from ietf-pcp-authentication-02 to -03 33
10.6. Changes from ietf-pcp-authentication-03 to -04 33
10.7. Changes from ietf-pcp-authentication-04 to -05 33
10.8. Changes from ietf-pcp-authentication-05 to -06 33
11. References 34
11.1. Normative References 34
11.2. Informative References 35
Authors' Addresses 35

1. Introduction

Using the Port Control Protocol (PCP) [RFC6887], an application can flexibly manage the IP address mapping information on its network address translators (NATs) and firewalls, and control their policies in processing incoming and outgoing IP packets. Because NATs and firewalls both play important roles in network security architectures, there are many situations in which authentication and access control are required to prevent un-authorized users from accessing such devices. This document defines a PCP security extension that enables PCP servers to authenticate their clients with Extensible Authentication Protocol (EAP). The EAP messages are encapsulated within PCP messages during transportation.

The following issues are considered in the design of this extension:

- Loss of EAP messages during transportation

- Reordered delivery of EAP messages

- Generation of transport keys

- o Integrity protection and data origin authentication for PCP messages

- o Algorithm agility

The mechanism described in this document meets the security requirements to address the Advanced Threat Model described in the base PCP specification [RFC6887]. This mechanism can be used to secure PCP in the following situations:

- o On security infrastructure equipment, such as corporate firewalls, that do not create implicit mappings for specific traffic.

- o On equipment (such as CGNs or service provider firewalls) that serve multiple administrative domains and do not have a mechanism to securely partition traffic from those domains.

- o For any implementation that wants to be more permissive in authorizing applications to create mappings for successful inbound communications destined to machines located behind a NAT or a firewall.

2. Terminology

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 RFC 2119 [RFC2119].

Most of the terms used in this document are introduced in [RFC6887].

PCP Client: A PCP software instance that is responsible for issuing PCP requests to a PCP server. In this document, a PCP client is also a EAP peer [RFC3748], and it is the responsibility of a PCP client to provide the credentials when authentication is required.

PCP Server: A PCP software instance that resides on the PCP-Controlled Device that receives PCP requests from the PCP client and creates appropriate state in response to that request. In this document, a PCP server is integrated with an EAP authenticator [RFC3748]. Therefore, when necessary, a PCP server can verify the credentials provided by a PCP client and make an access control decision based on the authentication result.

PCP-Authentication (PA) Session: A series of PCP message exchanges transferred between a PCP client and a PCP server. The PCP messages involved within a session includes the PA messages used to perform EAP authentication, key distribution and session management, and the common PCP messages secured with the keys distributed during

authentication. Each PA session is assigned a distinctive Session ID.

Session Partner: A PCP implementation involved within a PA session. Each PA session has two session partners (a PCP server and a PCP client).

PCP device: A PCP client or a PCP server.

Session Lifetime: The lifetime associated with a PA session, which decides the lifetime of the current authorization given to the PCP client.

PCP Security Association (PCP SA): A PCP security association is formed between a PCP client and a PCP server by sharing cryptographic keying material and associated context. The formed duplex security association is used to protect the bidirectional PCP signaling traffic between the PCP client and PCP server.

Master Session Key (MSK): A key derived by the partners of a PA session, using an EAP key generating method (e.g., the one defined in [RFC5448]).

PCP-Authentication (PA) message: A PCP message containing an AUTHENTICATION Opcode. Particularly, a PA message sent from a PCP server to a PCP client is referred to as a PA-Server message, while a PA message sent from a PCP client to a PCP server is referred to as a PA-Client message. Therefore, a PA-Server message is actually a PCP response message specified in [RFC6887], and a PA-Client message is a PCP request message. This document specifies an option, the PA_AUTHENTICATION_TAG Option defined in Section 5.5 for PCP authentication, to provide integrity protection and message origin authentication for PA messages.

Common PCP message: A PCP message which does not contain an AUTHENTICATION Opcode. This document specifies an AUTHENTICATION_TAG Option to provide integrity protection and message origin authentication for the common PCP messages.

3. Protocol Details

3.1. Session Initiation

At the beginning of a PA session, a PCP client and a PCP server need to exchange a series of PA messages in order to perform an EAP authentication process. Each PA message MUST contain an AUTHENTICATION Opcode and may optionally contain a set of Options for various purposes (e.g., transporting authentication messages and

session management). The opcode-specific information in a AUTHENTICATION Opcode consists of two fields : Session ID and Sequence Number. The Session ID field is used to identify the PA session to which the message belongs. The sequence number field is used to detect whether reordering or duplication occurred during message delivery.

3.1.1. Authentication triggered by the client

When a PCP client intends to proactively initiate a PA session with a PCP server, it sends a PA-Initiation message (a PA-Client message with the result code "INITIATION") to the PCP server. Section 5.1 updates the PCP request message format with result codes for the PCP Authentication mechanism. In the opcode-specific information of the message, the Session ID and Sequence Number fields are set as 0. The PA-Client message MUST also contain a NONCE option defined in Section 5.3 which consists of a random nonce.

After receiving the PA-Initiation, if the PCP server agrees to initiate a PA session with the PCP client, it will reply with a PA-Server message which contains an EAP Request and the result code field of this PA-Server message is set to AUTHENTICATION_REQUEST. In addition, the server MUST assign a unique session identifier to distinctly identify this session, and fill the identifier into the Session ID field in the opcode-specific information of the PA-Server message. The Sequence Number field of the message is set as 0. The PA-Server message MUST contain a NONCE option so as to send the nonce value back. The nonce will then be used by the PCP client to check the freshness of this message. Subsequent PCP messages within this PA session MUST contain this session identifier.

```
  PCP                                                PCP
  client                                            server
    |-- PA-Initiation--------------------------------->|
    |   (Seq=0, rc=INITIATION, Session ID=0)           |
    |                                                  |
    |<-- PA-Server -----------------------------------|
    |    (Seq=0, Session ID=X, EAP request,            |
    |     rc=AUTHENTICATION_REQUEST)                   |
    |                                                  |
    |-- PA-Client ------------------------------------>|
    |    (Seq=1, Session ID=X, EAP response,           |
    |     rc=AUTHENTICATION_REPLY)                     |
    |                                                  |
    |<-- PA-Server -----------------------------------|
    |    (Seq=1, Session ID=X, EAP request,            |
    |     rc=AUTHENTICATION_REQUEST)                   |
```

3.1.2. Authentication triggered by the server

In the scenario where a PCP server receives a common PCP request message from a PCP client which needs to be authenticated, the PCP server rejects the request with a AUTHENTICATION_REQUIRED error code and can reply with a unsolicited PA-Server message to initiate a PA session. The result code field of this PA-Server message is set to AUTHENTICATION_REQUEST. In addition, the PCP server MUST assign a Session ID for the session and transfer it within the PA-Server message. The Sequence Number field in the PA-Server message is set as 0. If the PCP client retries the common request before EAP authentication is successful then it will receive AUTHENTICATION_REQUIRED error code from the PCP server. In the PA messages exchanged afterwards in this session, the Session ID will be used in order to help session partners distinguish the messages within this session from those not within. When the PCP client receives this initial PA-Server message from the PCP server, it can reply with a PA-Client message or silently discard the request message according to its local policies. In the PA-Client message, a NONCE option which consists of a random nonce MAY be appended. If so, in the next PA-Server message, the PCP server MUST forward the nonce back within a NONCE option.

```
PCP                                                PCP
client                                            server
  |-- Common PCP request--------------------------->|
  |                                                 |
  |<- Common PCP response---------------------------|
  |   rc=AUTHENTICATION_REQUIRED)                   |
  |                                                 |
  |<-- PA-Server -----------------------------------|
  |    (Seq=0, Session ID=X, EAP request)           |
  |     rc=AUTHENTICATION_REQUEST)                  |
  |                                                 |
  |-- PA-Client ------------------------------------>|
  |    (Seq=0, Session ID=X, EAP response)          |
  |     rc=AUTHENTICATION_REPLY)                    |
  |                                                 |
  |<-- PA-Server -----------------------------------|
  |    (Seq=1, Session ID=X, EAP request,           |
  |     rc=AUTHENTICATION_REQUEST)                  |
```

3.1.3. Authentication using EAP

In a PA session, an EAP request message is transported within a PA-Server message and an EAP response message is transported within a PA-Client message. EAP relies on the underlying protocol to provide

reliable transmission; any reordered delivery or loss of packets occurring during transportation must be detected and addressed. Therefore, after sending out a PA-Server message, the PCP server will not send a new PA-Server message in the same PA session until it receives a PA-Client message with a proper sequence number from the PCP client, and vice versa. If a PCP client receives a PA message containing an EAP request and cannot generate an EAP response immediately due to certain reasons (e.g., waiting for human input to construct a EAP message or due to EAP message fragmentation waiting for the additional PA messages in order to construct a complete EAP message), the PCP device MUST reply with a PA-Acknowledgement message (PA message with a RECEIVED_PAK Option) to indicate that the message has been received. This approach not only can avoid unnecessary retransmission of the PA message but also can guarantee the reliable message delivery in conditions where a PCP device needs to receive multiple PA messages carrying the fragmented EAP request before generating an EAP response. The number of EAP messages exchanged between the PCP client and PCP server depends on the EAP method used for authentication.

In this approach, PCP client and a PCP server MUST perform a key-generating EAP method in authentication. Particularly, a PCP authentication implementation MUST support EAP-TTLS [RFC5281] and SHOULD support TEAP [RFC7170]. Therefore, after a successful authentication procedure, a Master Session Key (MSK) will be generated. If the PCP client and the PCP server want to generate a transport key using the MSK, they need to agree upon a Pseudo-Random Function (PRF) for the transport key derivation and a MAC algorithm to provide data origin authentication for subsequent PCP messages. In order to do this, the PCP server needs to append a set of PRF Options and MAC_ALGORITHM Options to the initial PA-Server message. Each PRF Option contains a PRF that the PCP server supports, and each MAC_ALGORITHM Option contains a MAC (Message Authentication Code) algorithm that the PCP server supports. Moreover, in the first PA-Server message, the server MAY also attach an ID_INDICATOR Option defined in Section 5.11 to direct the client to choose correct credentials. After receiving the options, the PCP client MUST select the PRF and the MAC algorithm which it would like to use, and then adds the associated PRF and MAC Algorithm Options to the next PA-Client message.

After the EAP authentication, the PCP server sends out a PA-Server message to indicate the EAP authentication and PCP authorization results. If the EAP authentication succeeds, the result code of the PA-Server message is AUTHENTICATION_SUCCEEDED. In this case, before sending out the PA-Server message, the PCP server MUST update the PCP SA with the MSK and transport key, and use the derived transport key to generate a digest for the message. The digest is transported

within an PA_AUTHENTICATION_TAG Option for PCP Auth. A more detailed description of generating the authentication data can be found in Section 6.1. In addition, the PA-Server message MUST also contain a SESSION_LIFETIME Option defined in Section 5.9 which indicates the lifetime of the PA session (i.e., the lifetime of the MSK). After receiving the PA-Server message, the PCP client then needs to generate a PA-Client message as response. If the PCP client also authenticates the PCP server, the result code of the PA-Client message is AUTHENTICATION_SUCCEEDED. In addition, the PCP client needs to update the PCP SA with the MSK and transport key, and uses the derived transport key to secure the message. From then on, all the PCP messages within the session are secured with the transport key and the MAC algorithm specified in the PCP SA. The first secure PA-client message from the client MUST include the set of PRF and MAC_ALGORITHM options received from the PCP server. The PCP server determines if the set of algorithms conveyed by the client matches the set it had initially sent, to detect an algorithm downgrade attack. If the server detects a downgrade attack then it MUST send a PA-Server message with result code DOWNGRADE_ATTACK_DETECTED and terminate the session. If the PCP client sends common PCP request within the PA session without AUTHENTICATION_TAG option then the PCP server rejects the request by returning AUTHENTICATION_REQUIRED error code.

If a PCP client/server cannot authenticate its session partner, the device sends out a PA message with the result code, AUTHENTICATION_FAILED. If the EAP authentication succeeds but authorization fails, the device making the decision sends out a PA message with the result code, AUTHORIZATION_FAILED. In these two cases, after the PA message is sent out, the PA session MUST be terminated immediately. It is possible for independent PCP clients on the host to create multiple PA sessions with the PCP server.

3.2. Recovery from lost PA session

If a PCP server resets or loses the PCP SA due to reboot, power failure, or any reason then it sends unsolicited ANNOUNCE response as explained in section 14.1.3 of [RFC6887] to the PCP client. Upon receiving the ANNOUNCE response with an anomalous Epoch time, PCP client deduces that the server may have lost state. The ANNOUNCE is either bogus (an attack), legitimate, or not seen by the client. These three cases are described below:

- o PCP client sends integrity-protected unicast ANNOUNCE request to the PCP server to check if the PCP server has indeed lost the state or an attacker has sent the ANNOUNCE response.

  * If integrity-protected success response is recevied from the PCP server then the PCP client determines that the PCP server has not lost the PA session, and the unsolicited ANNOUNCE response was sent by an attacker.

  * If the PCP server responds to the ANNOUNCE request with UNKNOWN_SESSION_ID error code then the PCP client MUST initiate full EAP authentication with the PCP server as explained in Section 3.1.1. After EAP authentication is successful PCP client updates the PCP SA and issues new common PCP requests to recreate any lost mapping state.

- o In a scenario where the PCP server has lost the PCP SA but did not inform the PCP client, if the PCP client sends PCP request integrity-protected then the PCP server rejects the request with UNKNOWN_SESSION_ID error code. The PCP client then initiates full EAP authentication with the PCP server as explained in Section 3.1.1 and updates the PCP SA after successful authentication.

If the PCP client resets or loses the PCP SA due to reboot, power failure, or any reason and sends common PCP request then the PCP server rejects the request with AUTHENTICATION_REQUIRED error code. The PCP client MUST authenticate with the PCP server and after EAP authentication is successful retry the common PCP request with AUTHENTICATION_TAG option. The PCP server MUST update the PCP SA after successful EAP authentication.

3.3. Session Termination

A PA session can be explicitly terminated by either session partner. A PCP Server may explicitly request termination of the session by sending an unsolicited termination-indicating PA response (a PA response with a result code "SESSION-TERMINATED"). Upon receiving a termination-indicating message, the PCP client MUST respond with a termination-indicating PA message, and MUST then remove the associated PCP SA. To accommodate packet loss, the PCP server MAY transmit the termination-indicating PA response up to ten times (with an appropriate Epoch Time value in each to reflect the passage of time between transmissions) provided that the interval between the first two notifications is at least 250 ms, and the interval between subsequent notification at least doubles.

A PCP client may explicitly request termination of the session by sending a termination-indicating PA request (a PA request with a result code "SESSION-TERMINATED"). After receiving a termination-indicating message from the PCP client, a PCP server MUST respond with a termination-indicating PA response and remove the PCP SA

immediately. When the PCP client receives the termination-indicating PA response, it MUST remove the associated PCP SA immediately.

3.4. Session Re-Authentication

A session partner may select to perform EAP re-authentication if it would like to update the PCP SA without initiating a new PA session. For example a re-authentication procedure could be triggered for the following reasons:

- o The session lifetime needs to be extended.

- o The sequence number is going to reach the maximum value. Specifically, when the sequence number reaches 2**32 - 2**16, the session partner MUST trigger re-authentication.

When the PCP server would like to initiate a re-authentication, it sends the PCP client a PA-Server message. The result code of the message is set to "RE-AUTHENTICATION", which indicates the message is for a re-authentication process. If the PCP client would like to start the re-authentication, it will send a PA-Client message to the PCP server, with the result code of the PA-Client message set to "RE-AUTHENTICATION". Then, the session partners exchange PA messages to transfer EAP messages for the re-authentication. During the re-authentication procedure, the session partners protect the integrity of PA messages with the key and MAC algorithm specified in the current PCP SA; the sequence numbers associated with the message will continue to keep increasing according to Section 6.3. The result code for PA-Sever message carrying EAP request will be set to AUTHENTICATION_REQUIRED and PA-Client message carrying EAP response will be set to AUTHENTICATION_REPLY.

If the EAP re-authentication succeeds, the result code of the last PA-Server message is "AUTHENTICATION_SUCCEEDED". In this case, before sending out the PA-Server message, the PCP server MUST update the SA and use the new key to generate a digest for the PA-Server message and subsequent PCP messages. In addition, the PA-Server message MUST be appended with a SESSION_LIFETIME Option which indicates the new lifetime of the PA session. PA and PCP message sequence numbers must also be reset to zero.

If the EAP authentication fails, the result code of the last PA-Server message is "AUTHENTICATION_FAILED". If the EAP authentication succeeds but authorization fails, the result code of the last PA-Server message is "AUTHORIZATION_FAILED". In the latter two cases, the PA session MUST be terminated immediately after the last PA message exchange. If for some unknown reason re-authentication is

not performed and session lifetime has expired then PA session MUST be terminated immediately.

During re-authentication, the session partners can also exchange common PCP messages in parallel. The common PCP messages MUST be protected with the current SA until the new SA has been generated. The sequence of EAP messages exchanged for re-authentication will not change, regardless of the PCP device triggering re-authentication. If the PCP server receives re-authentication request from the PCP client after it had signaled re-authentication request then it should discard its request and respond to the re-authentication request from the PCP client.

4. PA Security Association

At the beginning of a new PA session, each PCP device must create and initialize state information for a new PA Security Association (PCP SA) to maintain its state information for the duration of the PA session. The parameters of a PCP SA are listed as follows:

- o IP address and UDP port number of the PCP client
- o IP address and UDP port number of the PCP server
- o Session Identifier
- o Sequence number for the next outgoing PA message
- o Sequence number for the next incoming PA message
- o Sequence number for the next outgoing common PCP message
- o Sequence number for the next incoming common PCP message
- o Last outgoing message payload
- o Retransmission interval
- o The master session key (MSK) generated by the EAP method.
- o The MAC algorithm that the transport key should use to generate digests for PCP messages.
- o The pseudo random function negotiated in the initial PA-Server and PA-Client message exchange for the transport key derivation
- o The transport key derived from the MSK to provide integrity protection and data origin authentication for the messages in the

PA session. The lifetime of the transport key SHOULD be identical to the lifetime of the session.

- o The nonce selected by the PCP client at the initiation of the session.

- o The Key ID associated with Transport key.

Particularly, the transport key is computed in the following way: Transport key = prf(MSK, "IETF PCP" || Session ID || Nonce || key ID), where:

- o prf: The pseudo-random function assigned in the Pseudo-random function parameter.

- o MSK: The master session key generated by the EAP method.

- o "IETF PCP": The ASCII code representation of the non-NULL terminated string (excluding the double quotes around it).

- o '||' : is the concatenation operator.

- o Session ID: The ID of the session which the MSK is derived from.

- o Nonce: The nonce selected by the client and transported in the Initial PA-Client message.

- o Key ID: The ID assigned for the transport key.

5. Packet Format

5.1. Packet Format of PCP Auth Messages

The format of the PA-Server message is identical to the response message format specified in Section 7.2 of [RFC6887]. The result code for PA-Sever message carrying EAP request MUST be set to AUTHENTICATION_REQUEST.

As illustrated in Figure 1, this document updates the reserved field in the request header specified in Section 7.1 of [RFC6887] to carry Opcode-specific data.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Version = 2  |R|   Opcode    |   Reserved    |Opcode-specific|
|               | |             |               |   data        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                 Requested Lifetime (32 bits)                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|            PCP Client's IP Address (128 bits)                 |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
:                                                               :
:                   Opcode-specific information                 :
:                                                               :
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
:                                                               :
:                    (optional) PCP Options                     :
:                                                               :
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 1. Request Packet Format

As illustrated in Figure 2, the PA-Client messages use the request header specified in Figure 1. The Opcode-specific data is used to transfer the result codes (e.g., "INITIATION", "AUTHENTICATION_FAILED"). Other fields in Figure 2 are described in Section 7.1 of [RFC6887]. The result code for PA-Client message carrying EAP response MUST be set to AUTHENTICATION_REPLY.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Version = 2  |R|   Opcode    |   Reserved    |  Result Code  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                 Requested Lifetime (32 bits)                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|            PCP Client's IP Address (128 bits)                 |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
:                                                               :
:                   Opcode-specific information                 :
:                                                               :
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
:                                                               :
:                    (optional) PCP Options                     :
:                                                               :
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2. PA-Client message Format

The Requested Lifetime field of PA-Client message and Lifetime field of PA-Server message are both set to 0 on transmission and ignored on reception.

5.2. Opcode-specific information of AUTHENTICATION Opcode

The following diagram shows the format of the Opcode-specific information for the AUTHENTICATION Opcode.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Session ID                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Sequence Number                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Session ID: This field contains a 32-bit PA session identifier.

Sequence Number: This field contains a 32-bit sequence number. A sequence number needs to be incremented on every new (non-retransmission) outgoing PA message in order to provide an ordering guarantee for PA messages.

5.3. NONCE Option

Because the session identifier of a PA session is determined by the PCP server, a PCP client does not know the session identifier which will be used when it sends out a PA-Initiation message. In order to prevent an attacker from interrupting the authentication process by sending off-line generated PA-Server messages, the PCP client needs to generate a random number as a nonce in the PA-Initiation message. The PCP server will append the nonce within the initial PA-Server message. If the PA-Server message does not carry the correct nonce, the message MUST be discarded silently.

```
 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           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Nonce                                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Option Code: TBA-130.

Reserved: 8 bits. MUST be set to 0 on transmission and MUST be ignored on reception.

Option-Length: 4 octets.

Nonce: A random 32 bit number which is transported within a PA-Initiation message and the corresponding reply message from the PCP server.

5.4. AUTHENTICATION_TAG Option

```
 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           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Session ID                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     Sequence Number                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Key ID                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                Authentication Data (Variable)                 |
~                                                               ~
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Because there is no authentication Opcode in common PCP messages, the authentication tag for common PCP messages needs to carry the Session ID and Sequence Number.

Option Code: TBA-131.

Reserved: 8 bits. MUST be set to 0 on transmission and MUST be ignored on reception.

Option-Length: The length of the AUTHENTICATION_TAG Option for Common PCP message (in octets), including the 12 octet fixed header and the variable length of the authentication data.

Session ID: A 32-bit field used to identify the session to which the message belongs and identify the secret key used to create the message digest appended to the PCP message.

Sequence Number: A 32-bit sequence number. In this solution, a sequence number needs to be incremented on every new (non-retransmission) outgoing common PCP message in order to provide ordering guarantee for common PCP messages.

Key ID: The ID associated with the transport key used to generate authentication data. This field is filled with zero if the MSK is directly used to secure the message.

Authentication Data: A variable-length field that carries the Message Authentication Code for the Common PCP message. The generation of the digest varies according to the algorithms

specified in different PCP SAs. This field MUST end on a 32-bit boundary, padded with 0's when necessary.

5.5. PA_AUTHENTICATION_TAG option

This option is used to provide message authentication for PA messages. Compared with the AUTHENTICATION_TAG Option for Common PCP Messages, the Session ID field and the Sequence Number field are removed because such information is provided in the Opcode-specific information of AUTHENTICATION Opcode.

```
 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           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           Key ID                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                 Authentication Data (Variable)                |
~                                                               ~
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Option Code: TBA-132.

Reserved: 8 bits. MUST be set to 0 on transmission and MUST be ignored on reception.

Option-Length: The length of the PA_AUTHENTICATION Option for PCP Auth message (in octet), including the 4 octet fixed header and the variable length of the authentication data.

Key ID: The ID associated with the transport key used to generate authentication data. This field is filled with zero if the MSK is directly used to secure the message.

Authentication Data: A variable-length field that carries the Message Authentication Code for the PCP Auth message. The generation of the digest varies according to the algorithms specified in different PCP SAs. This field MUST end on a 32-bit boundary, padded with null characters when necessary.

5.6. EAP_PAYLOAD Option

```
 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           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                           EAP Message                         |
~                                                               ~
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Option Code: TBA-133.

Reserved: 8 bits. MUST be set to 0 on transmission and MUST be ignored on reception.

Option-Length: Variable

EAP Message: The EAP message transferred. Note this field MUST end on a 32-bit boundary, padded with 0's when necessary.

5.7. PRF Option

```
 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           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             PRF                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Option Code: TBA-134.

Reserved: 8 bits. MUST be set to 0 on transmission and MUST be ignored on reception.

Option-Length: 4 octets.

PRF: The Pseudo-Random Function which the sender supports to generate an MSK. This field contains an IKEv2 Transform ID of Transform Type 2 [RFC7296][RFC4868]. A PCP implementation MUST support PRF_HMAC_SHA2_256 (5).

5.8. MAC_ALGORITHM Option

```
 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           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    MAC Algorithm ID                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Option Code: TBA-135.

Reserved: 8 bits. MUST be set to 0 on transmission and MUST be ignored on reception.

Option-Length: 4 octets.

MAC Algorithm ID: Indicate the MAC algorithm which the sender supports to generate authentication data. The MAC Algorithm ID field contains an IKEv2 Transform ID of Transform Type 3 [RFC7296][RFC4868]. A PCP implementation MUST support AUTH_HMAC_SHA2_256_128 (12).

5.9. SESSION_LIFETIME Option

```
 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           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   Session Lifetime                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Option Code: TBA-136.

Reserved: 8 bits. MUST be set to 0 on transmission and MUST be ignored on reception.

Option-Length: 4 octets.

Session Lifetime: An unsigned 32-bit integer, in seconds, ranging from 0 to 2^32-1 seconds. The lifetime of the PA Session, which is decided by the authorization result.

5.10. RECEIVED_PAK Option

This option is used in a PA-Acknowledgement message to indicate that a PA message with the contained sequence number has been received.

```
 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           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Received Sequence Number                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Option Code: TBA-137.

Reserved: 8 bits. MUST be set to 0 on transmission and MUST be ignored on reception.

Option-Length: 4 octets.

Received Sequence Number: The sequence number of the last received PA message.

5.11. ID_INDICATOR Option

The ID_INDICATOR option is used by the PCP client to determine which credentials to provide to the PCP server.

```
 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           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                           ID Indicator                        |
~                                                               ~
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Option Code: TBA-138.

Reserved: 8 bits. MUST be set to 0 on transmission and MUST be ignored on reception.

Option-Length: Variable.

ID Indicator: The identity of the authority that issued the EAP credentials to be used to authenticate the client. The field MUST

NOT be null terminated and its length is indicated by the Option-Length field. In particular when a client receives a ID_INDICATOR option, it MUST NOT rely on the presence of a NUL character in the wire format data to identify the end of the ID Indicator field.

The field MUST end on a 32-bit boundary, padded with 0's when necessary. The ID indicator field is UTF-8 encoded [RFC3629] Unicode string conforming to the "UsernameCaseMapped" profile of the PRECIS IdentifierClass [I-D.ietf-precis-saslprepbis]. The PCP client validates that the ID indicator field conforms to the "UsernameCaseMapped" profile of the PRECIS IdentifierClass. The PCP client enforces the rules specified in section 3.2.2 of [I-D.ietf-precis-saslprepbis] to map the ID indicator field. The PCP client compares the resulting string with the ID indicators stored locally on the PCP client to pick the credentials for authentication. The two indicator strings are to be considered equivalent by the client if and only if they are an exact octet-for-octet match.

6. Processing Rules

6.1. Authentication Data Generation

After successful EAP authentication process, every subsequent PCP message within the PA session MUST carry an authentication tag which contains the digest of the PCP message for data origin authentication and integrity protection.

- o Before generating a digest for a PA message, a device needs to first locate the PCP SA according to the session identifier and then get the transport key. Then the device appends an PA_AUTHENTICATION_TAG Option for PCP Auth at the end of the PCP Auth message. The length of the Authentication Data field is decided by the MAC algorithm adopted in the session. The device then fills the Key ID field with the key ID of the transport key, and sets the Authentication Data field to 0. After this, the device generates a digest for the entire PCP message (including the PCP header and PA_AUTHENTICATION_TAG Option) using the transport key and the associated MAC algorithm, and inserts the generated digest into the Authentication Data field.

- o Similar to generating a digest for a PA message, before generating a digest for a common PCP message, a device needs to first locate the PCP SA according to the session identifier and then get the transport key. Then the device appends the AUTHENTICATION_TAG Option at the end of common PCP message. The length of the Authentication Data field is decided by the MAC algorithm adopted in the session. The device then uses the corresponding values

derived from the SA to fill the Session ID field, the Sequence Number field and the Key ID field, and sets the Authentication Data field to 0. After this, the device generates a digest for the entire PCP message (including the PCP header and AUTHENTICATION_TAG Option) using the transport key and the associated MAC algorithm, and inputs the generated digest into the Authentication Data field.

6.2. Authentication Data Validation

When a device receives a common PCP message with an AUTHENTICATION_TAG Option for Common PCP Messages, the device needs to use the Session ID transported in the option to locate the proper SA, and then find the associated transport key (using the key ID in the option) and the MAC algorithm. If no proper SA or transport key is found or the sequence number is invalid (see Section 6.5), the PCP device stops processing the PCP message and discards the message silently. After storing the value of the Authentication field of the AUTHENTICATION_TAG Option, the device fills the Authentication field with zeros. Then, the device generates a digest for the message (including the PCP header and Authentication Tag Option) with the transport key and the MAC algorithm. If the value of the newly generated digest is identical to the stored one, the device can ensure that the message has not been tampered with, and the validation succeeds. Otherwise, the PCP device stops processing the PCP message and silently discards the message.

Similarly, when a device receives a PA message with an PA_AUTHENTICATION_TAG Option for PCP Authentication, the device needs to use the Session ID transported in the Opcode to locate the proper SA, and then find the associated transport key (using the key ID in the option) and the MAC algorithm. If no proper SA or transport key is found or the sequence number is invalid (see Section 6.4), the PCP device stops processing the PCP message and discards the message. After storing the value of the Authentication field of the PA_AUTHENTICATION_TAG Option, the device fills the Authentication field with zeros. Then, the device generates a digest for the message (including the PCP header and PA_AUTHENTICATION_TAG Option) with the transport key and the MAC algorithm. If the value of the newly generated digest is identical to the stored one, the device can ensure that the message has not been tampered with, and the validation succeeds. Otherwise, the PCP device stops processing the PCP message and silently discards the message.

6.3. Retransmission Policies for PA Messages

Because EAP relies on the underlying protocols to provide reliable transmission, after sending a PA message, a PCP client/server MUST NOT send out any subsequent messages until receiving a PA message with a proper sequence number from the peer. If no such a message is received the PCP device will re-send the last message according to retransmission policies. This work reuses the retransmission policies specified in the base PCP protocol (Section 8.1.1 of [RFC6887]). In the base PCP protocol, such retransmission policies are only applied by PCP clients. However, in this work, such retransmission policies are also applied by the PCP servers. If Maximum retransmission duration seconds have elapsed and no expected response is received, the device will terminate the session and discard the current SA.

As illustrated in Section 3.1.3, in order to avoid unnecessary re-transmission, the device receiving a PA message MUST send a PA-Acknowledgement message to the sender of the PA message when it cannot send a PA response immediately. The PA-Acknowledgement message is used to indicate the receipt of the PA message. When the sender receives the PA-Acknowledgement message, it will stop the retransmission.

Note that the last PA messages transported within the phases of session initiation, session re-authentication, and session termination do not have to follow the above policies since the devices sending out those messages do not expect any further PA messages.

When a device receives a re-transmitted last incoming PA message from its session partner, it MUST try to answer it by sending the last outgoing PA message again. However, if the duplicate message has the same sequence number but is not bit-wise identical to the original message then the device MUST discard it. In order to achieve this function, the device may need to maintain the last incoming and the associated outgoing messages. In this case, if no outgoing PA message has been generated for the received duplicate PA message yet, the device needs to send a PA-Acknowledgement message. The rate of replying to duplicate PA messages MUST be limited to provide robustness against denial of service (DoS) attacks. The details of rate limiting are outside the scope of this specification.

6.4. Sequence Numbers for PCP Auth Messages

PCP uses UDP to transport signaling messages. As an un-reliable transport protocol, UDP does not guarantee ordered packet delivery and does not provide any protection from packet loss. In order to

ensure the EAP messages are exchanged in a reliable way, every PCP message exchanged during EAP authentication must carry a monotonically increasing sequence number. During a PA session, a PCP device needs to maintain two sequence numbers for PA messages, one for incoming PA messages and one for outgoing PA messages. When generating an outgoing PA message, the device adds the associated outgoing sequence number to the message and increments the sequence number maintained in the SA by 1. When receiving a PA message from its session partner, the device will not accept it if the sequence number carried in the message does not match the incoming sequence number the device maintains. After confirming that the received message is valid, the device increments the incoming sequence number maintained in the SA by 1.

The above rules are not applicable to PA-Acknowledgement messages (i.e., PA messages containing a RECEIVED_PAK Option). A PA-Acknowledgement message does not transport any EAP message and only indicates that a PA message is received. Therefore, reliable transmission of PA-Acknowledgement messages is not required. For instance, after sending out a PA-Acknowledgement message, a device generates an EAP response. In this case, the device need not have to confirm whether the PA-Acknowledgement message has been received by its session partner or not. Therefore, when receiving or sending out a PA-Acknowledgement message, the device MUST NOT increase the corresponding sequence number stored in the SA. Otherwise, loss of a PA-Acknowledgement message will cause a mismatch in sequence numbers.

Another exception is the message retransmission scenario. As discussed in Section 6.3, when a PCP device does not receive any response from its session partner it needs to retransmit the last outgoing PA message following the retransmission procedure specified in section 8.1.1 of [RFC6887]. The original message and duplicate messages MUST be bit-wise identical. When the device receives such a duplicate PA message from its session partner, it MUST send the last outgoing PA message again. In such cases, the maintained incoming and outgoing sequence numbers will not be affected by the message retransmission.

6.5. Sequence Numbers for Common PCP Messages

When transporting common PCP messages within a PA session, a PCP device needs to maintain a sequence number for outgoing common PCP messages and a sequence number for incoming common PCP messages. When generating a new outgoing PCP message, the PCP device updates the Sequence Number field in the AUTHENTICATION_TAG option with the outgoing sequence number maintained in the SA and increments the outgoing sequence number by 1.

When receiving a PCP message from its session partner, the PCP device will not accept it if the sequence number carried in the message is smaller than the incoming sequence number the device maintains. This approach can protect the PCP device from replay attacks. After confirming that the received message is valid, the PCP device will update the incoming sequence number maintained in the PCP SA with the sequence number of the incoming message.

Note that the sequence number in the incoming message may not exactly match the incoming sequence number maintained locally. As discussed in the base PCP specification [RFC6887], if a PCP client is no longer interested in the PCP transaction and has not yet received a PCP response from the server then it will stop retransmitting the PCP request. After that, the PCP client might generate new PCP requests for other purposes using the current SA. In this case, the sequence number in the new request will be larger than the sequence number in the old request and so will be larger than the incoming sequence number maintained in the PCP server.

Note that in the base PCP specification [RFC6887], a PCP client needs to select a nonce in each MAP or PEER request, and the nonce is sent back in the response. However, it is possible for a client to use the same nonce in multiple MAP or PEER requests, and this may cause a potential risk of replay attacks. This attack is addressed by using the sequence number in the PCP response.

6.6. MTU Considerations

EAP methods are responsible for MTU handling, so no special facilities are required in PCP to deal with MTU issues. Particularly, EAP lower layers indicate to EAP methods and AAA servers the MTU of the lower layer. EAP methods such as EAP-TLS [RFC5216], TEAP [RFC7170], and others that are likely to exceed reasonable MTUs provide support for fragmentation and reassembly. Others, such as EAP-GPSK [RFC5433] assume they will never send packets larger than the MTU and use small EAP packets.

If an EAP message is too long to be transported within a single PA message, it will be divided into multiple sections and sent within different PA messages. Note that the receiver may not be able to know what to do in the next step until it has received all the sections and reconstructed the complete EAP message. In this case, in order to guarantee reliable message transmission, after receiving a PA message, the receiver replies with a PA-Acknowledgement message to notify the sender to send the next PA message.

7. IANA Considerations

The following PCP Opcode is to be allocated in the mandatory-to-process range from the standards action range (the registry for PCP Opcodes is maintained in http://www.iana.org/assignments/pcp-parameters):

TBA AUTHENTICATION Opcode.

The following PCP result codes are to be allocated in the mandatory-to-process range from the standards action range (the registry for PCP result codes is maintained in http://www.iana.org/assignments/pcp-parameters):

TBA INITIATION: The client indication to the server for authentication.

TBA AUTHENTICATION_REQUIRED: The error response is signaled to the client that EAP authentication is required.

TBA AUTHENTICATION_FAILED: This error response is signaled to the client if EAP authentication had failed.

TBA AUTHENTICATION_SUCCEEDED:This success response is signaled to the client if EAP authentication had succeeded.

TBA AUTHORIZATION_FAILED: This error response is signaled to the client if the EAP authentication had succeeded but authorization failed.

TBA SESSION_TERMINATED: This PCP result code indicates to the partner that the PA session must be terminated.

TBA UNKNOWN_SESSION_ID: The error response is signaled from the PCP server that there is no known PA session associated with the Session ID signaled in the PA request or common PCP request from the PCP client.

TBA DOWNGRADE_ATTACK_DETECTED: This error response is signaled to the client if the server detects downgrade attack.

TBA AUTHENTICATION_REQUEST: The server indication to the client that EAP request is signaled in the PA message.

TBA AUTHENTICATION_REPLY: The client indication to the server that EAP response is signaled in the PA message.

The following PCP Option Codes are to be allocated in the mandatory-to-process range from the standards action range (the registry for PCP Options is maintained in http://www.iana.org/assignments/pcp-parameters):

7.1. NONCE

Option Name: NONCE

option-code: TBA-130 in the mandatory-to-process range (IANA).

Purpose: See Section 5.3.

Valid for Opcodes: Authentication Opcode.

option-len: Option Length is 4 octets.

May appear in: request and response.

Maximum occurrences: 1.

7.2. AUTHENTICATION_TAG

Option Name: AUTHENTICATION_TAG

option-code: TBA-131 in the mandatory-to-process range (IANA).

Purpose: See Section 5.4.

Valid for Opcodes: MAP, PEER and ANNOUNCE Opcodes.

option-len: Variable length.

May appear in: request and response.

Maximum occurrences: 1.

7.3. PA_AUTHENTICATION_TAG

Option Name: PA_AUTHENTICATION_TAG

option-code: TBA-132 in the mandatory-to-process range (IANA).

Purpose: See Section 5.5.

Valid for Opcodes: Authentication Opcode.

option-len: Variable length.

May appear in: request and response.

Maximum occurrences: 1.

7.4. EAP_PAYLOAD

Option Name: EAP_PAYLOAD.

option-code: TBA-133 in the mandatory-to-process range (IANA).

Purpose: See Section 5.6.

Valid for Opcodes: Authentication Opcode.

option-len: Variable length.

May appear in: request and response.

Maximum occurrences: 1.

7.5. PRF

Option Name: PRF.

option-code: TBA-134 in the mandatory-to-process range (IANA).

Purpose: See Section 5.7.

Valid for Opcodes: Authentication Opcode.

option-len: Option Length is 4 octets.

May appear in: request and response.

Maximum occurrences: as many as fit within maximum PCP message size.

7.6. MAC_ALGORITHM

Option Name: MAC_ALGORITHM.

option-code: TBA-135 in the mandatory-to-process range (IANA).

Purpose: See Section 5.8.

Valid for Opcodes: Authentication Opcode.

option-len: Option Length is 4 octets.

May appear in: request and response.

Maximum occurrences: as many as fit within maximum PCP message size.

7.7. SESSION_LIFETIME

Option Name: SESSION_LIFETIME.

option-code: TBA-136 in the mandatory-to-process range (IANA).

Purpose: See Section 5.9.

Valid for Opcodes: Authentication Opcode.

option-len: Option Length is 4 octets.

May appear in: response.

Maximum occurrences: 1.

7.8. RECEIVED_PAK

Option Name: RECEIVED_PAK.

option-code: TBA-137 in the mandatory-to-process range (IANA).

Purpose: See Section 5.10.

Valid for Opcodes: Authentication Opcode.

option-len: Option Length is 4 octets.

May appear in: request and response.

Maximum occurrences: 1.

7.9. ID_INDICATOR

Option Name: ID_INDICATOR.

option-code: TBA-138 in the mandatory-to-process range (IANA).

Purpose: See Section 5.11.

Valid for Opcodes: Authentication Opcode.

option-len: Variable length.

May appear in: response.

Maximum occurrences: 1.

8. Security Considerations

In this work, after a successful EAP authentication process is performed between two PCP devices, an MSK will be exported. The MSK will be used to derive the transport keys to generate MAC digests for subsequent PCP message exchanges. However, before a transport key has been generated, the PA messages exchanged within a PA session have little cryptographic protection, and if there is no already established security channel between two session partners, these messages are subject to man-in-the-middle attacks and DOS attacks. For instance, the initial PA-Server and PA-Client message exchange is vulnerable to spoofing attacks as these messages are not authenticated and integrity protected. In addition, because the PRF and MAC algorithms are transported at this stage, an attacker may try to remove the PRF and MAC options containing strong algorithms from the initial PA-Server message and force the client choose the weakest algorithms. Therefore, the server needs to guarantee that all the PRF and MAC algorithms it provides support for are strong enough.

In order to prevent very basic DOS attacks, a PCP device SHOULD generate state information as little as possible in the initial PA-Server and PA-Client message exchanges. The choice of EAP method is also very important. The selected EAP method must be resilient to the attacks possible in an insecure network environment, provide user-identity confidentiality, protection against dictionary attacks, and support session-key establishment.

When a PCP proxy [I-D.ietf-pcp-proxy] is located between a PCP server and PCP clients, the proxy may perform authentication with the PCP server before it processes requests from the clients. In addition, re-authentication between the PCP proxy and PCP server will not interrupt the service that the proxy provides to the clients since the proxy is still allowed to send common PCP messages to the PCP server during that period.

9. Acknowledgements

Thanks to Dan Wing, Prashanth Patil, Dave Thaler, Peter Saint-Andre, Carlos Pignataro, Brian Haberman, Paul Kyzivat, Jouni Korhonen, Stephen Farrell and Terry Manderson for the valuable comments.

10. Change Log

[Note: This section should be removed by the RFC Editor upon publication]

10.1. Changes from wasserman-pcp-authentication-02 to ietf-pcp-authentication-00

- o Added discussion of in-band and out-of-band key management options, leaving choice open for later WG decision.
- o Removed support for fragmenting EAP messages, as that is handled by EAP methods.

10.2. Changes from wasserman-pcp-authentication-01 to -02

- o Add a nonce into the first two exchanged PCP-Auth message between the PCP client and PCP server. When a PCP client initiate the session, it can use the nonce to detect offline attacks.
- o Add the key ID field into the authentication tag option so that a MSK can generate multiple transport keys.
- o Specify that when a PCP device receives a PCP-Auth-Server or a PCP-Auth-Client message from its partner the PCP device needs to reply with a PCP-Auth-Acknowledge message to indicate that the message has been received.
- o Add the support of fragmenting EAP messages.

10.3. Changes from ietf-pcp-authentication-00 to -01

- o Editorial changes, added use cases to introduction.

10.4. Changes from ietf-pcp-authentication-01 to -02

- o Add the support of re-authentication initiated by PCP server.
- o Specify that when a PCP device receives a PCP-Auth-Server or a PCP-Auth-Client message from its partner the PCP device MAY reply with a PCP-Auth-Acknowledge message to indicate that the message has been received.
- o Discuss the format of the PCP-Auth-Acknowledge message.
- o Remove the redundant information from the Auth Opcode, and specify new result codes transported in PCP packet headers

- o

10.5. Changes from ietf-pcp-authentication-02 to -03

- o Change the name "PCP-Auth-Request" to "PCP-Auth-Server"
- o Change the name "PCP-Auth-Response" to "PCP-Auth-Client"
- o Specify two new sequence numbers for common PCP messages in the PCP SA, and describe how to use them
- o Specify a Authentication Tag Option for PCP Common Messages
- o Introduce the scenario where a EAP message has to be divided into multiple sections and transported in different PCP-Auth messages (for the reasons of MTU), and introduce how to use PCP-Auth-Acknowledge messages to ensure reliable packet delivery in this case.

10.6. Changes from ietf-pcp-authentication-03 to -04

- o Change the name "PCP-Auth" to "PA".
- o Refine the retransmission policies.
- o Add more discussion about the sequence number management .
- o Provide the discussion about how to instruct a PCP client to choose proper credential during authentication, and an ID Indicator Option is defined for that purpose.

10.7. Changes from ietf-pcp-authentication-04 to -05

- o Add contents in IANA considerations.
- o Add discussions in fragmentation.
- o Refine the PA messages retransmission policies.
- o Add IANA considerations.

10.8. Changes from ietf-pcp-authentication-05 to -06

- o Added mechanism to handle algorithm downgrade attack.
- o Updated Security Considerations section.
- o Updated ID Indicator Option.

11. References

11.1. Normative References

[I-D.ietf-pcp-proxy]
Perreault, S., Boucadair, M., Penno, R., Wing, D., and S. Cheshire, "Port Control Protocol (PCP) Proxy Function", draft-ietf-pcp-proxy-09 (work in progress), July 2015.

[I-D.ietf-precis-saslprepbis]
Saint-Andre, P. and A. Melnikov, "Preparation, Enforcement, and Comparison of Internationalized Strings Representing Usernames and Passwords", draft-ietf-precis-saslprepbis-18 (work in progress), May 2015.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[RFC3629] Yergeau, F., "UTF-8, a transformation format of ISO 10646", STD 63, RFC 3629, DOI 10.17487/RFC3629, November 2003, <http://www.rfc-editor.org/info/rfc3629>.

[RFC3748] Aboba, B., Blunk, L., Vollbrecht, J., Carlson, J., and H. Levkowetz, Ed., "Extensible Authentication Protocol (EAP)", RFC 3748, DOI 10.17487/RFC3748, June 2004, <http://www.rfc-editor.org/info/rfc3748>.

[RFC4868] Kelly, S. and S. Frankel, "Using HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512 with IPsec", RFC 4868, DOI 10.17487/RFC4868, May 2007, <http://www.rfc-editor.org/info/rfc4868>.

[RFC5281] Funk, P. and S. Blake-Wilson, "Extensible Authentication Protocol Tunneled Transport Layer Security Authenticated Protocol Version 0 (EAP-TTLSv0)", RFC 5281, DOI 10.17487/RFC5281, August 2008, <http://www.rfc-editor.org/info/rfc5281>.

[RFC6887] Wing, D., Ed., Cheshire, S., Boucadair, M., Penno, R., and P. Selkirk, "Port Control Protocol (PCP)", RFC 6887, DOI 10.17487/RFC6887, April 2013, <http://www.rfc-editor.org/info/rfc6887>.

[RFC7170] Zhou, H., Cam-Winget, N., Salowey, J., and S. Hanna, "Tunnel Extensible Authentication Protocol (TEAP) Version 1", RFC 7170, DOI 10.17487/RFC7170, May 2014, <http://www.rfc-editor.org/info/rfc7170>.

[RFC7296] Kaufman, C., Hoffman, P., Nir, Y., Eronen, P., and T. Kivinen, "Internet Key Exchange Protocol Version 2 (IKEv2)", STD 79, RFC 7296, DOI 10.17487/RFC7296, October 2014, <http://www.rfc-editor.org/info/rfc7296>.

11.2. Informative References

[RFC5216] Simon, D., Aboba, B., and R. Hurst, "The EAP-TLS Authentication Protocol", RFC 5216, DOI 10.17487/RFC5216, March 2008, <http://www.rfc-editor.org/info/rfc5216>.

[RFC5433] Clancy, T. and H. Tschofenig, "Extensible Authentication Protocol - Generalized Pre-Shared Key (EAP-GPSK) Method", RFC 5433, DOI 10.17487/RFC5433, February 2009, <http://www.rfc-editor.org/info/rfc5433>.

[RFC5448] Arkko, J., Lehtovirta, V., and P. Eronen, "Improved Extensible Authentication Protocol Method for 3rd Generation Authentication and Key Agreement (EAP-AKA')", RFC 5448, DOI 10.17487/RFC5448, May 2009, <http://www.rfc-editor.org/info/rfc5448>.

Authors' Addresses

Margaret Wasserman
Painless Security
356 Abbott Street
North Andover, MA 01845
USA

Phone: +1 781 405 7464
Email: mrw@painless-security.com
URI: http://www.painless-security.com

Sam Hartman
Painless Security
356 Abbott Street
North Andover, MA 01845
USA

Email: hartmans@painless-security.com
URI: http://www.painless-security.com

Dacheng Zhang
Huawei
Beijing
China

Email: zhang_dacheng@hotmail.com

Tirumaleswar Reddy
Cisco Systems, Inc.
Cessna Business Park, Varthur Hobli
Sarjapur Marathalli Outer Ring Road
Bangalore, Karnataka 560103
India

Email: tireddy@cisco.com

PCP working group
Internet-Draft
Intended status: Standards Track
Expires: May 11, 2013

D. Wing, Ed.
Cisco
S. Cheshire
Apple
M. Boucadair
France Telecom
R. Penno
Cisco
P. Selkirk
ISC

November 7, 2012

Port Control Protocol (PCP)
draft-ietf-pcp-base-29

Abstract

The Port Control Protocol allows an IPv6 or IPv4 host to control how incoming IPv6 or IPv4 packets are translated and forwarded by a network address translator (NAT) or simple firewall, and also allows a host to optimize its outgoing NAT keepalive messages.

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 May 11, 2013.

Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of

publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction 5
2. Scope 6
2.1. Deployment Scenarios 6
2.2. Supported Protocols 6
2.3. Single-homed Customer Premises Network 6
3. Terminology 7
4. Relationship between PCP Server and its NAT/firewall 11
5. Note on Fixed-Size Addresses 11
6. Protocol Design Note 12
7. Common Request and Response Header Format 14
7.1. Request Header 15
7.2. Response Header 16
7.3. Options 17
7.4. Result Codes 20
8. General PCP Operation 21
8.1. General PCP Client: Generating a Request 22
8.1.1. PCP Client Retransmission 23
8.2. General PCP Server: Processing a Request 25
8.3. General PCP Client: Processing a Response 27
8.4. Multi-Interface Issues 28
8.5. Epoch 28
9. Version Negotiation 30
10. Introduction to MAP and PEER Opcodes 31
10.1. For Operating a Server 33
10.2. For Operating a Symmetric Client/Server 36
10.3. For Reducing NAT or Firewall Keepalive Messages 38
10.4. For Restoring Lost Implicit TCP Dynamic Mapping State . . 39
11. MAP Opcode 40
11.1. MAP Operation Packet Formats 41
11.2. Generating a MAP Request 44
11.2.1. Renewing a Mapping 45
11.3. Processing a MAP Request 45
11.4. Processing a MAP Response 48
11.5. Address Change Events 49
11.6. Learning the External IP Address Alone 50
12. PEER Opcode 51
12.1. PEER Operation Packet Formats 51
12.2. Generating a PEER Request 55

12.3. Processing a PEER Request 56
12.4. Processing a PEER Response 57
13. Options for MAP and PEER Opcodes 58
13.1. THIRD_PARTY Option for MAP and PEER Opcodes 58
13.2. PREFER_FAILURE Option for MAP Opcode 60
13.3. FILTER Option for MAP Opcode 62
14. Rapid Recovery 64
14.1. ANNOUNCE Opcode 65
14.1.1. ANNOUNCE Operation 65
14.1.2. Generating and Processing a Solicited ANNOUNCE Message 66
14.1.3. Generating and Processing an Unsolicited ANNOUNCE Message 66
14.2. PCP Mapping Update 68
15. Mapping Lifetime and Deletion 69
15.1. Lifetime Processing for the MAP Opcode 71
16. Implementation Considerations 72
16.1. Implementing MAP with EDM port-mapping NAT 72
16.2. Lifetime of Explicit and Implicit Dynamic Mappings . . . 73
16.3. PCP Failure Recovery 73
16.3.1. Recreating Mappings 73
16.3.2. Maintaining Mappings 74
16.3.3. SCTP 74
16.4. Source Address Replicated in PCP Header 75
16.5. State Diagram 76
17. Deployment Considerations 77
17.1. Ingress Filtering 77
17.2. Mapping Quota 78
18. Security Considerations 78
18.1. Simple Threat Model 78
18.1.1. Attacks Considered 79
18.1.2. Deployment Examples Supporting the Simple Threat Model 80
18.2. Advanced Threat Model 80
18.3. Residual Threats 81
18.3.1. Denial of Service 81
18.3.2. Ingress Filtering 81
18.3.3. Mapping Theft 81
18.3.4. Attacks Against Server Discovery 82
19. IANA Considerations 82
19.1. Port Number 82
19.2. Opcodes 82
19.3. Result Codes 82
19.4. Options 83
20. Acknowledgments 83
21. References 84
21.1. Normative References 84
21.2. Informative References 84

Appendix A. NAT-PMP Transition 87
Appendix B. Change History 87
B.1. Changes from draft-ietf-pcp-base-28 to -29 88
B.2. Changes from draft-ietf-pcp-base-27 to -28 88
B.3. Changes from draft-ietf-pcp-base-26 to -27 88
B.4. Changes from draft-ietf-pcp-base-25 to -26 90
B.5. Changes from draft-ietf-pcp-base-24 to -25 90
B.6. Changes from draft-ietf-pcp-base-23 to -24 91
B.7. Changes from draft-ietf-pcp-base-22 to -23 93
B.8. Changes from draft-ietf-pcp-base-21 to -22 95
B.9. Changes from draft-ietf-pcp-base-20 to -21 95
B.10. Changes from draft-ietf-pcp-base-19 to -20 95
B.11. Changes from draft-ietf-pcp-base-18 to -19 95
B.12. Changes from draft-ietf-pcp-base-17 to -18 96
B.13. Changes from draft-ietf-pcp-base-16 to -17 96
B.14. Changes from draft-ietf-pcp-base-15 to -16 96
B.15. Changes from draft-ietf-pcp-base-14 to -15 97
B.16. Changes from draft-ietf-pcp-base-13 to -14 97
B.17. Changes from draft-ietf-pcp-base-12 to -13 98
B.18. Changes from draft-ietf-pcp-base-11 to -12 99
B.19. Changes from draft-ietf-pcp-base-10 to -11 99
B.20. Changes from draft-ietf-pcp-base-09 to -10 99
B.21. Changes from draft-ietf-pcp-base-08 to -09 99
B.22. Changes from draft-ietf-pcp-base-07 to -08 100
B.23. Changes from draft-ietf-pcp-base-06 to -07 101
B.24. Changes from draft-ietf-pcp-base-05 to -06 102
B.25. Changes from draft-ietf-pcp-base-04 to -05 104
B.26. Changes from draft-ietf-pcp-base-03 to -04 104
B.27. Changes from draft-ietf-pcp-base-02 to -03 105
B.28. Changes from draft-ietf-pcp-base-01 to -02 105
B.29. Changes from draft-ietf-pcp-base-00 to -01 106
Authors' Addresses 106

1. Introduction

The Port Control Protocol (PCP) provides a mechanism to control how incoming packets are forwarded by upstream devices such as Network Address Translator IPv6/IPv4 (NAT64), Network Address Translator IPv4/IPv4 (NAT44), IPv6 and IPv4 firewall devices, and a mechanism to reduce application keepalive traffic. PCP is designed to be implemented in the context of Carrier-Grade NATs (CGNs), small NATs (e.g., residential NATs), as well as with dual-stack and IPv6-only Customer Premises Equipment (CPE) routers, and all of the currently-known transition scenarios towards IPv6-only CPE routers. PCP allows hosts to operate servers for a long time (e.g., a network-attached home security camera) or a short time (e.g., while playing a game or on a phone call) when behind a NAT device, including when behind a CGN operated by their Internet service provider or an IPv6 firewall integrated in their CPE router.

PCP allows applications to create mappings from an external IP address, protocol, and port to an internal IP address, protocol, and port. These mappings are required for successful inbound communications destined to machines located behind a NAT or a firewall.

After creating a mapping for incoming connections, it is necessary to inform remote computers about the IP address, protocol, and port for the incoming connection. This is usually done in an application-specific manner. For example, a computer game might use a rendezvous server specific to that game (or specific to that game developer), a SIP phone would use a SIP proxy, and a client using DNS-Based Service Discovery [I-D.cheshire-dnsext-dns-sd] would use DNS Update [RFC2136] [RFC3007]. PCP does not provide this rendezvous function. The rendezvous function may support IPv4, IPv6, or both. Depending on that support and the application’s support of IPv4 or IPv6, the PCP client may need an IPv4 mapping, an IPv6 mapping, or both.

Many NAT-friendly applications send frequent application-level messages to ensure their session will not be timed out by a NAT. These are commonly called "NAT keepalive" messages, even though they are not sent to the NAT itself (rather, they are sent ’through’ the NAT). These applications can reduce the frequency of such NAT keepalive messages by using PCP to learn (and influence) the NAT mapping lifetime. This helps reduce bandwidth on the subscriber’s access network, traffic to the server, and battery consumption on mobile devices.

Many NATs and firewalls include Application Layer Gateways (ALGs) to create mappings for applications that establish additional streams or accept incoming connections. ALGs incorporated into NATs may also

modify the application payload. Industry experience has shown that these ALGs are detrimental to protocol evolution. PCP allows an application to create its own mappings in NATs and firewalls, reducing the incentive to deploy ALGs in NATs and firewalls.

2. Scope

2.1. Deployment Scenarios

PCP can be used in various deployment scenarios, including:

- o Basic NAT [RFC3022]
- o Network Address and Port Translation [RFC3022], such as commonly deployed in residential NAT devices
- o Carrier-Grade NAT [I-D.ietf-behave-lsn-requirements]
- o Dual-Stack Lite (DS-Lite) [RFC6333]
- o Layer-2 Aware NAT [I-D.miles-behave-l2nat]
- o Dual-Stack Extra Lite [RFC6619]
- o NAT64, both Stateless [RFC6145] and Stateful [RFC6146]
- o IPv4 and IPv6 simple firewall control [RFC6092]
- o IPv6-to-IPv6 Network Prefix Translation (NPTv6) [RFC6296]

2.2. Supported Protocols

The PCP Opcodes defined in this document are designed to support transport-layer protocols that use a 16-bit port number (e.g., TCP, UDP, SCTP [RFC4960], DCCP [RFC4340]). Protocols that do not use a port number (e.g., RSVP, IPsec ESP [RFC4303], ICMP, ICMPv6) are supported for IPv4 firewall, IPv6 firewall, and NPTv6 functions, but are out of scope for any NAT functions.

2.3. Single-homed Customer Premises Network

PCP assumes a single-homed IP address model. That is, for a given IP address of a host, only one default route exists to reach other hosts on the Internet from that source IP address. This is important because after a PCP mapping is created and an inbound packet (e.g., TCP SYN) is rewritten and delivered to a host, the outbound response (e.g., TCP SYNACK) has to go through the same (reverse) path so it

passes through the same NAT to have the necessary inverse rewrite performed. This restriction exists because otherwise there would need to be a PCP-enabled NAT for every egress (because the host could not reliably determine which egress path packets would take) and the client would need to be able to reliably make the same internal/ external mapping in every NAT gateway, which in general is not possible (because the other NATs might already have the necessary External Port mapped to another host).

3. Terminology

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 "Key words for use in RFCs to Indicate Requirement Levels" [RFC2119].

Internal Host:
A host served by a NAT gateway, or protected by a firewall. This is the host that will receive incoming traffic resulting from a PCP mapping request, or the host that initiated an implicit dynamic outbound mapping (e.g., by sending a TCP SYN) across a firewall or a NAT.

Remote Peer Host:
A host with which an Internal Host is communicating. This can include another Internal Host (or even the same Internal Host); if a NAT is involved, the NAT would need to hairpin the traffic [RFC4787].

Internal Address:
The address of an Internal Host served by a NAT gateway or protected by a firewall.

External Address:
The address of an Internal Host as seen by other Remote Peers on the Internet with which the Internal Host is communicating, after translation by any NAT gateways on the path. An External Address is generally a public routable (i.e., non-private) address. In the case of an Internal Host protected by a pure firewall, with no address translation on the path, its External Address is the same as its Internal Address.

Endpoint-Dependent Mapping (EDM): A term applied to NAT operation where an implicit mapping created by outgoing traffic (e.g., TCP SYN) from a single Internal Address, Protocol, and Port to different Remote Peers and Ports may be assigned different External Ports, and a subsequent PCP mapping request for that

Internal Address, Protocol, and Port may be assigned yet another different External Port. This term encompasses both Address-Dependent Mapping and Address and Port-Dependent Mapping [RFC4787].

Endpoint-Independent Mapping (EIM): A term applied to NAT operation where all mappings from a single Internal Address, Protocol, and Port to different Remote Peers and Ports are all assigned the same External Address and Port.

Remote Peer Address:
The address of a Remote Peer, as seen by the Internal Host. A Remote Address is generally a publicly routable address. In the case of a Remote Peer that is itself served by a NAT gateway, the Remote Address may in fact be the Remote Peer’s External Address, but since this remote translation is generally invisible to software running on the Internal Host, the distinction can safely be ignored for the purposes of this document.

Third Party:
In the common case, an Internal Host manages its own Mappings using PCP requests, and the Internal Address of those Mappings is the same as the source IP address of the PCP request packet.

In the case where one device is managing Mappings on behalf of some other device that does not implement PCP, the presence of the THIRD_PARTY Option in the MAP request signifies that the specified address, rather than the source IP address of the PCP request packet, should be used as the Internal Address for the Mapping.

Mapping, Port Mapping, Port Forwarding:
A NAT mapping creates a relationship between an internal IP address, protocol, and port, and an external IP address, protocol, and port. More specifically, it creates a translation rule where packets destined to the external IP and port are translated to the internal IP address, protocol, and port, and vice versa. In the case of a pure firewall, the "Mapping" is the identity function, translating an internal IP address, protocol, and port number to the same external IP address, protocol, and port number. Firewall filtering, applied in addition to that identity mapping function, is separate from the mapping itself.

Mapping Types:
There are three dimensions to classifying mapping types: how they are created (implicitly/explicitly), their primary purpose (outbound/inbound), and how they are deleted (dynamic/static). Implicit mappings are created as a side-effect of some other operation; explicit mappings are created by a mechanism explicitly

dealing with mappings. Outbound mappings exist primarily to facilitate outbound communication; inbound mappings exist primarily to facilitate inbound communication. Dynamic mappings are deleted when their lifetime expires, or though other protocol action; static mappings are permanent until the user chooses to delete them.

* Implicit dynamic mappings are created implicitly as a side-effect of traffic such as an outgoing TCP SYN or outgoing UDP packet. Such packets were not originally designed explicitly for creating NAT (or firewall) state, but they can have that effect when they pass through a NAT (or firewall) device. Implicit dynamic mappings usually have a finite lifetime, though this lifetime is generally not known to the client using them.

* Explicit dynamic mappings are created as a result of explicit PCP MAP and PEER requests. Like a DHCP address lease, explicit dynamic mappings have finite lifetime, and this lifetime is communicated to the client. As with a DHCP address lease, if the client wants a mapping to persist the client must prove that it is still present by periodically renewing the mapping to prevent it from expiring. If a PCP client goes away, then any mappings it created will be automatically cleaned up when they expire.

* Explicit static mappings are created by manual configuration (e.g., via command-line interface or other user interface) and persist until the user changes that manual configuration.

Both implicit and explicit dynamic mappings are dynamic in the sense that they are created on demand, as requested (implicitly or explicitly) by the Internal Host, and have a lifetime. After the lifetime, the mapping is deleted unless the lifetime is extended by action by the Internal Host (e.g., sending more traffic or sending another PCP request).

Static mappings are by their nature always explicit. Static mappings differ from explicit dynamic mappings in that their lifetime is effectively infinite (they exist until manually removed) but otherwise they behave exactly the same as explicit MAP mappings.

While all mappings are by necessity bidirectional (most Internet communication requires information to flow in both directions for successful operation) when talking about mappings it can be helpful to identify them loosely according to their ’primary’ purpose.

* Outbound mappings exist primarily to enable outbound communication. For example, when a host calls connect() to make an outbound connection, a NAT gateway will create an implicit dynamic outbound mapping to facilitate that outbound communication.

* Inbound mappings exist primarily to enable listening servers to receive inbound connections. Generally, when a client calls listen() to listen for inbound connections, a NAT gateway will not implicitly create any mapping to facilitate that inbound communication. A PCP MAP request can be used explicitly to create a dynamic inbound mapping to enable the desired inbound communication.

Explicit static (manual) mappings and explicit dynamic (MAP) mappings both allow Internal Hosts to receive inbound traffic that is not in direct response to any immediately preceding outbound communication (i.e., to allow Internal Hosts to operate a "server" that is accessible to other hosts on the Internet).

PCP Client:
A PCP software instance responsible for issuing PCP requests to a PCP server. Several independent PCP Clients can exist on the same host. Several PCP Clients can be located in the same local network. A PCP Client can issue PCP requests on behalf of a third party device for which it is authorized to do so. An interworking function from Universal Plug and Play Internet Gateway Device (UPnP IGDv1 [IGDv1]) to PCP is another example of a PCP Client. A PCP server in a NAT gateway that is itself a client of another NAT gateway (nested NAT) may itself act as a PCP client to the upstream NAT.

PCP-Controlled Device:
A NAT or firewall that controls or rewrites packet flows between internal hosts and remote peer hosts. PCP manages the Mappings on this device.

PCP Server:
A PCP software instance that resides on the NAT or firewall that receives PCP requests from the PCP client and creates appropriate state in response to that request.

Subscriber:
The unit of billing for a commercial ISP. A subscriber may have a single IP address from the commercial ISP (which can be shared among multiple hosts using a NAT gateway, thereby making them appear to be a single host to the ISP) or may have multiple IP addresses provided by the commercial ISP. In either case, the IP

address or addresses provided by the ISP may themselves be further translated by a Carrier-Grade NAT (CGN) operated by the ISP.

4. Relationship between PCP Server and its NAT/firewall

The PCP server receives and responds to PCP requests. The PCP server functionality is typically a capability of a NAT or firewall device, as shown in Figure 1. It is also possible for the PCP functionality to be provided by some other device, which communicates with the actual NAT(s) or firewall(s) via some other proprietary mechanism, as long as from the PCP client's perspective such split operation is indistinguishable from the integrated case.

```
                                +-----------------+
      +------------+            | NAT or firewall |
      | PCP client |-<network>-+      with       +---<Internet>
      +------------+            |    PCP server   |
                                +-----------------+
```

Figure 1: PCP-Enabled NAT or Firewall

A NAT or firewall device, between the PCP client and the Internet, might implement simple or advanced firewall functionality. This may be a side-effect of the technology implemented by the device (e.g., a network address and port translator, by virtue of its port rewriting, normally requires connections to be initiated from an inside host towards the Internet), or this might be an explicit firewall policy to deny unsolicited traffic from the Internet. Some firewall devices deny certain unsolicited traffic from the Internet (e.g., TCP, UDP to most ports) but allow certain other unsolicited traffic from the Internet (e.g., UDP port 500 and IPsec ESP) [RFC6092]. Such default filtering (or lack thereof) is out of scope of PCP itself. If a client device wants to receive traffic and supports PCP, and does not possess prior knowledge of such default filtering policy, it SHOULD use PCP to request the necessary mappings to receive the desired traffic.

5. Note on Fixed-Size Addresses

For simplicity in building and parsing request and response packets, PCP always uses fixed-size 128-bit IP address fields for both IPv6 addresses and IPv4 addresses.

When the address field holds an IPv6 address, the fixed-size 128-bit IP address field holds the IPv6 address stored as-is.

When the address field holds an IPv4 address, IPv4-mapped IPv6 addresses [RFC4291] are used (::ffff:0:0/96). This has the first 80 bits set to zero and the next 16 set to one, while its last 32 bits are filled with the IPv4 address. This is unambiguously distinguishable from a native IPv6 address, because an IPv4-mapped IPv6 address [RFC4291] would not be valid for a mapping.

When checking for an IPv4-mapped IPv6 address, all of the first 96 bits MUST be checked for the pattern -- it is not sufficient to check for ones in bits 81-96.

The all-zeroes IPv6 address MUST be expressed by filling the fixed-size 128-bit IP address field with all zeroes (::).

The all-zeroes IPv4 address MUST be expressed by 80 bits of zeros, 16 bits of ones, and 32 bits of zeros (::ffff:0:0).

6. Protocol Design Note

PCP can be viewed as a request/response protocol, much like many other UDP-based request/response protocols, and can be implemented perfectly well as such. It can also be viewed as what might be called a hint/notification protocol, and this observation can help simplify implementations.

Rather than viewing the message streams between PCP client and PCP server as following a strict request/response pattern, where every response is associated with exactly one request, the message flows can be viewed as two somewhat independent streams carrying information in opposite directions:

- o A stream of hints flowing from PCP client to PCP server, where the client indicates to the server what it would like the state of its mappings to be, and

- o A stream of notifications flowing from PCP server to PCP client, where the server informs the clients what the state of its mappings actually is.

To an extent, some of this approach is required anyway in a UDP-based request/response protocol, since UDP packets can be lost, duplicated, or reordered.

In this view of the protocol, the client transmits hints to the server at various intervals signaling its desires, and the server transmits notifications to the client signaling the actual state of its mappings. These two message flows are loosely correlated in that

a client request (hint) usually elicits a server response (notification), but only loosely, in that a client request may result in no server response (in the case of packet loss) and a server response may be generated gratuitously without an immediately preceding client request (in the case where server configuration change, e.g. change of external IP address on a NAT gateway, results in a change of mapping state).

The exact times that client requests are sent are influenced by a client timing state machine taking into account whether (i) the client has not yet received a response from the server for a prior request (retransmission), or (ii) the client has previously received a response from the server saying how long the indicated mapping would remain active (renewal). This design philosophy is the reason why PCP's retransmissions and renewals are exactly the same packet on the wire. Typically, retransmissions are sent with exponentially increasing intervals as the client waits for the server to respond, whereas renewals are sent with exponentially decreasing intervals as the expiry time approaches, but from the server's point of view both packets are identical, and both signal the client's desire that the stated mapping exist or continue to exist.

A PCP server usually sends responses as a direct result of client requests, but not always. For example, if a server is too overloaded to respond, it is allowed to silently ignore a request message and let the client retransmit. Also, if external factors cause a NAT gateway or firewall's configuration to change, then the PCP server can send unsolicited responses to clients informing them of the new state of their mappings. Such reconfigurations are expected to be rare, because of the disruption they can cause to clients, but should they happen, PCP provides a way for servers to communicate the new state to clients promptly, without having to wait for the next periodic renewal request.

This design goal helps explain why PCP request and response messages have no transaction ID, because such a transaction ID is unnecessary, and would unnecessarily limit the protocol and unnecessarily complicate implementations. A PCP server response (i.e. notification) is self-describing and complete. It communicates the internal and external addresses, protocol, and ports for a mapping, and its remaining lifetime. If the client does in fact currently want such a mapping to exist then it can identify the mapping in question from the internal address, protocol, and port, and update its state to reflect the current external address and port, and remaining lifetime. If a client does not currently want such a mapping to exist then it can safely ignore the message. No client action is required for unexpected mapping notifications. In today's world a NAT gateway can have a static mapping, and the client device

has no explicit knowledge of this, and no way to change the fact. Also, in today's world a client device can be connected directly to the public Internet, with a globally-routable IP address, and in this case it effectively has "mappings" for all of its listening ports. Such a device has to be responsible for its own security, and cannot rely on assuming that some other network device will be blocking all incoming packets.

7. Common Request and Response Header Format

All PCP messages are sent over UDP, with a maximum UDP payload length of 1100 octets. The PCP messages contain a request or response header containing an Opcode, any relevant Opcode-specific information, and zero or more Options. All numeric quantities larger than a single octet (e.g. Result codes, Lifetimes, Epoch times, etc.) are represented in conventional IETF network order, i.e. most significant octet first. Non-numeric quantities are represented as-is on all platforms, with no byte swapping (e.g. IP addresses and ports are placed in PCP messages using the same representation as when placed in IP or TCP headers).

The packet layout for the common header, and operation of the PCP client and PCP server, are described in the following sections. The information in this section applies to all Opcodes. Behavior of the Opcodes defined in this document is described in Section 11 and Section 12.

7.1. Request Header

All requests have the following format:

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Version = 2  |R|   Opcode    |         Reserved              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                 Requested Lifetime (32 bits)                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|            PCP Client's IP Address (128 bits)                 |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
:                                                               :
:             (optional) Opcode-specific information            :
:                                                               :
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
:                                                               :
:             (optional) PCP Options                            :
:                                                               :
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2: Common Request Packet Format

These fields are described below:

Version: This document specifies protocol version 2. PCP clients and servers compliant with this document use the value 2. This field is used for version negotiation as described in Section 9.

R: Indicates Request (0) or Response (1).

Opcode: A seven-bit value specifying the operation to be performed. Opcodes are defined in Section 11 and Section 12.

Reserved: 16 reserved bits. MUST be zero on transmission and MUST be ignored on reception.

Requested Lifetime: An unsigned 32-bit integer, in seconds, ranging from 0 to 2^32-1 seconds. This is used by the MAP and PEER Opcodes defined in this document for their requested lifetime.

PCP Client's IP Address: The source IPv4 or IPv6 address in the IP header used by the PCP client when sending this PCP request. IPv4 is represented using an IPv4-mapped IPv6 address. This is used to detect an unexpected NAT on the path between the PCP client and the PCP-controlled NAT or firewall device. See Section 8.1

Opcode-specific information: Payload data for this Opcode. The length of this data is determined by the Opcode definition.

PCP Options: Zero, one, or more Options that are legal for both a PCP request and for this Opcode. See Section 7.3.

7.2. Response Header

All responses have the following format:

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Version = 2  |R|   Opcode    |   Reserved    |  Result Code  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Lifetime (32 bits)                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     Epoch Time (32 bits)                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                      Reserved (96 bits)                       |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
:                                                               :
:             (optional) Opcode-specific response data          :
:                                                               :
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
:             (optional) Options                                :
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 3: Common Response Packet Format

These fields are described below:

Version: Responses from servers compliant with this specification MUST use version 2. This is set by the server.

R: Indicates Request (0) or Response (1). All Responses MUST use 1. This is set by the server.

Opcode: The 7-bit Opcode value. The server copies this value from the request.

Reserved: 8 reserved bits, MUST be sent as 0, MUST be ignored when received. This is set by the server.

Result Code: The result code for this response. See Section 7.4 for values. This is set by the server.

Lifetime: An unsigned 32-bit integer, in seconds, ranging from 0 to 2^32-1 seconds. On an error response, this indicates how long clients should assume they'll get the same error response from that PCP server if they repeat the same request. On a success response for the PCP Opcodes that create a mapping (MAP and PEER), the Lifetime field indicates the lifetime for this mapping. This is set by the server.

Epoch Time: The server's Epoch time value. See Section 8.5 for discussion. This value is set by the server, in both success and error responses.

Reserved: 96 reserved bits. For requests that were successfully parsed, this MUST be sent as 0, MUST be ignored when received. This is set by the server. For requests that were not successfully parsed, the server copies the last 96 bits of the PCP Client's IP Address field from the request message into this corresponding 96 bit field of the response.

Opcode-specific information: Payload data for this Opcode. The length of this data is determined by the Opcode definition.

PCP Options: Zero, one, or more Options that are legal for both a PCP response and for this Opcode. See Section 7.3.

7.3. Options

A PCP Opcode can be extended with one or more Options. Options can be used in requests and responses. The design decisions in this specification about whether to include a given piece of information in the base Opcode format or in an Option were an engineering trade-off between packet size and code complexity. For information that is usually (or always) required, placing it in the fixed Opcode data results in simpler code to generate and parse the packet, because the information is a fixed location in the Opcode data, but wastes space in the packet in the event that field is all-zeroes because the information is not needed or not relevant. For information that is required less often, placing it in an Option results in slightly more complicated code to generate and parse packets containing that

Option, but saves space in the packet when that information is not needed. Placing information in an Option also means that an implementation that never uses that information doesn't even need to implement code to generate and parse it. For example, a client that never requests mappings on behalf of some other device doesn't need to implement code to generate the THIRD_PARTY Option, and a PCP server that doesn't implement the necessary security measures to create third-party mappings safely doesn't need to implement code to parse the THIRD_PARTY Option.

Options use the following Type-Length-Value format:

```
 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           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
:                       (optional) data                         :
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 4: Options Header

The description of the fields is as follows:

Option Code: 8 bits. Its most significant bit indicates if this Option is mandatory (0) or optional (1) to process.

Reserved: 8 bits. MUST be set to 0 on transmission and MUST be ignored on reception.

Option Length: 16 bits. Indicates the length of the enclosed data, in octets. Options with length of 0 are allowed. Options that are not a multiple of four octets long are followed by one, two, or three zero octets to pad their effective length in the packet to be a multiple of four octets. The Option Length reflects the semantic length of the option, not including any padding octets.

data: Option data.

If several Options are included in a PCP request, they MAY be encoded in any order by the PCP client, but MUST be processed by the PCP server in the order in which they appear. It is the responsibility of the PCP client to ensure the server has sufficient room to reply without exceeding the 1100 octet size limit; if its reply would exceed that size, the server generates an error.

If, while processing a PCP request, including its options, an error is encountered that causes a PCP error response to be generated, the

PCP request MUST cause no state change in the PCP server or the PCP-controlled device (i.e., it rolls back any changes it might have made while processing the request). Such an error response MUST consist of a complete copy of the request packet with the error code and other appropriate fields set in the header.

An Option MAY appear more than once in a request or in a response, if permitted by the definition of the Option. If the Option's definition allows the Option to appear only once but it appears more than once in a request, and the Option is understood by the PCP server, the PCP server MUST respond with the MALFORMED_OPTION result code. If the PCP server encounters an invalid option (e.g., PCP option length is longer than the UDP packet length) the error MALFORMED_OPTION SHOULD be returned (rather than MALFORMED_REQUEST), as that helps the client better understand how the packet was malformed. If a PCP response would have exceeded the maximum PCP message size, the PCP server SHOULD respond with MALFORMED_REQUEST.

If the overall Option structure of a request cannot successfully be parsed (e.g. a nonsensical option length) the PCP server MUST generate an error response with code MALFORMED_OPTION.

If the overall Option structure of a request is valid then how each individual Option is handled is determined by the most significant bit in the Option Code. If the most significant bit is set, handling this Option is optional, and a PCP server MAY process or ignore this Option, entirely at its discretion. If the most significant bit is clear, handling this Option is mandatory, and a PCP server MUST return the error MALFORMED_OPTION if the option contents are malformed, or UNSUPP_OPTION if the Option is unrecognized, unimplemented, or disabled, or if the client is not authorized to use the Option. In error responses all options are returned. In success responses all processed options are included and unprocessed options are not included.

PCP clients are free to ignore any or all Options included in responses, although naturally if a client explicitly requests an Option where correct execution of that Option requires processing the Option data in the response, that client is expected to implement code to do that.

Different options are valid for different Opcodes. For example:

- o The THIRD_PARTY Option is valid for both MAP and PEER Opcodes.

- o The FILTER Option is valid only for the MAP Opcode (for the PEER Opcode it would have no meaning).

- o The PREFER_FAILURE Option is valid only for the MAP Opcode (for the PEER Opcode, similar semantics are automatically implied).

7.4. Result Codes

The following result codes may be returned as a result of any Opcode received by the PCP server. The only success result code is 0; other values indicate an error. If a PCP server encounters multiple errors during processing of a request, it SHOULD use the most specific error message. Each error code below is classified as either a 'long lifetime' error or a 'short lifetime' error, which provides guidance to PCP server developers for the value of the Lifetime field for these errors. It is RECOMMENDED that short lifetime errors use a 30 second lifetime and long lifetime errors use a 30 minute lifetime.

0 SUCCESS: Success.

1 UNSUPP_VERSION: The version number at the start of the PCP Request header is not recognized by this PCP server. This is a long lifetime error. This document describes PCP version 2.

2 NOT_AUTHORIZED: The requested operation is disabled for this PCP client, or the PCP client requested an operation that cannot be fulfilled by the PCP server's security policy. This is a long lifetime error.

3 MALFORMED_REQUEST: The request could not be successfully parsed. This is a long lifetime error.

4 UNSUPP_OPCODE: Unsupported Opcode. This is a long lifetime error.

5 UNSUPP_OPTION: Unsupported Option. This error only occurs if the Option is in the mandatory-to-process range. This is a long lifetime error.

6 MALFORMED_OPTION: Malformed Option (e.g., appears too many times, invalid length). This is a long lifetime error.

7 NETWORK_FAILURE: The PCP server or the device it controls are experiencing a network failure of some sort (e.g., has not obtained an External IP address). This is a short lifetime error.

8 NO_RESOURCES: Request is well-formed and valid, but the server has insufficient resources to complete the requested operation at this time. For example, the NAT device cannot create more mappings at this time, is short of CPU cycles or memory, or is unable to handle the request due to some other temporary condition. The same request may succeed in the future. This is a system-wide error, different from USER_EX_QUOTA. This can be used as a catch-all error, should no other error message be suitable. This is a short lifetime error.

9 UNSUPP_PROTOCOL: Unsupported transport protocol, e.g. SCTP in a NAT that handles only UDP and TCP. This is a long lifetime error.

10 USER_EX_QUOTA: This attempt to create a new mapping would exceed this subscriber's port quota. This is a short lifetime error.

11 CANNOT_PROVIDE_EXTERNAL: The suggested external port and/or external address cannot be provided. This error MUST only be returned for:
* MAP requests that included the PREFER_FAILURE Option (normal MAP requests will return an available external port)
* MAP requests for the SCTP protocol (PREFER_FAILURE is implied)
* PEER requests

See Section 13.2 for processing details. The error lifetime depends on the reason for the failure.

12 ADDRESS_MISMATCH: The source IP address of the request packet does not match the contents of the PCP Client's IP Address field, due to an unexpected NAT on the path between the PCP client and the PCP-controlled NAT or firewall. This is a long lifetime error.

13 EXCESSIVE_REMOTE_PEERS: The PCP server was not able to create the filters in this request. This result code MUST only be returned if the MAP request contained the FILTER Option. See Section 13.3 for processing information. This is a long lifetime error.

8. General PCP Operation

PCP messages MUST be sent over UDP [RFC0768]. Every PCP request generates at least one response, so PCP does not need to run over a reliable transport protocol.

When receiving multiple identical requests, the PCP server will generate identical responses, provided the PCP server's state did not change between those requests due to other activity. For example, if a request is received while the PCP-controlled device has no mappings available, it will generate an error response. If mappings become available and then a (duplicated or re-transmitted) request is seen by the server, it will generate a non-error response. A PCP client MUST handle such updated responses for any request it sends, most notably to support Rapid Recovery (Section 14). Also see the Protocol Design Note (Section 6).

8.1. General PCP Client: Generating a Request

This section details operation specific to a PCP client, for any Opcode. Procedures specific to the MAP Opcode are described in Section 11, and procedures specific to the PEER Opcode are described in Section 12.

Prior to sending its first PCP message, the PCP client determines which server to use. The PCP client performs the following steps to determine its PCP server:

1. if a PCP server is configured (e.g., in a configuration file or via DHCP), that single configuration source is used as the list of PCP Server(s), else;

2. the default router list (for IPv4 and IPv6) is used as the list of PCP Server(s). Thus, if a PCP client has both an IPv4 and IPv6 address, it will have an IPv4 PCP server (its IPv4 default router) for its IPv4 mappings, and an IPv6 PCP server (its IPv6 default router) for its IPv6 mappings.

For the purposes of this document, only a single PCP server address is supported. Should future specifications define configuration methods that provide a longer list of PCP server addresses, those specifications will define how clients select one or more addresses from that list.

With that PCP server address, the PCP client formulates its PCP request. The PCP request contains a PCP common header, PCP Opcode and payload, and (possibly) Options. As with all UDP client software on any operating system, when several independent PCP clients exist on the same host, each uses a distinct source port number to disambiguate their requests and replies. The PCP client's source port SHOULD be randomly generated [RFC6056].

The PCP client MUST include the source IP address of the PCP message in the PCP request. This is typically its own IP address; see

Section 16.4 for how this can be coded. This is used to detect an unexpected NAT on the path between the PCP client and the PCP-controlled NAT or firewall device, to avoid wasting state on the PCP-controlled NAT creating pointless non-functional mappings. When such an intervening non-PCP-aware inner NAT is detected, mappings must first be created by some other means in the inner NAT, before mappings can be usefully created in the outer PCP-controlled NAT. Having created mappings in the inner NAT by some other means, the PCP client should then use the inner NAT's External Address as the Client IP Address, to signal to the outer PCP-controlled NAT that the client is aware of the inner NAT, and has taken steps to create mappings in it by some other means, so that mappings created in the outer NAT will not be a pointless waste of state.

8.1.1. PCP Client Retransmission

PCP clients are responsible for reliable delivery of PCP request messages. If a PCP client fails to receive an expected response from a server, the client must retransmit its message. The retransmissions MUST use the same Mapping Nonce value (see Section 11.1 and Section 12.1). The client begins the message exchange by transmitting a message to the server. The message exchange continues for as long as the client wishes to maintain the mapping, and terminates when the PCP client is no longer interested in the PCP transaction (e.g., the application that requested the mapping is no longer interested in the mapping) or (optionally) when the message exchange is considered to have failed according to the retransmission mechanism described below.

The client retransmission behavior is controlled and described by the following variables:

RT: Retransmission timeout, calculated as described below

IRT: Initial retransmission time, SHOULD be 3 seconds

MRC: Maximum retransmission count, SHOULD be 0 (0 indicates no maximum)

MRT: Maximum retransmission time, SHOULD be 1024 seconds

MRD: Maximum retransmission duration, SHOULD be 0 (0 indicates no maximum)

RAND: Randomization factor, calculated as described below

With each message transmission or retransmission, the client sets RT according to the rules given below. If RT expires before a response

is received, the client recomputes RT and retransmits the request.

Each of the computations of a new RT include a new randomization factor (RAND), which is a random number chosen with a uniform distribution between -0.1 and +0.1. The randomization factor is included to minimize synchronization of messages transmitted by PCP clients. The algorithm for choosing a random number does not need to be cryptographically sound. The algorithm SHOULD produce a different sequence of random numbers from each invocation of the PCP client.

The RT value is initialized based on IRT:

```
RT = (1 + RAND) * IRT
```

RT for each subsequent message transmission is based on the previous value of RT, subject to the upper bound on the value of RT specified by MRT. If MRT has a value of 0, there is no upper limit on the value of RT, and MRT is treated as "infinity":

```
RT = (1 + RAND) * MIN (2 * RTprev, MRT)
```

MRC specifies an upper bound on the number of times a client may retransmit a message. Unless MRC is zero, the message exchange fails once the client has transmitted the message MRC times.

MRD specifies an upper bound on the length of time a client may retransmit a message. Unless MRD is zero, the message exchange fails once MRD seconds have elapsed since the client first transmitted the message.

If both MRC and MRD are non-zero, the message exchange fails whenever either of the conditions specified in the previous two paragraphs are met. If both MRC and MRD are zero, the client continues to transmit the message until it receives a response, or the client no longer wants a mapping.

Once a PCP client has successfully received a response from a PCP server on that interface, it resets RT to a value randomly selected in the range 1/2 to 5/8 of the mapping lifetime, as described in Section 11.2.1, and sends subsequent PCP requests for that mapping to that same server.

> Note: If the server’s state changes between retranmissions and the server’s response is delayed or lost, the state in the PCP client and server may not be synchronized. This is not unique to PCP, but also occurs with other network protocols (e.g., TCP). In the unlikely event that such de-synchronization occurs, PCP heals itself after Lifetime seconds.

8.2. General PCP Server: Processing a Request

This section details operation specific to a PCP server. Processing SHOULD be performed in the order of the following paragraphs.

A PCP server MUST only accept normal (non-THIRD_PARTY) PCP requests from a client on the same interface it would normally receive packets from that client, and MUST silently ignore PCP requests arriving on any other interface. For example, a residential NAT gateway accepts PCP requests only when they arrive on its (LAN) interface connecting to the internal network, and silently ignores any PCP requests arriving on its external (WAN) interface. A PCP server which supports THIRD_PARTY requests MAY be configured to accept THIRD_PARTY requests on other configured interfaces (see Section 13.1).

Upon receiving a request, the PCP server parses and validates it. A valid request contains a valid PCP common header, one valid PCP Opcode, and zero or more Options (which the server might or might not comprehend). If an error is encountered during processing, the server generates an error response which is sent back to the PCP client. Processing an Opcode and the Options are specific to each Opcode.

Error responses have the same packet layout as success responses, with certain fields from the request copied into the response, and other fields assigned by the PCP server set as indicated in Figure 3.

Copying request fields into the response is important because this is what enables a client to identify to which request a given response pertains. For Opcodes that are understood by the PCP server, it follows the requirements of that Opcode to copy the appropriate fields. For Opcodes that are not understood by the PCP server, it simply generates the UNSUPP_OPCODE response and copies fields from the PCP header and copies the rest of the PCP payload as-is (without attempting to interpret it).

All responses (both error and success) contain the same Opcode as the request, but with the "R" bit set.

Any error response has a nonzero Result Code, and is created by:

- o Copying the entire UDP payload, or 1100 octets, whichever is less, and zero-padding the response to a multiple of 4 octets if necessary
- o Setting the R bit
- o Setting the Result Code
- o Setting the Lifetime, Epoch Time and Reserved fields
- o Updating other fields in the response, as indicated by 'set by the server' in the PCP response field description.

A success response has a zero Result Code, and is created by:

- o Copying the first four octets of request packet header
- o Setting the R bit
- o Setting the Result Code to zero
- o Setting the Lifetime, Epoch Time and Reserved fields
- o Possibly setting opcode-specific response data if appropriate
- o Adding any processed options to the response message

If the received PCP request message is less than two octets long it is silently dropped.

If the R bit is set the message is silently dropped.

If the first octet (version) is a version that is not supported, a response is generated with the UNSUPP_VERSION result code, and the other steps detailed in Section 9 are followed.

Otherwise, if the version is supported but the received message is shorter than 24 octets, the message is silently dropped.

If the server is overloaded by requests (from a particular client or from all clients), it MAY simply silently discard requests, as the requests will be retried by PCP clients, or it MAY generate the NO_RESOURCES error response.

If the length of the message exceeds 1100 octets, is not a multiple of 4 octets, or is too short for the opcode in question, it is invalid and a MALFORMED_REQUEST response is generated, and the response message is truncated to 1100 octets.

The PCP server compares the source IP address (from the received IP header) with the field PCP Client IP Address. If they do not match, the error ADDRESS_MISMATCH MUST be returned. This is done to detect and prevent accidental use of PCP where a non-PCP-aware NAT exists between the PCP client and PCP server. If the PCP client wants such a mapping it needs to ensure the PCP field matches its apparent IP address from the perspective of the PCP server.

8.3. General PCP Client: Processing a Response

The PCP client receives the response and verifies that the source IP address and port belong to the PCP server of a previously-sent PCP request. If not, the response is silently dropped.

If the received PCP response message is less than four octets long it is silently dropped.

If the R bit is clear the message is silently dropped.

If the error code is UNSUPP_VERSION processing continues as described in Section 9.

The PCP client then validates that the Opcode matches a previous PCP request. Responses shorter than 24 octets, longer than 1100 octets, or not a multiple of 4 octets are invalid and ignored, likely causing the request to be re-transmitted. The response is further matched by comparing fields in the response Opcode-specific data to fields in the request Opcode-specific data, as described by the processing for that Opcode.

After these matches are successful, the PCP client checks the Epoch Time field to determine if it needs to restore its state to the PCP server (see Section 8.5). A PCP client SHOULD be prepared to receive multiple responses from the PCP Server at any time after a single request is sent. This allows the PCP server to inform the client of mapping changes such as an update or deletion. For example, a PCP Server might send a SUCCESS response and, after a configuration change on the PCP Server, later send a NOT_AUTHORIZED response. A PCP client MUST be prepared to receive responses for requests it never sent (which could have been sent by a previous PCP instance on this same host, or by a previous host that used the same client IP address, or by a malicious attacker) by simply ignoring those unexpected messages.

If the error ADDRESS_MISMATCH is received, it indicates the presence of a NAT between the PCP client and PCP server. Procedures to resolve this problem are beyond the scope of this document.

For both success and error responses a Lifetime value is returned. The Lifetime indicates how long this request is considered valid by the server. The PCP client SHOULD impose an upper limit on this returned value (to protect against absurdly large values, e.g., 5 years), detailed in Section 15.

If the result code is 0 (SUCCESS), the request succeeded.

If the result code is not 0, the request failed, and the PCP client SHOULD NOT resend the same request for the indicated Lifetime of the error (as limited by the sanity checking detailed in Section 15).

If the PCP client has discovered a new PCP server (e.g., connected to a new network), the PCP client MAY immediately begin communicating with this PCP server, without regard to hold times from communicating with a previous PCP server.

8.4. Multi-Interface Issues

Hosts that desire a PCP mapping might be multi-interfaced (i.e., own several logical/physical interfaces). Indeed, a host can be configured with several IPv4 addresses (e.g., Wi-Fi and Ethernet) or dual-stacked. These IP addresses may have distinct reachability scopes (e.g., if IPv6 they might have global reachability scope as for Global Unicast Address (GUA, [RFC3587]) or limited scope as for Unique Local Address (ULA) [RFC4193]).

IPv6 addresses with global reachability (e.g., GUA) SHOULD be used as the source address when generating a PCP request. IPv6 addresses without global reachability (e.g., ULA [RFC4193]), SHOULD NOT be used as the source interface when generating a PCP request. If IPv6 privacy addresses [RFC4941] are used for PCP mappings, a new PCP request will need to be issued whenever the IPv6 privacy address is changed. This PCP request SHOULD be sent from the IPv6 privacy address itself. It is RECOMMENDED that the client delete its mappings to the previous privacy address after it no longer needs those old mappings.

Due to the ubiquity of IPv4 NAT, IPv4 addresses with limited scope (e.g., private addresses [RFC1918]) MAY be used as the source interface when generating a PCP request.

8.5. Epoch

Every PCP response sent by the PCP server includes an Epoch time field. This time field increments by one every second. Anomalies in the received Epoch time value provide a hint to PCP clients that a PCP server state loss may have occurred. Clients respond to such state loss hints by promptly renewing their mappings, so as to quickly restore any lost state at the PCP server.

If the PCP server resets or loses the state of its explicit dynamic Mappings (that is, those mappings created by PCP requests), due to reboot, power failure, or any other reason, it MUST reset its Epoch time to its initial starting value (usually zero) to provide this hint to PCP clients. After resetting its Epoch time, the PCP server

resumes incrementing the Epoch time value by one every second. Similarly, if the External IP Address(es) of the NAT (controlled by the PCP server) changes, the Epoch time MUST be reset. A PCP server MAY maintain one Epoch time value for all PCP clients, or MAY maintain distinct Epoch time values (per PCP client, per interface, or based on other criteria); this choice is implementation-dependent.

Whenever a client receives a PCP response, the client validates the received Epoch time value according to the procedure below, using integer arithmetic:

- o If this is the first PCP response the client has received from this PCP server, the Epoch time value is treated as necessarily valid, otherwise

  - * If the current PCP server Epoch time (curr_server_time) is less than the previously received PCP server Epoch time (prev_server_time) by more than one second, then the client treats the Epoch time as obviously invalid (time should not go backwards). The server Epoch time apparently going backwards by *up to* one second is not deemed invalid, so that minor packet re-ordering on the path from PCP Server to PCP Client does not trigger a cascade of unnecessary mapping renewals. If the server Epoch time passes this check, then further validation checks are performed:

    - + The client computes the difference between its current local time (curr_client_time) and the time the previous PCP response was received from this PCP server (prev_client_time):
      client_delta = curr_client_time - prev_client_time;

    - + The client computes the difference between the current PCP server Epoch time (curr_server_time) and the previously received Epoch time (prev_server_time):
      server_delta = curr_server_time - prev_server_time;

    - + If client_delta+2 < server_delta - server_delta/16
      or server_delta+2 < client_delta - client_delta/16
      then the client treats the Epoch time value as invalid,
      else the client treats the Epoch time value as valid

- o The client records the current time values for use in its next comparison:
  prev_client_time = curr_client_time
  prev_server_time = curr_server_time

If the PCP client determined that the Epoch time value it received

was invalid then it concludes that the PCP server may have lost state, and promptly renews all its active port mapping leases as described in Section 16.3.1.

Notes:

- o The client clock MUST never go backwards. If curr_client_time is found to be less than prev_client_time then this is a client bug, and how the client deals with this client bug is implementation specific.

- o The calculations above are constructed to allow client_delta and server_delta to be computed as unsigned integer values.

- o The "+2" in the calculations above is to accommodate quantization errors in client and server clocks (up to one second quantization error each in server and client time intervals).

- o The "/16" in the calculations above is to accommodate inaccurate clocks in low-cost devices. This allows for a total discrepancy of up to 1/16 (6.25%) to be considered benign, e.g., if one clock were to run too fast by 3% while the other clock ran too slow by 3% then the client would not consider this difference to be anomalous or indicative of a restart having occurred. This tolerance is strict enough to be effective at detecting reboots, while not being so strict as to generate false alarms.

9. Version Negotiation

A PCP client sends its requests using PCP version number 2. Should later updates to this document specify different message formats with a version number greater than 2 it is expected that PCP servers will still support version 2 in addition to the newer version(s). However, in the event that a server returns a response with result code UNSUPP_VERSION, the client MAY log an error message to inform the user that it is too old to work with this server.

Should later updates to this document specify different message formats with a version number greater than 2, and backwards compatibility is desired, this first octet can be used for forward and backward compatibility.

If future PCP versions greater than 2 are specified, version negotiation proceeds as follows:

1. The client sends its first request using the highest (i.e., presumably ’best’) version number it supports.

2. If the server supports that version it responds normally.

3. If the server does not support that version it replies giving a result containing the result code UNSUPP_VERSION, and the closest version number it does support (if the server supports a range of versions higher than the client's requested version, the server returns the lowest of that supported range; if the server supports a range of versions lower than the client's requested version, the server returns the highest of that supported range).

4. If the client receives an UNSUPP_VERSION result containing a version it does support, it records this fact and proceeds to use this message version for subsequent communication with this PCP server (until a possible future UNSUPP_VERSION response if the server is later updated, at which point the version negotiation process repeats).

5. If the client receives an UNSUPP_VERSION result containing a version it does not support then the client SHOULD try the next-lower version supported by the client. The attempt to use the next-lower version repeats until the client has tried version 2. If using version 2 fails, the client MAY log an error message to inform the user that it is too old to work with this server, and the client SHOULD set a timer to retry its request in 30 minutes or the returned Lifetime value, whichever is smaller. By automatically retrying in 30 minutes, the protocol accommodates an upgrade of the PCP server.

10. Introduction to MAP and PEER Opcodes

There are four uses for the MAP and PEER Opcodes defined in this document:

- a host operating a server and wanting an incoming connection (Section 10.1);
- a host operating a client and server on the same port (Section 10.2);
- a host operating a client and wanting to optimize the application keepalive traffic (Section 10.3);
- and a host operating a client and wanting to restore lost state in its NAT (Section 10.4).

These are discussed in the following sections, and a (non-normative) state diagram is provided in Section 16.5.

When operating a server (Section 10.1 and Section 10.2) the PCP client knows if it wants an IPv4 listener, IPv6 listener, or both on the Internet. The PCP client also knows if it has an IPv4 address or IPv6 address configured on one of its interfaces. It takes the union of this knowledge to decide to which of its PCP servers to send the request (e.g., an IPv4 address or an IPv6 address), and if to send one or two MAP requests for each of its interfaces (e.g., if the PCP client has only an IPv4 address but wants both IPv6 and IPv4 listeners, it sends a MAP request containing the all-zeros IPv6 address in the Suggested External Address field, and sends a second MAP request containing the all-zeros IPv4 address in the Suggested External Address field. If the PCP client has both an IPv4 and IPv6 address, and only wants an IPv4 listener, it sends one MAP request from its IPv4 address (if the PCP server supports NAT44 or IPv4 firewall) or one MAP request from its IPv6 address (if the PCP server supports NAT64). The PCP client can simply request the desired mapping to determine if the PCP server supports the desired mapping. Applications that embed IP addresses in payloads (e.g., FTP, SIP) will find it beneficial to avoid address family translation, if possible.

The MAP and PEER requests include a Suggested External IP Address field. Some PCP-controlled devices, especially CGN but also multi-homed NPTv6 networks, have a pool of public-facing IP addresses. PCP allows the client to indicate if it wants a mapping assigned on a specific address of that pool or any address of that pool. Some applications will break if mappings are created on different IP addresses (e.g., active mode FTP), so applications should carefully consider the implications of using this capability. Static mappings for that Internal Address (e.g., those created by a command-line interface on the PCP server or PCP-controlled device) may exist to a certain External Address, and if the Suggested External IP Address is the all-zeros address, PCP SHOULD assign its mappings to the same External Address, as this can also help applications using a mix of both static mappings and PCP-created mappings. If, on the other hand, the Suggested External IP Address contains a non-zero IP address the PCP Server SHOULD create a mapping to that external address, even if there are other mappings from that same Internal Address to a different External Address. Once an Internal Address has no implicit dynamic mappings and no explicit dynamic mappings in the PCP-controlled device, a subsequent implicit or explicit mapping for that Internal Address MAY be assigned to a different External Address. Generally, this re-assignment would occur when a CGN device is load balancing newly-seen Internal Addresses to its public pool of External Addresses.

The following table summarizes how various common PCP deployments use IPv6 and IPv4 addresses.

The 'internal' address is implicitly the same as the source IP address of the PCP request, except when the THIRD_PARTY option is used.

The 'external' address is the Suggested External Address field of the MAP or PEER request, and is address family is usually the same as the 'internal' address family, except when technologies like NAT64 are used.

The 'remote peer' address is the Remote Peer IP Address of the PEER request or the FILTER option of the MAP request, and is always the same address family as the 'internal' address, even when NAT64 is used.

In NAT64, the IPv6 PCP client is not necessarily aware of the NAT64 or aware of the actual IPv4 address of the remote peer, so it expresses the IPv6 address from its perspective, as shown in the table.

	internal	external	PCP remote peer	actual remote peer
IPv4 firewall	IPv4	IPv4	IPv4	IPv4
IPv6 firewall	IPv6	IPv6	IPv6	IPv6
NAT44	IPv4	IPv4	IPv4	IPv4
NAT46	IPv4	IPv6	IPv4	IPv6
NAT64	IPv6	IPv4	IPv6	IPv4
NPTv6	IPv6	IPv6	IPv6	IPv6

Figure 5: Address Families with MAP and PEER

10.1. For Operating a Server

A host operating a server (e.g., a web server) listens for traffic on a port, but the server never initiates traffic from that port. For this to work across a NAT or a firewall, the host needs to (a) create a mapping from a public IP address, protocol, and port to itself as described in Section 11, (b) publish that public IP address, protocol, and port via some sort of rendezvous server (e.g., DNS, a SIP message, a proprietary protocol), and (c) ensure that any other non-PCP-speaking packet filtering middleboxes on the path (e.g., host-based firewall, network-based firewall, or other NATs) will also allow the incoming traffic. Publishing the public IP address and port is out of scope of this specification. To accomplish (a), the host follows the procedures described in this section.

As normal, the application needs to begin listening on a port. Then, the application constructs a PCP message with the MAP Opcode, with the external address set to the appropriate all-zeroes address, depending on whether it wants a public IPv4 or IPv6 address.

The following pseudo-code shows how PCP can be reliably used to operate a server:

```
 /* start listening on the local server port */
 int s = socket(...);
 bind(s, ...);
 listen(s, ...);

 getsockname(s, &internal_sockaddr, ...);
 bzero(&external_sockaddr, sizeof(external_sockaddr));

 while (1)
     {
     /* Note: The "time_to_send_pcp_request()" check below includes:
      * 1. Sending the first request
      * 2. Retransmitting requests due to packet loss
      * 3. Resending a request due to impending lease expiration
      * 4. Resending a request due to server state loss
      * The PCP packet sent is identical in all four cases; from
      * the PCP server’s point of view they are the same operation.
      * The Suggested External Address and Port may be updated
      * repeatedly during the lifetime of the mapping.
      * Other fields in the packet generally remain unchanged.
      */
     if (time_to_send_pcp_request())
         pcp_send_map_request(internal_sockaddr.sin_port,
             internal_sockaddr.sin_addr,
             &external_sockaddr, /* will be zero the first time */
             requested_lifetime, &assigned_lifetime);

     if (pcp_response_received())
         update_rendezvous_server("Client Ident", external_sockaddr);

     if (received_incoming_connection_or_packet())
         process_it(s);

     if (other_work_to_do())
         do_it();

     /* ... */

     block_until_we_need_to_do_something_else();
     }
```

Figure 6: Pseudo-code for using PCP to operate a server

10.2. For Operating a Symmetric Client/Server

A host operating a client and server on the same port (e.g., Symmetric RTP [RFC4961] or SIP Symmetric Response Routing (rport) [RFC3581]) first establishes a local listener, (usually) sends the local and public IP addresses, protocol, and ports to a rendezvous service (which is out of scope of this document), and initiates an outbound connection from that same source address and same port. To accomplish this, the application uses the procedure described in this section.

An application that is using the same port for outgoing connections as well as incoming connections MUST first signal its operation of a server using the PCP MAP Opcode, as described in Section 11, and receive a positive PCP response before it sends any packets from that port.

Discussion: In general, a PCP client doesn’t know in advance if it is behind a NAT or firewall. On detecting the host has connected to a new network, the PCP client can attempt to request a mapping using PCP, and if that succeeds then the client knows it has successfully created a mapping. If after multiple retries it has received no PCP response, then either the client is *not* behind a NAT or firewall and has unfettered connectivity, or the client *is* behind a NAT or firewall which doesn’t support PCP (and the client may still have working connectivity by virtue of static mappings previously created manually by the user). Retransmitting PCP requests multiple times before giving up and assuming unfettered connectivity adds delay in that case. Initiating outbound TCP connections immediately without waiting for PCP avoids this delay, and will work if the NAT has endpoint-independent mapping EIM behavior, but may fail if the NAT has endpoint-dependent mapping EDM behavior. Waiting enough time to allow an explicit PCP MAP Mapping to be created (if possible) first ensures that the same External Port will then be used for all subsequent implicit dynamic mappings (e.g., TCP SYNs) sent from the specified Internal Address, Protocol, and Port. PCP supports both EIM and EDM NATs, so clients need to assume they may be dealing with an EDM NAT. In this case, the client will experience more reliable connectivity if it attempts explicit PCP MAP requests first, before initiating any outbound TCP connections from that Internal Address and Port. See also Section 16.1.

The following pseudo-code shows how PCP can be used to operate a symmetric client and server:

```
/* start listening on the local server port */
int s = socket(...);
bind(s, ...);
listen(s, ...);

getsockname(s, &internal_sockaddr, ...);
bzero(&external_sockaddr, sizeof(external_sockaddr));

while (1)
    {
    /* Note: The "time_to_send_pcp_request()" check below includes:
     * 1. Sending the first request
     * 2. Retransmitting requests due to packet loss
     * 3. Resending a request due to impending lease expiration
     * 4. Resending a request due to server state loss
     */
    if (time_to_send_pcp_request())
        pcp_send_map_request(internal_sockaddr.sin_port,
            internal_sockaddr.sin_addr,
            &external_sockaddr, /* will be zero the first time */
            requested_lifetime, &assigned_lifetime);

    if (pcp_response_received())
        update_rendezvous_server("Client Ident", external_sockaddr);

    if (received_incoming_connection_or_packet())
        process_it(s);

    if (need_to_make_outgoing_connection())
        make_outgoing_connection(s, ...);

    if (data_to_send())
        send_it(s);

    if (other_work_to_do())
        do_it();

    /* ... */

    block_until_we_need_to_do_something_else();
    }
```

Figure 7: Pseudo-code for using PCP to operate a symmetric client/server

10.3. For Reducing NAT or Firewall Keepalive Messages

A host operating a client (e.g., XMPP client, SIP client) sends from a port, and may receive responses, but never accepts incoming connections from other Remote Peers on this port. It wants to ensure the flow to its Remote Peer is not terminated (due to inactivity) by an on-path NAT or firewall. To accomplish this, the application uses the procedure described in this section.

Middleboxes such as NATs or firewalls need to see occasional traffic or will terminate their session state, causing application failures. To avoid this, many applications routinely generate keepalive traffic for the primary (or sole) purpose of maintaining state with such middleboxes. Applications can reduce such application keepalive traffic by using PCP.

> Note: For reasons beyond NAT, an application may find it useful to perform application-level keepalives, such as to detect a broken path between the client and server, keep state alive on the Remote Peer, or detect a powered-down client. These keepalives are not related to maintaining middlebox state, and PCP cannot do anything useful to reduce those keepalives.

To use PCP for this function, the application first connects to its server, as normal. Afterwards, it issues a PCP request with the PEER Opcode as described in Section 12.

The following pseudo-code shows how PCP can be reliably used with a dynamic socket, for the purposes of reducing application keepalive messages:

```
 int s = socket(...);
 connect(s, &remote_peer, ...);

 getsockname(s, &internal_sockaddr, ...);
 bzero(&external_sockaddr, sizeof(external_sockaddr));

 while (1)
     {
     /* Note: The "time_to_send_pcp_request()" check below includes:
      * 1. Sending the first request
      * 2. Retransmitting requests due to packet loss
      * 3. Resending a request due to impending lease expiration
      * 4. Resending a request due to server state loss
      */
     if (time_to_send_pcp_request())
         pcp_send_peer_request(internal_sockaddr.sin_port,
             internal_sockaddr.sin_addr,
             &external_sockaddr, /* will be zero the first time */
             remote_peer, requested_lifetime, &assigned_lifetime);

     if (data_to_send())
         send_it(s);

     if (other_work_to_do())
         do_it();

     /* ... */

     block_until_we_need_to_do_something_else();
     }
```

Figure 8: Pseudo-code using PCP with a dynamic socket

10.4. For Restoring Lost Implicit TCP Dynamic Mapping State

After a NAT loses state (e.g., because of a crash or power failure), it is useful for clients to re-establish TCP mappings on the NAT. This allows servers on the Internet to see traffic from the same IP address and port, so that sessions can be resumed exactly where they were left off. This can be useful for long-lived connections (e.g., instant messaging) or for connections transferring a lot of data (e.g., FTP). This can be accomplished by first establishing a TCP connection normally and then sending a PEER request/response and remembering the External Address and External Port. Later, when the

NAT has lost state, the client can send a PEER request with the Suggested External Port and Suggested External Address remembered from the previous session, which will create a mapping in the NAT that functions exactly as an implicit dynamic mapping. The client then resumes sending TCP data to the server.

> Note: This procedure works well for TCP, provided the NAT creates a new implicit dynamic outbound mapping only for TCP segments with the SYN bit set (i.e., the newly-booted NAT drops the re-transmitted data segments from the client because the NAT does not have an active mapping for those segments), and if the server is not sending data that elicits a RST from the NAT. This is not the case for UDP, because a new UDP mapping will be created (probably on a different port) as soon as UDP traffic is seen by the NAT.

11. MAP Opcode

This section defines an Opcode which controls forwarding from a NAT (or firewall) to an Internal Host.

MAP: Create an explicit dynamic mapping between an Internal Address + Port and an External Address + Port.

PCP Servers SHOULD provide a configuration option to allow administrators to disable MAP support if they wish.

Mappings created by PCP MAP requests are, by definition, Endpoint Independent Mappings (EIM) with Endpoint Independent Filtering (EIF) (unless the FILTER Option is used), even on a NAT that usually creates Endpoint Dependent Mappings (EDM) or Endpoint Dependent Filtering (EDF) for outgoing connections, since the purpose of an (unfiltered) MAP mapping is to receive inbound traffic from any remote endpoint, not from only one specific remote endpoint.

Note also that all NAT mappings (created by PCP or otherwise) are by necessity bidirectional and symmetric. For any packet going in one direction (in or out) that is translated by the NAT, a reply going in the opposite direction needs to have the corresponding opposite translation done so that the reply arrives at the right endpoint. This means that if a client creates a MAP mapping, and then later sends an outgoing packet using the mapping’s Internal Address, Protocol and Port, the NAT should translate that packet’s Internal Address and Port to the mapping’s External Address and Port, so that replies addressed to the External Address and Port are correctly translated back to the mapping’s Internal Address and Port.

On Operating Systems that allow multiple listening servers to bind to

the same internal address, protocol and port, servers MUST ensure that they have exclusive use of that internal address, protocol and port (e.g., by binding the port using INADDR_ANY, or using SO_EXCLUSIVEADDRUSE or similar) before sending their PCP MAP request, to ensure that no other PCP clients on the same machine are also listening on the same internal protocol and internal port.

As a side-effect of creating a mapping, ICMP messages associated with the mapping MUST be forwarded (and also translated, if appropriate) for the duration of the mapping's lifetime. This is done to ensure that ICMP messages can still be used by hosts, without application programmers or PCP client implementations needing to use PCP separately to create ICMP mappings for those flows.

The operation of the MAP Opcode is described in this section.

11.1. MAP Operation Packet Formats

The MAP Opcode has a similar packet layout for both requests and responses. If the Assigned External IP address and Port in the PCP response always match the Internal IP Address and Port from the PCP request, then the functionality is purely a firewall; otherwise it pertains to a network address translator which might also perform firewall-like functions.

The following diagram shows the format of the Opcode-specific information in a request for the MAP Opcode.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                 Mapping Nonce (96 bits)                       |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Protocol    |          Reserved (24 bits)                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        Internal Port          |    Suggested External Port    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|           Suggested External IP Address (128 bits)            |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 9: MAP Opcode Request

These fields are described below:

Requested lifetime (in common header): Requested lifetime of this mapping, in seconds. The value 0 indicates "delete".

Mapping Nonce: Random value chosen by the PCP client. See Section 11.2. Zero is a legal value (but unlikely, occurring in roughly one in 2^96 requests).

Protocol: Upper-layer protocol associated with this Opcode. Values are taken from the IANA protocol registry [proto_numbers]. For example, this field contains 6 (TCP) if the Opcode is intended to create a TCP mapping. The value 0 has a special meaning for 'all protocols'.

Reserved: 24 reserved bits, MUST be sent as 0 and MUST be ignored when received.

Internal Port: Internal port for the mapping. The value 0 indicates 'all ports', and is legal when the lifetime is zero (a delete request), if the Protocol does not use 16-bit port numbers, or the client is requesting 'all ports'. If Protocol is zero (meaning 'all protocols'), then Internal Port MUST be zero on transmission and MUST be ignored on reception.

Suggested External Port: Suggested external port for the mapping. This is useful for refreshing a mapping, especially after the PCP server loses state. If the PCP client does not know the external port, or does not have a preference, it MUST use 0.

Suggested External IP Address: Suggested external IPv4 or IPv6 address. This is useful for refreshing a mapping, especially after the PCP server loses state. If the PCP client does not know the external address, or does not have a preference, it MUST use the address-family-specific all-zeroes address (see Section 5).

The internal address for the request is the source IP address of the PCP request message itself, unless the THIRD_PARTY Option is used.

The following diagram shows the format of Opcode-specific information in a response packet for the MAP Opcode:

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                 Mapping Nonce (96 bits)                       |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Protocol    |          Reserved (24 bits)                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        Internal Port          |    Assigned External Port     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|            Assigned External IP Address (128 bits)            |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 10: MAP Opcode Response

These fields are described below:

Lifetime (in common header): On an error response, this indicates how long clients should assume they'll get the same error response from the PCP server if they repeat the same request. On a success response, this indicates the lifetime for this mapping, in seconds.

Mapping Nonce: Copied from the request.

Protocol: Copied from the request.

Reserved: 24 reserved bits, MUST be sent as 0 and MUST be ignored when received.

Internal Port: Copied from the request.

Assigned External Port: On a success response, this is the assigned external port for the mapping. On an error response, the Suggested External Port is copied from the request.

Assigned External IP Address: On a success response, this is the assigned external IPv4 or IPv6 address for the mapping. An IPv4 address is encoded using IPv4-mapped IPv6 address. On an error response, the Suggested External IP Address is copied from the request.

11.2. Generating a MAP Request

This section describes the operation of a PCP client when sending requests with the MAP Opcode.

The request MAY contain values in the Suggested External Port and Suggested External IP Address fields. This allows the PCP client to attempt to rebuild lost state on the PCP server, which improves the chances of existing connections surviving, and helps the PCP client avoid having to change information maintained at its rendezvous server. Of course, due to other activity on the network (e.g., by other users or network renumbering), the PCP server may not be able to grant the suggested External IP Address, Protocol, and Port, and in that case it will assign a different External IP Address and Port.

A PCP client MUST be written assuming that it may *never* be assigned the external port it suggests. In the case of recreating state after a NAT gateway crash, the Suggested External Port, being one that was previously allocated to this client, is likely to be available for this client to continue using. In all other cases, the client MUST assume that it is unlikely that its Suggested External Port will be granted. For example, when many subscribers are sharing a Carrier-Grade NAT, popular ports such as 80, 443 and 8080 are likely to be in high demand. At most one client can have each of those popular ports for each External IP Address, and all the other clients will be assigned other, dynamically allocated, External Ports. Indeed, some ISPs may, by policy, choose not to grant those External Ports to *anyone*, so that none of their clients are *ever* assigned External Ports 80, 443 or 8080.

If the Protocol does not use 16-bit port numbers (e.g., RSVP, IP protocol number 46), the port number MUST be zero. This will cause all traffic matching that protocol to be mapped.

If the client wants all protocols mapped it uses Protocol 0 (zero) and Internal Port 0 (zero).

The Mapping Nonce value is randomly chosen by the PCP client, following accepted practices for generating unguessable random numbers [RFC4086], and is used as part of the validation of PCP responses (see below) by the PCP client, and validation for mapping refreshes by the PCP server. The client MUST use a different Mapping Nonce for each PCP server it communicates with, and it is RECOMMENDED to choose a new random Mapping Nonce whenever the PCP client is initialized. The client MAY use a different Mapping Nonce for every mapping.

11.2.1. Renewing a Mapping

An existing mapping can have its lifetime extended by the PCP client. To do this, the PCP client sends a new MAP request indicating the internal port. The PCP MAP request SHOULD also include the currently assigned external IP address and port in the Suggested External IP address and Suggested External Port fields, so if the PCP server has lost state it can recreate the lost mapping with the same parameters.

The PCP client SHOULD renew the mapping before its expiry time, otherwise it will be removed by the PCP server (see Section 15). To reduce the risk of inadvertent synchronization of renewal requests, a random jitter component should be included. It is RECOMMENDED that PCP clients send a single renewal request packet at a time chosen with uniform random distribution in the range 1/2 to 5/8 of expiration time. If no SUCCESS response is received, then the next renewal request should be sent 3/4 to 3/4 + 1/16 to expiration, and then another 7/8 to 7/8 + 1/32 to expiration, and so on, subject to the constraint that renewal requests MUST NOT be sent less than four seconds apart (a PCP client MUST NOT send a flood of ever-closer-together requests in the last few seconds before a mapping expires).

11.3. Processing a MAP Request

This section describes the operation of a PCP server when processing a request with the MAP Opcode. Processing SHOULD be performed in the order of the following paragraphs.

The Protocol, Internal Port, and Mapping Nonce fields from the MAP request are copied into the MAP response. If present and processed by the PCP server the THIRD_PARTY Option is also copied into the MAP response.

If the Requested Lifetime is non-zero then:

- o If both the protocol and internal port are non-zero, it indicates a request to create a mapping or extend the lifetime of an existing mapping. If the PCP server or PCP-controlled device does not support the Protocol, the UNSUPP_PROTOCOL error MUST be returned.

- o If the protocol is non-zero and the internal port is zero, it indicates a request to create or extend a mapping for all incoming traffic for that entire Protocol. If this request cannot be fulfilled in its entirety, the UNSUPP_PROTOCOL error MUST be returned.

- o If both the protocol and internal port are zero, it indicates a request to create or extend a mapping for all incoming traffic for all protocols (commonly called a "DMZ host"). If this request cannot be fulfilled in its entirety, the UNSUPP_PROTOCOL error MUST be returned.

- o If the protocol is zero and the internal port is non-zero, then the request is invalid and the PCP Server MUST return a MALFORMED_REQUEST error to the client.

If the requested lifetime is zero, it indicates a request to delete an existing mapping.

Further processing of the lifetime is described in Section 15.

If operating in the Simple Threat Model (Section 18.1), and the Internal port, Protocol, and Internal Address match an existing explicit dynamic mapping, but the Mapping Nonce does not match, the request MUST be rejected with a NOT_AUTHORIZED error with the Lifetime of the error indicating duration of that existing mapping. The PCP server only needs to remember one Mapping Nonce value for each explicit dynamic mapping.

If the Internal port, Protocol, and Internal Address match an existing static mapping (which will have no nonce) then a PCP reply is sent giving the External Address and Port of that static mapping, using the nonce from the PCP request. The server does not record the nonce.

If an Option with value less than 128 exists (i.e., mandatory to process) but that Option does not make sense (e.g., the PREFER_FAILURE Option is included in a request with lifetime=0), the request is invalid and generates a MALFORMED_OPTION error.

If the PCP-controlled device is stateless (that is, it does not establish any per-flow state, and simply rewrites the address and/or port in a purely algorithmic fashion), the PCP server simply returns an answer indicating the external IP address and port yielded by this stateless algorithmic translation. This allows the PCP client to learn its external IP address and port as seen by remote peers. Examples of stateless translators include stateless NAT64, 1:1 NAT44, and NPTv6 [RFC6296], all of which modify addresses but not port numbers.

It is possible that a mapping might already exist for a requested Internal Address, Protocol, and Port. If so, the PCP server takes the following actions:

1. If the MAP request contains the PREFER_FAILURE Option, but the Suggested External Address and Port do not match the External Address and Port of the existing mapping, the PCP server MUST return CANNOT_PROVIDE_EXTERNAL.

2. If the existing mapping is static (created outside of PCP), the PCP server MUST return the External Address and Port of the existing mapping in its response and SHOULD indicate a Lifetime of 2^32-1 seconds, regardless of the Suggested External Address and Port in the request.

3. If the existing mapping is explicit dynamic inbound (created by a previous MAP request), the PCP server MUST return the existing External Address and Port in its response, regardless of the Suggested External Address and Port in the request. Additionally, the PCP server MUST update the lifetime of the existing mapping, in accordance with section 10.5.

4. If the existing mapping is dynamic outbound (created by outgoing traffic or a previous PEER request), the PCP server SHOULD create a new explicit inbound mapping, replicating the ports and addresses from the outbound mapping (but the outbound mapping continues to exist, and remains in effect if the explicit inbound mapping is later deleted).

If no mapping exists for the Internal Address, Protocol, and Port, and the PCP server is able to create a mapping using the Suggested External Address and Port, it SHOULD do so. This is beneficial for re-establishing state lost in the PCP server (e.g., due to a reboot). There are, however, cases where the PCP server is not able to create a new mapping using the Suggested External Address and Port:

- The Suggested External Address, Protocol, and Port is already assigned to another existing explicit or implicit mapping (i.e., is already forwarding traffic to some other internal address and port).

- The Suggested External Address, Protocol, and Port is already used by the NAT gateway for one of its own services. For example, TCP port 80 for the NAT gateway’s own configuration web pages, or UDP ports 5350 and 5351, used by PCP itself. A PCP server MUST NOT create client mappings for External UDP ports 5350 or 5351.

- The Suggested External Address, Protocol, and Port is otherwise prohibited by the PCP server’s policy.

- The Suggested External IP Address, Protocol, or Suggested Port are invalid or invalid combinations (e.g., External Address 127.0.0.1,

::1, a multicast address, or the Suggested Port is not valid for the Protocol).

- o The Suggested External Address does not belong to the NAT gateway.

- o The Suggested External Address is not configured to be used as an external address of the firewall or NAT gateway.

If the PCP server cannot assign the Suggested External Address, Protocol, and Port, then:

- o If the request contained the PREFER_FAILURE Option, then the PCP server MUST return CANNOT_PROVIDE_EXTERNAL.

- o If the request did not contain the PREFER_FAILURE Option, and the PCP server can assign some other External Address and Port for that protocol, then the PCP server MUST do so and return the newly assigned External Address and Port in the response. In no case is the client penalized for a 'poor' choice of Suggested External Address and Port. The Suggested External Address and Port may be used by the server to guide its choice of what External Address and Port to assign, but in no case do they cause the server to fail to allocate an External Address and Port where otherwise it would have succeeded. The presence of a non-zero Suggested External Address or Port is merely a hint; it never does any harm.

By default, a PCP-controlled device MUST NOT create mappings for a protocol not indicated in the request. For example, if the request was for a TCP mapping, a UDP mapping MUST NOT be created.

Mappings typically consume state on the PCP-controlled device, and it is RECOMMENDED that a per-host and/or per-subscriber limit be enforced by the PCP server to prevent exhausting the mapping state. If this limit is exceeded, the result code USER_EX_QUOTA is returned.

If all of the preceding operations were successful (did not generate an error response), then the requested mapping is created or refreshed as described in the request and a SUCCESS response is built.

11.4. Processing a MAP Response

This section describes the operation of the PCP client when it receives a PCP response for the MAP Opcode.

After performing common PCP response processing, the response is further matched with a previously-sent MAP request by comparing the Internal IP Address (the destination IP address of the PCP response,

or other IP address specified via the THIRD_PARTY option), the Protocol, the Internal Port, and the Mapping Nonce. Other fields are not compared, because the PCP server sets those fields. The PCP server will send a Mapping Update (Section 14.2) if the mapping changes (e.g., due to IP renumbering).

If the result code is NO_RESOURCES and the request was for the creation or renewal of a mapping, then the PCP client SHOULD NOT send further requests for any new mappings to that PCP server for the (limited) value of the Lifetime. If the result code is NO_RESOURCES and the request was for the deletion of a mapping, then the PCP client SHOULD NOT send further requests of *any kind* to that PCP server for the (limited) value of the Lifetime.

On a success response, the PCP client can use the External IP Address and Port as needed. Typically the PCP client will communicate the External IP Address and Port to another host on the Internet using an application-specific rendezvous mechanism such as DNS SRV records.

As long as renewal is desired, the PCP client MUST also set a timer or otherwise schedule an event to renew the mapping before its lifetime expires. Renewing a mapping is performed by sending another MAP request, exactly as described in Section 11.2, except that the Suggested External Address and Port SHOULD be set to the values received in the response. From the PCP server’s point of view a MAP request to renew a mapping is identical to a MAP request to create a new mapping, and is handled identically. Indeed, in the event of PCP server state loss, a renewal request from a PCP client will appear to the server to be a request to create a new mapping, with a particular Suggested External Address and Port, which happens to be what the PCP server previously assigned. See also Section 16.3.1.

On an error response, the client SHOULD NOT repeat the same request to the same PCP server within the lifetime returned in the response.

11.5. Address Change Events

A customer premises router might obtain a new External IP address, for a variety of reasons including a reboot, power outage, DHCP lease expiry, or other action by the ISP. If this occurs, traffic forwarded to the host’s previous address might be delivered to another host which now has that address. This affects all mapping types, whether implicit or explicit. This same problem already occurs today when a host’s IP address is re-assigned, without PCP and without an ISP-operated CGN. The solution is the same as today: the problems associated with host renumbering are caused by host renumbering, and are eliminated if host renumbering is avoided. PCP defined in this document does not provide machinery to reduce the

host renumbering problem.

When an Internal Host changes its Internal IP address (e.g., by having a different address assigned by the DHCP server) the NAT (or firewall) will continue to send traffic to the old IP address. Typically, the Internal Host will no longer receive traffic sent to that old IP address. Assuming the Internal Host wants to continue receiving traffic, it needs to install new mappings for its new IP address. The suggested external port field will not be fulfilled by the PCP server, in all likelihood, because it is still being forwarded to the old IP address. Thus, a mapping is likely to be assigned a new External Port number and/or External IP Address. Note that such host renumbering is not expected to happen routinely on a regular basis for most hosts, since most hosts renew their DHCP leases before they expire (or re-request the same address after reboot) and most DHCP servers honor such requests and grant the host the same address it was previously using before the reboot.

A host might gain or lose interfaces while existing mappings are active (e.g., Ethernet cable plugged in or removed, joining/leaving a Wi-Fi network). Because of this, if the PCP client is sending a PCP request to maintain state in the PCP server, it SHOULD ensure those PCP requests continue to use the same interface (e.g., when refreshing mappings). If the PCP client is sending a PCP request to create new state in the PCP server, it MAY use a different source interface or different source address.

11.6. Learning the External IP Address Alone

NAT-PMP [I-D.cheshire-nat-pmp] includes a mechanism to allow clients to learn the External IP Address alone, without also requesting a port mapping. NAT-PMP was designed for residential NAT gateways, where such an operation makes sense because the residential NAT gateway has only one External IP Address. PCP has broader scope, and also supports Carrier-Grade NATs (CGN) which may have a pool of External IP Addresses, not just one. A client may not be assigned any particular External IP Address from that pool until it has at least one implicit, explicit or static port mapping, and even then only for as long as that mapping remains valid. Client software that just wishes to display the user’s External IP Address for cosmetic purposes can achieve that by requesting a short-lived mapping (e.g., to the Discard service (TCP/9 or UDP/9) or some other port) and then displaying the resulting External IP Address. However, once that mapping expires a subsequent implicit or explicit dynamic mapping might be mapped to a different external IP address.

12. PEER Opcode

This section defines an Opcode for controlling dynamic mappings.

PEER: Create a new dynamic outbound mapping to a remote peer's IP address and port, or extend the lifetime of an existing outbound mapping.

The use of this Opcodes is described in this section.

PCP Servers SHOULD provide a configuration option to allow administrators to disable PEER support if they wish.

Because a mapping created or managed by PEER behaves almost exactly like an implicit dynamic mapping created as a side-effect of a packet (e.g., TCP SYN) sent by the host, mappings created or managed using PCP PEER requests may be Endpoint Independent Mappings (EIM) or Endpoint Dependent Mappings (EDM), with Endpoint Independent Filtering (EIF) or Endpoint Dependent Filtering (EDF), consistent with the existing behavior of the NAT gateway or firewall in question for implicit outbound mappings it creates automatically as a result of observing outgoing traffic from Internal Hosts.

12.1. PEER Operation Packet Formats

The PEER Opcode allows a PCP client to create a new explicit dynamic outbound mapping (which functions similarly to an outbound mapping created implicitly when a host sends an outbound TCP SYN) or to extend the lifetime of an existing outbound mapping.

The following diagram shows the Opcode layout for the PEER Opcode. This packet format is aligned with the response packet format:

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                 Mapping Nonce (96 bits)                       |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Protocol    |          Reserved (24 bits)                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        Internal Port          |    Suggested External Port    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|           Suggested External IP Address (128 bits)            |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       Remote Peer Port        |     Reserved (16 bits)        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|               Remote Peer IP Address (128 bits)               |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 11: PEER Opcode Request

These fields are described below:

Requested Lifetime (in common header): Requested lifetime of this mapping, in seconds. Note that it is not possible to reduce the lifetime of a mapping (or delete it, with requested lifetime=0) using PEER.

Mapping Nonce: Random value chosen by the PCP client. See Section 12.2. Zero is a legal value (but unlikely, occurring in roughly one in 2^96 requests).

Protocol: Upper-layer protocol associated with this Opcode. Values are taken from the IANA protocol registry [proto_numbers]. For example, this field contains 6 (TCP) if the Opcode is describing a TCP mapping. Protocol MUST NOT be zero.

Reserved: 24 reserved bits, MUST be set to 0 on transmission and MUST be ignored on reception.

Internal Port: Internal port for the mapping. Internal Port MUST NOT be zero.

Suggested External Port: Suggested external port for the mapping. If the PCP client does not know the external port, or does not have a preference, it MUST use 0.

Suggested External IP Address: Suggested External IP Address for the mapping. If the PCP client does not know the external address, or does not have a preference, it MUST use the address-family-specific all-zeroes address (see Section 5).

Remote Peer Port: Remote peer’s port for the mapping. Remote Peer Port MUST NOT be zero.

Reserved: 16 reserved bits, MUST be set to 0 on transmission and MUST be ignored on reception.

Remote Peer IP Address: Remote peer’s IP address. This is from the perspective of the PCP client, so that the PCP client does not need to concern itself with NAT64 or NAT46 (which both cause the client’s idea of the remote peer’s IP address to differ from the remote peer’s actual IP address). This field allows the PCP client and PCP server to disambiguate multiple connections from the same port on the Internal Host to different servers. An IPv6 address is represented directly, and an IPv4 address is represented using the IPv4-mapped address syntax (Section 5).

When attempting to re-create a lost mapping, the Suggested External IP Address and Port are set to the External IP Address and Port fields received in a previous PEER response from the PCP server. On an initial PEER request, the External IP Address and Port are set to zero.

Note that semantics similar to the PREFER_FAILURE option are automatically implied by PEER requests. If the Suggested External IP Address or Suggested External Port fields are non-zero, and the PCP server is unable to honor the Suggested External IP Address, Protocol, or Port, then the PCP server MUST return a CANNOT_PROVIDE_EXTERNAL error response. The PREFER_FAILURE Option is neither required nor allowed in PEER requests, and if PCP server receives a PEER request containing the PREFER_FAILURE Option it MUST return a MALFORMED_REQUEST error response.

The following diagram shows the Opcode response for the PEER Opcode:

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                 Mapping Nonce (96 bits)                       |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Protocol    |          Reserved (24 bits)                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        Internal Port          |    Assigned External Port     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|            Assigned External IP Address (128 bits)            |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       Remote Peer Port        |     Reserved (16 bits)        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|               Remote Peer IP Address (128 bits)               |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 12: PEER Opcode Response

Lifetime (in common header): On a success response, this indicates the lifetime for this mapping, in seconds. On an error response, this indicates how long clients should assume they'll get the same error response from the PCP server if they repeat the same request.

Mapping Nonce: Copied from the request.

Protocol: Copied from the request.

Reserved: 24 reserved bits, MUST be set to 0 on transmission, MUST be ignored on reception.

Internal Port: Copied from request.

Assigned External Port: On a success response, this is the assigned external port for the mapping. On an error response, the Suggested External Port is copied from the request.

Assigned External IP Address: On a success response, this is the assigned external IPv4 or IPv6 address for the mapping. On an error response, the Suggested External IP Address is copied from the request.

Remote Peer port: Copied from request.

Reserved: 16 reserved bits, MUST be set to 0 on transmission, MUST be ignored on reception.

Remote Peer IP Address: Copied from the request.

12.2. Generating a PEER Request

This section describes the operation of a client when generating a message with the PEER Opcode.

The PEER Opcode MAY be sent before or after establishing bi-directional communication with the remote peer.

If sent before, this is considered a PEER-created mapping which creates a new dynamic outbound mapping in the PCP-controlled device. This is useful for restoring a mapping after a NAT has lost its mapping state (e.g., due to a crash).

If sent after, this allows the PCP client to learn the IP address, port, and lifetime of the assigned External Address and Port for the existing implicit dynamic outbound mapping, and potentially to extend this lifetime (for the purpose described in Section 10.3).

The Mapping Nonce value is randomly chosen by the PCP client, following accepted practices for generating unguessable random numbers [RFC4086], and is used as part of the validation of PCP responses (see below) by the PCP client, and validation for mapping refreshes by the PCP server. The client MUST use a different Mapping Nonce for each PCP server it communicates with, and it is RECOMMENDED to choose a new random Mapping Nonce whenever the PCP client is initialized. The client MAY use a different Mapping Nonce for every mapping.

The PEER Opcode contains a Remote Peer Address field, which is always from the perspective of the PCP client. Note that when the PCP-controlled device is performing address family translation (NAT46 or NAT64), the remote peer address from the perspective of the PCP client is different from the remote peer address on the other side of the address family translation device.

12.3. Processing a PEER Request

This section describes the operation of a server when receiving a request with the PEER Opcode. Processing SHOULD be performed in the order of the following paragraphs.

The following fields from a PEER request are copied into the response: Protocol, Internal Port, Remote Peer IP Address, Remote Peer Port, and Mapping Nonce.

When an implicit dynamic mapping is created, some NATs and firewalls validate destination addresses and will not create an implicit dynamic mapping if the destination address is invalid (e.g., 127.0.0.1). If a PCP-controlled device does such validation for implicit dynamic mappings, it SHOULD also do a similar validation of the Remote Peer IP Address, Protocol, and Port for PEER-created explicit dynamic mappings. If the validation determines the Remote Peer IP Address of a PEER request is invalid, then no mapping is created, and a MALFORMED_REQUEST error result is returned.

On receiving the PEER Opcode, the PCP server examines the mapping table for a matching five-tuple { Protocol, Internal Address, Internal Port, Remote Peer Address, Remote Peer Port }.

If no matching mapping is found, and the Suggested External Address and Port are either zero or can be honored for the specified Protocol, a new mapping is created. By having PEER create such a mapping, we avoid a race condition between the PEER request or the initial outgoing packet arriving at the NAT or firewall device first, and allow PEER to be used to recreate an outbound dynamic mapping (see last paragraph of Section 16.3.1). Thereafter, this PEER-created mapping is treated as if it was an implicit dynamic outbound mapping (e.g., as if the PCP client sent a TCP SYN) and a Lifetime appropriate to such a mapping is returned (note: on many NATs and firewalls, such mapping lifetimes are very short until the bi-directional traffic is seen by the NAT or firewall).

If no matching mapping is found, and the Suggested External Address and Port cannot be honored, then no new state is created, and the error CANNOT_PROVIDE_EXTERNAL is returned.

If a matching mapping is found, but no pervious PEER Opcode was successfully processed for this mapping, then the Suggested External Address and Port values in the request are ignored, Lifetime of that mapping is adjusted as described below, and information about the existing mapping is returned. This allows a client to explicitly extend the lifetime of an existing mapping and/or to learn an existing mapping’s External Address, Port and lifetime. The Mapping

Nonce is remembered for this mapping.

If operating in the Simple Threat Model (Section 18.1), and the Internal port, Protocol, and Internal Address match a mapping that already exists, but the Mapping Nonce does not match (that is, a previous PEER request was processed), the request MUST be rejected with a NOT_AUTHORIZED error with the Lifetime of the error indicating duration of that existing mapping. The PCP server only needs to remember one Mapping Nonce value for each mapping.

Processing the lifetime value of the PEER Opcode is described in Section 15. Sending a PEER request with a very short Requested Lifetime can be used to query the lifetime of an existing mapping.

If all of the preceding operations were successful (did not generate an error response), then a SUCCESS response is generated, with the Lifetime field containing the lifetime of the mapping.

If a PEER-created or PEER-managed mapping is not renewed using PEER, then it reverts to the NAT’s usual behavior for implicit mappings, e.g., continued outbound traffic keeps the mapping alive, as per the NAT or firewall device’s existing policy. A PEER-created or PEER-managed mapping may be terminated at any time by action of the TCP client or server (e.g., due to TCP FIN or TCP RST), as per the NAT or firewall device’s existing policy.

12.4. Processing a PEER Response

This section describes the operation of a client when processing a response with the PEER Opcode.

After performing common PCP response processing, the response is further matched with an outstanding PEER request by comparing the Internal IP Address (the destination IP address of the PCP response, or other IP address specified via the THIRD_PARTY option), the Protocol, the Internal Port, the Remote Peer Address, the Remote Peer Port, and the Mapping Nonce. Other fields are not compared, because the PCP server sets those fields to provide information about the mapping created by the Opcode. The PCP server will send a Mapping Update (Section 14.2) if the mapping changes (e.g., due to IP renumbering).

If the result code is NO_RESOURCES and the request was for the creation or renewal of a mapping, then the PCP client SHOULD NOT send further requests for any new mappings to that PCP server for the (limited) value of the Lifetime.

On a successful response, the application can use the assigned

lifetime value to reduce its frequency of application keepalives for that particular NAT mapping. Of course, there may be other reasons, specific to the application, to use more frequent application keepalives. For example, the PCP assigned lifetime could be one hour but the application may want to maintain state on its server (e.g., "busy" / "away") more frequently than once an hour. If the response indicates an unexpected IP address or port (e.g., due to IP renumbering), the PCP client will want to re-establish its connection to its remote server.

If the PCP client wishes to keep this mapping alive beyond the indicated lifetime, it MAY rely on continued inside-to-outside traffic to ensure the mapping will continue to exist, or it MAY issue a new PCP request prior to the expiration. The recommended timings for renewing PEER mappings are the same as for MAP mappings, as described in Section 11.2.1.

> Note: Implementations need to expect the PEER response may contain an External IP Address with a different family than the Remote Peer IP Address, e.g., when NAT64 or NAT46 are being used.

13. Options for MAP and PEER Opcodes

This section describes Options for the MAP and PEER Opcodes. These Options MUST NOT appear with other Opcodes, unless permitted by those other Opcodes.

13.1. THIRD_PARTY Option for MAP and PEER Opcodes

This Option is used when a PCP client wants to control a mapping to an Internal Host other than itself. This is used with both MAP and PEER Opcodes.

Due to security concerns with the THIRD_PARTY option, this Option MUST NOT be implemented or used unless the network on which the PCP messages are to be sent is fully trusted. For example if access control lists are installed on the PCP client, PCP server, and the network between them, so those ACLs allow only communications from a trusted PCP client to the PCP server.

A management device would use this Option to control a PCP server on behalf of users. For example, a management device located in a network operations center, which presents a user interface to end users or to network operations staff, and issues PCP requests with the THIRD_PARTY option to the appropriate PCP server.

The THIRD_PARTY Option is formatted as follows:

```
 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=1 |  Reserved     |   Option Length=16            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                Internal IP Address (128 bits)                 |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 13: THIRD_PARTY Option

The fields are described below:

Internal IP Address: Internal IP address for this mapping.

Option Name: THIRD_PARTY
Number: 1
Purpose: Indicates the MAP or PEER request is for a host other than the host sending the PCP Option.
Valid for Opcodes: MAP, PEER
Length: 16 octets
May appear in: request. May appear in response only if it appeared in the associated request.
Maximum occurrences: 1

A THIRD_PARTY Option MUST NOT contain the same address as the source address of the packet. This is because many PCP servers may not implement the THIRD_PARTY Option at all, and with those servers a client redundantly using the THIRD_PARTY Option to specify its own IP address would cause such mapping requests to fail where they would otherwise have succeeded. A PCP server receiving a THIRD_PARTY Option specifying the same address as the source address of the packet MUST return a MALFORMED_REQUEST result code.

A PCP server MAY be configured to permit or to prohibit the use of the THIRD_PARTY Option. If this Option is permitted, properly authorized clients may perform these operations on behalf of other hosts. If this Option is prohibited, and a PCP server receives a PCP MAP request with a THIRD_PARTY Option, it MUST generate a UNSUPP_OPTION response.

It is RECOMMENDED that customer premises equipment implementing a PCP Server be configured to prohibit third party mappings by default. With this default, if a user wants to create a third party mapping,

the user needs to interact out-of-band with their customer premises router (e.g., using its HTTP administrative interface).

It is RECOMMENDED that service provider NAT and firewall devices implementing a PCP Server be configured to permit the THIRD_PARTY Option, when sent by a properly authorized host. If the packet arrives from an unauthorized host, the PCP server MUST generate an UNSUPP_OPTION error.

Note that the THIRD_PARTY Option is not needed for today's common scenario of an ISP offering a single IP address to a customer who is using NAT to share that address locally, since in this scenario all the customer's hosts appear, from the point of view of the ISP, to be a single host.

When a PCP client is using the THIRD_PARTY Option to make and maintain mappings on behalf of some other device, it may be beneficial if, where possible, the PCP client verifies that the other device is actually present and active on the network. Otherwise the PCP client risks maintaining those mappings forever, long after the device that required them has gone. This would defeat the purpose of PCP mappings having a finite lifetime so that they can be automatically deleted after they are no longer needed.

13.2. PREFER_FAILURE Option for MAP Opcode

This Option is only used with the MAP Opcode.

This Option indicates that if the PCP server is unable to map both the Suggested External Port and Suggested External Address, the PCP server should not create a mapping. This differs from the behavior without this Option, which is to create a mapping.

The PREFER_FAILURE Option is formatted as follows:

```
 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=2 |  Reserved     |   Option Length=0             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 14: PREFER_FAILURE Option

Option Name: PREFER_FAILURE
Number: 2
Purpose: indicates that the PCP server should not create an alternative mapping if the suggested external port and address cannot be mapped.
Valid for Opcodes: MAP
Length: 0
May appear in: request. May appear in response only if it appeared in the associated request.
Maximum occurrences: 1

The result code CANNOT_PROVIDE_EXTERNAL is returned if the Suggested External Address, Protocol, and Port cannot be mapped. This can occur because the External Port is already mapped to another host's outbound dynamic mapping, an inbound dynamic mapping, a static mapping, or the same Internal Address, Protocol, and Port already has an outbound dynamic mapping which is mapped to a different External Port than suggested. This can also occur because the External Address is no longer available (e.g., due to renumbering). The server MAY set the Lifetime in the response to the remaining lifetime of the conflicting mapping + TIME_WAIT [RFC0793], rounded up to the next larger integer number of seconds.

PREFER_FAILURE is never necessary for a PCP client to manage mappings for itself, and its use causes additional work in the PCP client and in the PCP server. This Option exists for interworking with non-PCP mapping protocols that have different semantics than PCP (e.g., UPnP IGDv1 interworking [I-D.ietf-pcp-upnp-igd-interworking], where the semantics of UPnP IGDv1 only allow the UPnP IGDv1 client to dictate mapping a specific port), or separate port allocation systems which allocate ports to a subscriber (e.g., a subscriber-accessed web portal operated by the same ISP that operates the PCP server). A PCP server MAY support this Option, if its designers wish to support such downstream devices or separate port allocation systems. PCP servers that are not intended to interface with such systems are not required to support this Option. PCP clients other than UPnP IGDv1 interworking clients or other than a separate port allocation system SHOULD NOT use this Option because it results in inefficient operation, and they cannot safely assume that all PCP servers will implement it. It is anticipated that this Option will be deprecated in the future as more clients adopt PCP natively and the need for this Option declines.

If a PCP request contains the PREFER_FAILURE option and has zero in the Suggested External Port field, or has the all-zeros IPv4 or all-zeros IPv6 address in the Suggested External Address field, it is invalid. The PCP server MUST reject such a message with the MALFORMED_OPTION error code.

PCP servers MAY choose to rate-limit their handling of PREFER_FAILURE requests, to protect themselves from a rapid flurry of 65535 consecutive PREFER_FAILURE requests from clients probing to discover which external ports are available.

There can exist a race condition between the MAP Opcode using the PREFER_FAILURE option and Mapping Update (Section 14.2). For example, a previous host on the local network could have previously had the same Internal Address, with a mapping for the same Internal Port. At about the same moment that the current host sends a MAP Request using the PREFER_FAILURE option, the PCP server could send a spontaneous mapping update for the old mapping due to an external configuration change, which could appear to be a reply to the new mapping request. Because of this, the PCP client MUST validate that the External IP Address, Protocol, Port and Nonce in a success response matches the associated suggested values from the request. If they don’t match, it is because the Mapping Update was sent before the MAP request was processed.

13.3. FILTER Option for MAP Opcode

This Option is only used with the MAP Opcode.

This Option indicates that filtering incoming packets is desired. The protocol being filtered is indicated by the Protocol field in the MAP Request, and the Remote Peer IP Address and Remote Peer Port of the FILTER Option indicate the permitted remote peer’s source IP address and source port for packets from the Internet; other traffic from other addresses is blocked. The remote peer prefix length indicates the length of the remote peer’s IP address that is significant; this allows a single Option to permit an entire subnet. After processing this MAP request containing the FILTER Option and generating a successful response, the PCP-controlled device will drop packets received on its public-facing interface that don’t match the filter fields. After dropping the packet, if its security policy allows, the PCP-controlled device MAY also generate an ICMP error in response to the dropped packet.

The use of the FILTER Option can be seen as a performance optimization. Since all software using PCP to receive incoming connections also has to deal with the case where it may be directly connected to the Internet and receive unrestricted incoming TCP connections and UDP packets, if it wishes to restrict incoming traffic to a specific source address or group of source addresses such software already needs to check the source address of incoming traffic and reject unwanted traffic. However, the FILTER Option is a particularly useful performance optimization for battery powered wireless devices, because it can enable them to conserve battery

power by not having to wake up just to reject unwanted traffic.

The FILTER Option is formatted as follows:

```
 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=3 |  Reserved     |   Option Length=20            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Reserved   | Prefix Length |      Remote Peer Port         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|               Remote Peer IP address (128 bits)               |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 15: FILTER Option layout

These fields are described below:

Reserved: 8 reserved bits, MUST be sent as 0 and MUST be ignored when received.

Prefix Length: indicates how many bits of the IPv4 or IPv6 address are relevant for this filter. The value 0 indicates "no filter", and will remove all previous filters. See below for detail.

Remote Peer Port: the port number of the remote peer. The value 0 indicates "all ports".

Remote Peer IP address: The IP address of the remote peer.

Option Name: FILTER
Number: 3
Purpose: specifies a filter for incoming packets
Valid for Opcodes: MAP
Length: 20 octets
May appear in: request. May appear in response only if it appeared in the associated request.
Maximum occurrences: as many as fit within maximum PCP message size

The Prefix Length indicates how many bits of the address are used for the filter. For IPv4 addresses (which are encoded using the IPv4-mapped address format (::FFFF:0:0/96)), this means valid prefix lengths are between 96 and 128 bits, inclusive. That is, add 96 to the IPv4 prefix length. For IPv6 addresses, valid prefix lengths are

between 0 and 128 bits, inclusive. Values outside those ranges cause the PCP server to return the MALFORMED_OPTION result code.

If multiple occurrences of the FILTER Option exist in the same MAP request, they are processed in the order received (as per normal PCP Option processing) and they MAY overlap the filtering requested. If an existing mapping exists (with or without a filter) and the server receives a MAP request with FILTER, the filters indicated in the new request are added to any existing filters. If a MAP request has a lifetime of 0 and contains the FILTER Option, the error MALFORMED_OPTION is returned.

If any occurrences of the FILTER Option in a request packet are not successfully processed then an error is returned (e.g., MALFORMED_OPTION if one of the Options was malformed) and as with other PCP errors, returning an error causes no state to be changed in the PCP server or in the PCP-controlled device.

To remove all existing filters, the Prefix Length 0 is used. There is no mechanism to remove a specific filter.

To change an existing filter, the PCP client sends a MAP request containing two FILTER Options, the first Option containing a Prefix Length of 0 (to delete all existing filters) and the second containing the new remote peer’s IP address, protocol, and port. Other FILTER Options in that PCP request, if any, add more allowed Remote Peers.

The PCP server or the PCP-controlled device is expected to have a limit on the number of remote peers it can support. This limit might be as small as one. If a MAP request would exceed this limit, the entire MAP request is rejected with the result code EXCESSIVE_REMOTE_PEERS, and the state on the PCP server is unchanged.

All PCP servers MUST support at least one filter per MAP mapping.

14. Rapid Recovery

PCP includes a rapid recovery feature, which allows PCP clients to repair failed mappings within seconds, rather than the minutes or hours it might take if they relied solely on waiting for the next routine renewal of the mapping. Mapping failures may occur when a NAT gateway is rebooted and loses its mapping state, or when a NAT gateway has its external IP address changed so that its current mapping state becomes invalid.

The PCP rapid recovery feature enables users to, for example, connect

to remote machines using ssh, and then reboot their NAT or firewall device (or even replace it with completely new hardware) without losing their established ssh connections.

Use of PCP rapid recovery is a performance optimization to PCP's routine self-healing. Without rapid recovery, PCP clients will still recreate their correct state when they next renew their mappings, but this routine self-healing process may take hours rather than seconds, and will probably not happen fast enough to prevent active TCP connections from timing out.

There are two mechanisms to perform rapid recovery, described below. A PCP server that can lose state (e.g., due to reboot) or might have a mapping change (e.g., due to IP renumbering) MUST implement either the Announce Opcode or the Mapping Update mechanism and SHOULD implement both mechanisms. Failing to implement and deploy a rapid recovery mechanism will encourage application developers to feel the need to refresh their PCP state more frequently than necessary, causing more network traffic.

14.1. ANNOUNCE Opcode

This rapid recovery mechanism uses the ANNOUNCE Opcode. When the PCP server loses its state (e.g., it lost its state when rebooted), it sends the ANNOUNCE response to the link-scoped multicast address (specific address explained below) if a multicast network exists on its local interface or, if configured with the IP address(es) and port(s) of PCP client(s), sends unicast ANNOUNCE responses to those address(es) and port(s). This means ANNOUNCE may not be available on all networks (such as networks without a multicast link between the PCP server and its PCP clients). Additionally, an ANNOUNCE request can be sent (unicast) by a PCP client which elicits a unicast ANNOUNCE response like any other Opcode.

14.1.1. ANNOUNCE Operation

The PCP ANNOUNCE Opcode requests and respones have no Opcode-specific payload (that is, the length of the Opcode-specific data is zero). The Requested Lifetime field of requests and Lifetime field of responses are both set to 0 on transmission and ignored on reception.

If a PCP server receives an ANNOUNCE request, it first parses it and generates a SUCCESS if parsing and processing of ANNOUNCE is successful. An error is generated if the Client's IP Address field does not match the packet source address, or the request packet is otherwise malformed, such as packet length less than 24 octets. Note that, in the future, Options MAY be sent with the PCP ANNOUNCE Opcode; PCP clients and servers need to be prepared to receive

Options with the ANNOUNCE Opcode.

Discussion: Client-to-server request messages are sent to listening UDP port 5351 on the server; server-to-client multicast notifications are sent to listening UDP port 5350 on the client. The reason the same UDP port is not used for both purposes is that a single device may have multiple roles. For example, a multi-function home gateway that provides NAT service (PCP server) may also provide printer sharing (which wants a PCP client), or a home computer (PCP client) may also provide "Internet Sharing" (NAT) functionality (which needs to offer PCP service). Such devices need to act as both a PCP Server and a PCP Client at the same time, and the software that implements the PCP Server on the device may not be the same software component that implements the PCP Client. The software that implements the PCP Server needs to listen for unicast client requests, whereas the software that implements the PCP Client needs to listen for multicast restart announcements. In many networking APIs it is difficult or impossible to have two independent clients listening for both unicasts and multicasts on the same port at the same time. For this reason, two ports are used.

14.1.2. Generating and Processing a Solicited ANNOUNCE Message

The PCP ANNOUNCE Opcode MAY be sent (unicast) by a PCP client. The Requested Lifetime value MUST be set to zero.

When the PCP server receives the ANNOUNCE Opcode and successfully parses and processes it, it generates SUCCESS response with an Assigned Lifetime of zero.

This functionality allows a PCP client to determine a server’s Epoch, or to determine if a PCP server is running, without changing the server’s state.

14.1.3. Generating and Processing an Unsolicited ANNOUNCE Message

When sending unsolicited responses, the ANNOUNCE Opcode MUST have Result Code equal to zero (SUCCESS), and the packet MUST be sent from the unicast IP address and UDP port number on which PCP requests are received (so PCP response processing accepts the message, see Section 8.3). This message is most typically multicast, but can also be unicast. Multicast PCP restart announcements are sent to 224.0.0.1:5350 and/or [ff02::1]:5350, as described below. Sending PCP restart announcements via unicast requires that the PCP server know the IP address(es) and port(s) of its listening clients, which means that sending PCP restart announcements via unicast is only applicable to PCP servers that retain knowledge of the IP address(es)

and port(s) of their clients even after they otherwise lose the rest of their state.

When a PCP server device that implements this functionality reboots, restarts its NAT engine, or otherwise enters a state where it may have lost some or all of its previous mapping state (or enters a state where it doesn’t even know whether it may have had prior state that it lost) it MUST inform PCP clients of this fact by unicasting or multicasting a gratuitous PCP ANNOUNCE Opcode response packet, as shown below, via paths over which it accepts PCP requests. If sending a multicast ANNOUNCE message, a PCP server device which accepts PCP requests over IPv4 sends the Restart Announcement to the IPv4 multicast address 224.0.0.1:5350 (224.0.0.1 is the All Hosts multicast group address), and a PCP server device which accepts PCP requests over IPv6 sends the Restart Announcement to the IPv6 multicast address [ff02::1]:5350 (ff02::1 is for all nodes on the local segment). A PCP server device which accepts PCP requests over both IPv4 and IPv6 sends a pair of Restart Announcements, one to each multicast address. If sending a unicast ANNOUNCE messages, it sends ANNOUNCE response message to the IP address(es) and port(s) of its PCP clients. To accommodate packet loss, the PCP server device MAY transmit such packets (or packet pairs) up to ten times (with an appropriate Epoch time value in each to reflect the passage of time between transmissions) provided that the interval between the first two notifications is at least 250ms, and the interval between subsequent notification at least doubles.

A PCP client that sends PCP requests to a PCP Server via a multicast-capable path, and implements the Restart Announcement feature, and wishes to receive these announcements, MUST listen to receive these PCP Restart Announcements (gratuitous PCP ANNOUNCE Opcode response packets) on the appropriate multicast-capable interfaces on which it sends PCP requests, and MAY also listen for unicast announcements from the server too, (using the UDP port it already uses to issue unicast PCP requests to, and receive unicast PCP responses from, that server). A PCP client device which sends PCP requests using IPv4 listens for packets sent to the IPv4 multicast address 224.0.0.1: 5350. A PCP client device which sends PCP requests using IPv6 listens for packets sent to the IPv6 multicast address [ff02::1]: 5350. A PCP client device which sends PCP requests using both IPv4 and IPv6 listens for both types of Restart Announcement. The SO_REUSEPORT socket option or equivalent should be used for the multicast UDP port, if required by the host OS to permit multiple independent listeners on the same multicast UDP port.

Upon receiving a unicasted or multicasted PCP ANNOUNCE Opcode response packet, a PCP client MUST (as it does with all received PCP response packets) inspect the Announcement’s source IP address, and

if the Epoch time value is outside the expected range for that server, it MUST wait a random amount of time between 0 and 5 seconds (to prevent synchronization of all PCP clients), then for all PCP mappings it made at that server address the client issues new PCP requests to recreate any lost mapping state. The use of the Suggested External IP Address and Suggested External Port fields in the client's renewal requests allows the client to remind the restarted PCP server device of what mappings the client had previously been given, so that in many cases the prior state can be recreated. For PCP server devices that reboot relatively quickly it is usually possible to reconstruct lost mapping state fast enough that existing TCP connections and UDP communications do not time out, and continue without failure. As for all PCP response messages, if the Epoch time value is within the expected range for that server, the PCP client does not recreate its mappings. As for all PCP response messages, after receiving and validating the ANNOUNCE message, the client updates its own Epoch time for that server, as described in Section 8.5.

14.2. PCP Mapping Update

This rapid recovery mechanism is used when the PCP server remembers its state and determines its existing mappings are invalid (e.g., IP renumbering changes the External IP Address of a PCP-controlled NAT).

It is anticipated that servers which are routinely reconfigured by an administrator or have their WAN address changed frequently will implement this feature (e.g., residential CPE routers). It is anticipated that servers which are not routinely reconfigured will not implement this feature (e.g., service provider-operated CGN).

If a PCP server device has not forgotten its mapping state, but for some other reason has determined that some or all of its mappings have become unusable (e.g., when a home gateway is assigned a different external IPv4 address by the upstream DHCP server) then the PCP server device automatically repairs its mappings and notifies its clients by following the procedure described below.

For PCP-managed mappings, for each one the PCP server device should update the External IP Address and External Port to appropriate available values, and then send unicast PCP MAP or PEER responses (as appropriate for the mapping) to inform the PCP client of the new External IP Address and External Port. Such unsolicited responses are identical to the MAP or PEER responses normally returned in response to client MAP or PEER requests, containing newly updated External IP Address and External Port values, and are sent to the same client IP address and port that the PCP server used to send the prior response for that mapping. If the earlier associated request

contained the THIRD_PARTY Option, the THIRD_PARTY Option MUST also appear in the Mapping Update as it is necessary for the PCP client to disambiguate the response. If the earlier associated request contained the PREFER_FAILURE option, and the same external IP address, protocol, and port cannot be provided, the error CANNOT_PROVIDE_EXTERNAL SHOULD be sent. If the earlier associated request contained the FILTER option, the filters are moved to the new mapping and the FILTER Option is sent in the Mapping Update response. Non-mandatory Options SHOULD NOT be sent in the Mapping Update response.

Discussion: It could have been possible to design this so that the PCP server (1) sent an ANNOUNCE Opcode to the PCP client, the PCP client reacted by (2) sending a new MAP request and (3) receiving a MAP response. Instead, that design is short-cutted by the server simply sending the message it would have sent in (3).

To accommodate packet loss, the PCP server device SHOULD transmit such packets 3 times, with an appropriate Epoch time value in each to reflect the passage of time between transmissions. The interval between the first two notifications MUST be at least 250ms, and the third packet after a 500ms interval. Once the PCP server has received a refreshed state for that mapping, the PCP server SHOULD cease those retransmissions for that mapping, as it serves no further purpose to continue sending messages regarding that mapping.

Upon receipt of such an updated MAP or PEER response, a PCP client uses the information in the response to adjust rendezvous servers or re-connect to servers, respectively. For MAP, this would means updating the DNS entries or other address and port information recorded with some kind of application-specific rendezvous server. For PEER responses giving a CANNOT_PROVIDE_EXTERNAL error, this would typically mean establishing new connections to servers. Any time the external address or port changes, existing TCP and UDP connections will be lost; PCP can’t avoid that, but does provide immediate notification of the event to lessen the impact.

15. Mapping Lifetime and Deletion

The PCP client requests a certain lifetime, and the PCP server responds with the assigned lifetime. The PCP server MAY grant a lifetime smaller or larger than the requested lifetime. The PCP server SHOULD be configurable for permitted minimum and maximum lifetime, and the minimum value SHOULD be 120 seconds. The maximum value SHOULD be the remaining lifetime of the IP address assigned to the PCP client if that information is available (e.g., from the DHCP server), or half the lifetime of IP address assignments on that

network if the remaining lifetime is not available, or 24 hours. Excessively long lifetimes can cause consumption of ports even if the Internal Host is no longer interested in receiving the traffic or is no longer connected to the network. These recommendations are not strict, and deployments should evaluate the trade offs to determine their own minimum and maximum lifetime values.

Once a PCP server has responded positively to a MAP request for a certain lifetime, the port mapping is active for the duration of the lifetime unless the lifetime is reduced by the PCP client (to a shorter lifetime or to zero) or until the PCP server loses its state (e.g., crashes). Mappings created by PCP MAP requests are not special or different from mappings created in other ways. In particular, it is implementation-dependent if outgoing traffic extends the lifetime of such mappings beyond the PCP-assigned lifetime. PCP clients MUST NOT depend on this behavior to keep mappings active, and MUST explicitly renew their mappings as required by the Lifetime field in PCP response messages.

Upon receipt of a PCP response with an absurdly long Assigned Lifetime the PCP client SHOULD behave as if it received a more sane value (e.g., 24 hours), and renew the mapping accordingly, to ensure that if the static mapping is removed the client will continue to maintain the mapping it desires.

An application that forgets its PCP-assigned mappings (e.g., the application or OS crashes) will request new PCP mappings. This may consume port mappings, if the application binds to a different Internal Port every time it runs. The application will also likely initiate new implicit dynamic outbound mappings without using PCP, which will also consume port mappings. If there is a port mapping quota for the Internal Host, frequent restarts such as this may exhaust the quota and using the same Mapping Nonce can help alleviate such exhaustion.

To help clean PCP state, it is RECOMMENDED that devices which combine IP address assignment (e.g., DHCP server) with the PCP server function (e.g., such as a residential CPE) flush PCP state when an IP address is allocated to a new host, because the new host will be unable perform the functions described in the previous paragaph because the new host does not know the previous host’s Mapping Nonce value. It is good hygiene to also flush TCP and UDP flow state of NAT or fireall functions, although out of scope of this document.

To reduce unwanted traffic and data corruption for both TCP and UDP, the Assigned External Port created by the MAP Opcode or PEER Opcode SHOULD NOT be re-used for the same interval enforced by NAT for implicitly creating mappings, which is typically the maximum segment

lifetime interval of 120 seconds [RFC0793]. To reduce port stealing attacks, the Assigned External Port SHOULD NOT be re-used by the same Client IP Address (or Internal IP Address if using the THIRD_PARTY Option) for the duration the PCP-controlled device keeps a mapping for active bi-directional traffic (e.g., 2 minutes for UDP [RFC4787], 2 hours 4 minutes for TCP [RFC5382]). However, within the above times, the PCP server SHOULD allow a request using the same Client IP Address (and same Internal IP Address if using the THIRD_PARTY Option), Internal Port, and Mapping Nonce to re-acquire the same External Port.

The assigned lifetime is calculated by subtracting (a) zero or the number of seconds since the internal host sent a packet for this mapping from (b) the lifetime the PCP-controlled device uses for transitory connection idle-timeout (e.g., a NAT device might use 2 minutes for UDP [RFC4787] or 4 minutes for TCP [RFC5382]). If the result is a negative number, the assigned lifetime is 0.

15.1. Lifetime Processing for the MAP Opcode

If the the requested lifetime is zero then:

- o If both the protocol and internal port are non-zero, it indicates a request to delete the indicated mapping immediately.

- o If the protocol is non-zero and the internal port is zero, it indicates a request to delete a previous 'wildcard' (all-ports) mapping for that protocol.

- o If both the protocol and internal port are zero, it indicates a request to delete all mappings for this Internal Address for all transport protocols. Such a request is rejected with a NOT_AUTHORIZED error. To delete all mappings the client has to send separate MAP requests with appropriate Mapping Nonce values.

- o If the protocol is zero and the internal port is non-zero, then the request is invalid and the PCP Server MUST return a MALFORMED_REQUEST error to the client.

In requests where the requested Lifetime is 0, the Suggested External Address and Suggested External Port fields MUST be set to zero on transmission and MUST be ignored on reception, and these fields MUST be copied into the Assigned External IP Address and Assigned External Port of the response.

PCP MAP requests can only delete or shorten lifetimes of MAP-created mappings. If the PCP client attempts to delete a static mapping (i.e., a mapping created outside of PCP itself), or an outbound

(implicit or PEER-created) mapping, the PCP server MUST return NOT_AUTHORIZED. If the PCP client attempts to delete a mapping that does not exist, the SUCCESS result code is returned (this is necessary for PCP to return the same response for the same request). If the deletion request was properly formatted and successfully processed, a SUCCESS response is generated with the assigned lifetime of the mapping and the server copies the protocol and internal port number from the request into the response. An inbound mapping (i.e., static mapping or MAP- created dynamic mapping) MUST NOT have its lifetime reduced by transport protocol messages (e.g., TCP RST, TCP FIN). Note the THIRD_PARTY Option, if authorized, can also delete PCP-created mappings (see Section 13.1).

16. Implementation Considerations

Section 16 provides non-normative guidance that may be useful to implementers.

16.1. Implementing MAP with EDM port-mapping NAT

For implicit dynamic outbound mappings, some existing NAT devices have endpoint-independent mapping (EIM) behavior while other NAT devices have endpoint-dependent mapping (EDM) behavior. NATs which have EIM behavior do not suffer from the problem described in this section. The IETF strongly encourages EIM behavior [RFC4787][RFC5382].

In EDM NAT devices, the same external port may be used by an outbound dynamic mapping and an inbound dynamic mapping (from the same Internal Host or from a different Internal Host). This complicates the interaction with the MAP Opcode. With such NAT devices, there are two ways envisioned to implement the MAP Opcode:

1. Have outbound mappings use a different set of External ports than inbound mappings (e.g., those created with MAP), thus reducing the interaction problem between them; or

2. On arrival of a packet (inbound from the Internet or outbound from an Internal Host), first attempt to use a dynamic outbound mapping to process that packet. If none match, attempt to use an inbound mapping to process that packet. This effectively ’prioritizes’ outbound mappings above inbound mappings.

16.2. Lifetime of Explicit and Implicit Dynamic Mappings

No matter if a NAT is EIM or EDM, it is possible that one (or more) outbound mappings, using the same internal port on the Internal Host, might be created before or after a MAP request. When this occurs, it is important that the NAT honor the Lifetime returned in the MAP response. Specifically, if a mapping was created with the MAP Opcode, the implementation needs to ensure that termination of an outbound mapping (e.g., via a TCP FIN handshake) does not prematurely destroy the MAP-created inbound mapping.

16.3. PCP Failure Recovery

If an event occurs that causes the PCP server to lose dynamic mapping state (such as a crash or power outage), the mappings created by PCP are lost. Occasional loss of state may be unavoidable in a residential NAT device which does not write transient information to non-volatile memory. Loss of state is expected to be rare in a service provider environment (due to redundant power, disk drives for storage, etc.). Of course, due to outright failure of service provider equipment (e.g., software malfunction), state may still be lost.

The Epoch Time allows a client to deduce when a PCP server may have lost its state. When the Epoch Time value is observed to be outside the expected range, the PCP client can attempt to recreate the mappings following the procedures described in this section.

Further analysis of PCP failure scenarios is in [I-D.boucadair-pcp-failure].

16.3.1. Recreating Mappings

A mapping renewal packet is formatted identically to an original mapping request; from the point of view of the client it is a renewal of an existing mapping, but from the point of view of a newly rebooted PCP server it appears as a new mapping request. In the normal process of routinely renewing its mappings before they expire, a PCP client will automatically recreate all its lost mappings.

When the PCP server loses state and begins processing new PCP messages, its Epoch time is reset and begins counting again. As the result of receiving a packet where the Epoch time field is outside the expected range (Section 8.5), indicating that a reboot or similar loss of state has occurred, the client can renew its port mappings sooner, without waiting for the normal routine renewal time.

16.3.2. Maintaining Mappings

A PCP client refreshes a mapping by sending a new PCP request containing information from the earlier PCP response. The PCP server will respond indicating the new lifetime. It is possible, due to reconfiguration or failure of the PCP server, that the External IP Address and/or External Port, or the PCP server itself, has changed (due to a new route to a different PCP server). Such events are rare, but not an error. The PCP server will simply return a new External Address and/or External Port to the client, and the client should record this new External Address and Port with its rendezvous service. To detect such events more quickly, a server that requires extremely high availability may find it beneficial to use shorter lifetimes in its PCP mappings requests, so that it communicates with the PCP server more often. This is an engineering trade-off based on (i) the acceptable downtime for the service in question, (ii) the expected likelihood of NAT or firewall state loss, and (iii) the amount of PCP maintenance traffic that is acceptable.

If the PCP client has several mappings, the Epoch Time value only needs to be retrieved for one of them to determine whether or not it appears the PCP server may have suffered a catastrophic loss of state. If the client wishes to check the PCP server’s Epoch Time, it sends a PCP request for any one of the client’s mappings. This will return the current Epoch Time value. In that request the PCP client could extend the mapping lifetime (by asking for more time) or maintain the current lifetime (by asking for the same number of seconds that it knows are remaining of the lifetime).

If a PCP client changes its Internal IP Address (e.g., because the Internal Host has moved to a new network), and the PCP client wishes to still receive incoming traffic, it needs create new mappings on that new network. New mappings will typically also require an update to the application-specific rendezvous server if the External Address or Port are different from the previous values (see Section 10.1 and Section 11.5).

16.3.3. SCTP

Although SCTP has port numbers like TCP and UDP, SCTP works differently when behind an address-sharing NAT, in that SCTP port numbers are not changed [I-D.ietf-behave-sctpnat]. Outbound dynamic SCTP mappings use the verification tag of the association instead of the local and remote peer port numbers. As with TCP, explicit outbound mappings can be made to reduce keepalive intervals, and explicit inbound mappings can be made by passive listeners expecting to receive new associations at the external port.

Because an SCTP-aware NAT does not (currently) rewrite SCTP port numbers, it will not be able to assign an External Port that is different from the client's Internal Port. A PCP client making a MAP request for SCTP should be aware of this restriction. The PCP client SHOULD make its SCTP MAP request just as it would for a TCP MAP request: in its initial PCP MAP request it SHOULD specify zero for the External Address and Port, and then in subsequent renewals it SHOULD echo the assigned External Address and Port. However, since a current SCTP-aware NAT can only assign an External Port that is the same as the Internal Port, it may not be able to do that if the External Port is already assigned to a different PCP client. This is likely if there is more than one instance of a given SCTP service on the local network, since both instances are likely to listen on the same well-known SCTP port for that service on their respective hosts, but they can't both have the same External Port on the NAT gateway's External Address. A particular External Port may not be assignable for other reasons, such as when it is already in use by the NAT device itself, or otherwise prohibited by policy, as described in Section 11.3. In the event that the External Port matching the Internal Port cannot be assigned (and the SCTP-aware NAT does not perform SCTP port rewriting) then the SCTP-aware NAT MUST return a CANNOT_PROVIDE_EXTERNAL error to the requesting PCP client. Note that this restriction places extra burden on the SCTP server whose MAP request failed, because it then has to tear down its exiting listening socket and try again with a different Internal Port, repeatedly until it is successful in finding an External Port it can use.

The SCTP complications described above occur because of address sharing. The SCTP complications are avoided when address sharing is avoided (e.g., 1:1 NAT, firewall).

16.4. Source Address Replicated in PCP Header

All PCP requests include the PCP client's IP address replicated in the PCP header. This is used to detect address rewriting (NAT) between the PCP client and its PCP server. On operating systems that support the sockets API, the following steps are RECOMMENDED for a PCP client to insert the correct source address and port in the PCP header:

1. Create a UDP socket.
2. Call "connect" on this UDP socket using the address and port of the desired PCP server.
3. Call the getsockname() function to retrieve a sockaddr containing the source address the kernel will use for UDP packets sent through this socket.

4. If the IP address is an IPv4 address, encode the address into an IPv4-mapped IPv6 address. Place the native IPv6 address or IPv4-mapped IPv6 address into the PCP Client's IP Address field in the PCP header.
5. Send PCP requests using this connected UDP socket.

16.5. State Diagram

Each mapping entry of the PCP-controlled device would go through the state machine shown below. This state diagram is non-normative.

```
    CLOSE_MSG or
   (NO_TRAFFIC and EXPIRY)   +---------+  NO_TRAFFIC and EXPIRY
             +-------------->|         |<------------+
             |               |NO_ENTRY |             |
             |   +-----------|         |---------+   |
             |   |           +---------+         |   |
             |   |              ^  |              |   |
             |   |   NO_TRAFFIC |  |              |   |
             |   |           or |  |              |   |
             |   |   CLOSE_MSGS |  |              |   |
             |   |              |  |              |   |
             |   |PEER request  |  |  MAP request|   |
             |   V              |  |             V   |
          +---------+           |  |         +---------+
      +-->|  "P",   |           |  |    M-R  |  "M",   |<--+
  P-R |   | PEER    |-----------|--|-------->| MAP     |   | M-R or
      +---|  mapping|           |  |         |  mapping|---+ P-R or
          +---------+           |  |         +---------+  CLOSE_MSGS
             |   ^              |  |             ^   |
             |   |PEER request  |  |  MAP request|   |
             |   |              |  |              |   |
             |   |              |  |              |   |
             |   |              |  |              |   |
             |   |              |  | outbound     |   |
             |   |              |  | TRAFFIC      |   |
             |   |              |  V              |   |
             |   |           +---------+         |   |
             |   +-----------| "I",    |---------+   |
             |               | implicit|             |
             +-------------->| mapping |<------------+
         TRAFFIC and EXPIRY  +---------+  TRAFFIC and EXPIRY
```

Figure 16: PCP State Diagram

The meanings of the states and events are:

NO_ENTRY: Invalid state represents Entry does not exist. This is the only possible start state.

M-R: MAP request

P-R: PEER request

M: Mapping entry when created by MAP request

P: Mapping entry when created/managed by PEER request

I: Implicit mapping created by an outgoing packet from the client (e.g., TCP SYN), and also the state when a PCP-created mapping's lifetime expires while there is still active traffic.

EXPIRY: PEER or MAP lifetime expired

TRAFFIC: Traffic seen by PCP-controlled device using this entry within the expiry time for that entry. This traffic may be inbound or outbound.

NO_TRAFFIC: Indicates that there is no TRAFFIC.

CLOSE_MSG: Protocol messages from the client or server to close the session (e.g., TCP FIN or TCP RST), as per the NAT or firewall device's handling of such protocol messages.

Notes on the diagram:

1. The 'and' clause indicates the events on either side of 'and' are required for the state-transition. The 'or' clause indicates either one of the events are enough for the state-transition.

2. Transition from state M to state I is implementation dependent.

17. Deployment Considerations

17.1. Ingress Filtering

As with implicit dynamic mappings created by outgoing TCP SYN packets, explicit dynamic mappings created via PCP use the source IP address of the packet as the Internal Address for the mappings. Therefore ingress filtering [RFC2827] SHOULD be used on the path between the Internal Host and the PCP Server to prevent the injection

of spoofed packets onto that path.

17.2. Mapping Quota

On PCP-controlled devices that create state when a mapping is created (e.g., NAT), the PCP server SHOULD maintain per-host and/or per-subscriber quotas for mappings. It is implementation-specific whether the PCP server uses a separate quotas for implicit, explicit, and static mappings, a combined quota for all of them, or some other policy.

18. Security Considerations

The goal of the PCP protocol is to improve the ability of end nodes to control their associated NAT state, and to improve the efficiency and error handling of NAT mappings when compared to existing implicit mapping mechanisms in NAT boxes and stateful firewalls. It is the security goal of the PCP protocol to limit any new denial of service opportunities, and to avoid introducing new attacks that can result in unauthorized changes to mapping state. One of the most serious consequences of unauthorized changes in mapping state is traffic theft. All mappings that could be created by a specific host using implicit mapping mechanisms are inherently considered to be authorized. Confidentiality of mappings is not a requirement, even in cases where the PCP messages may transit paths that would not be travelled by the mapped traffic.

18.1. Simple Threat Model

PCP is secure against off-path attackers who cannot spoof a packet that the PCP Server will view as a packet received from the internal network. PCP is secure against off-path attackers who can spoof the PCP server’s IP address.

Defending against attackers who can modify or drop packets between the internal network and the PCP server, or who can inject spoofed packets that appear to come from the internal network is out of scope. Such an attacker can re-direct traffic to a host of their choosing.

A PCP Server is secure under this threat model if the PCP Server is constrained so that it does not configure any explicit mapping that it would not configure implicitly. In most cases, this means that PCP Servers running on NAT boxes or stateful firewalls that support the PEER and MAP Opcodes can be secure under this threat model if (1) all of their hosts are within a single administrative domain (or if the internal hosts can be securely partitioned into separate

administrative domains, as in the DS-Lite B4 case), (2) explicit mappings are created with the same lifetime as implicit mappings, and (3) the THIRD_PARTY option is not supported. PCP Servers can also securely support the MAP Opcode under this threat model if the security policy on the device running the PCP Server would permit endpoint independent filtering of implicit mappings.

PCP Servers that comply with the Simple Threat Model and do not implement a PCP security mechanism described in Section 18.2 MUST enforce the constraints described in the paragraph above.

18.1.1. Attacks Considered

- If you allow multiple administrative domains to send PCP requests to a single PCP server that does not enforce a boundary between the domains, it is possible for a node in one domain to perform a denial of service attack on other domains, or to capture traffic that is intended for a node in another domain.

- If explicit mappings have longer lifetimes than implicit mappings, it makes it easier to perpetrate a denial of service attack than it would be if the PCP Server was not present.

- If the PCP Server supports deleting or reducing the lifetime of existing mappings, this allows an attacking node to steal an existing mapping and receive traffic that was intended for another node.

- If the THIRD_PARTY Option is supported, this also allows an attacker to open a window for an external node to attack an internal node, allows an attacker to steal traffic that was intended for another node, or may facilitate a denial of service attack. One example of how the THIRD_PARTY Option could grant an attacker more capability than a spoofed implicit mapping is that the PCP server (especially if it is running in a service provider’s network) may not be aware of internal filtering that would prevent spoofing an equivalent implicit mapping, such as filtering between a guest and corporate network.

- If the MAP Opcode is supported by the PCP server in cases where the security policy would not support endpoint independent filtering of implicit mappings, then the MAP Opcode changes the security properties of the device running the PCP Server by allowing explicit mappings that violate the security policy.

18.1.2. Deployment Examples Supporting the Simple Threat Model

This section offers two examples of how the Simple Threat Model can be supported in real-world deployment scenarios.

18.1.2.1. Residential Gateway Deployment

Parity with many currently-deployed residential gateways can be achieved using a PCP Server that is constrained as described in Section 18.1 above.

18.2. Advanced Threat Model

In the Advanced Threat Model the PCP protocol ensures that attackers (on- or off-path) cannot create unauthorized mappings or make unauthorized changes to existing mappings. The protocol must also limit the opportunity for on- or off-path attackers to perpetrate denial of service attacks.

The Advanced Threat Model security model will be needed in the following cases:

- o Security infrastructure equipment, such as corporate firewalls, that does not create implicit mappings.

- o Equipment (such as CGNs or service provider firewalls) that serve multiple administrative domains and do not have a mechanism to securely partition traffic from those domains.

- o Any implementation that wants to be more permissive in authorizing explicit mappings than it is in authorizing implicit mappings.

- o Implementations that wish to support any deployment scenario that does not meet the constraints described in Section 18.1.

To protect against attacks under this threat model, a PCP security mechanism that provides an authenticated, integrity-protected signaling channel would need to be specified.

PCP Servers that implement a PCP security mechanism MAY accept unauthenticated requests. PCP Servers implementing the PCP security mechanism MUST enforce the constraints described in Section 18.1 above, in their default configuration, when processing unauthenticated requests.

18.3. Residual Threats

This section describes some threats that are not addressed in either of the above threat models, and recommends appropriate mitigation strategies.

18.3.1. Denial of Service

Because of the state created in a NAT or firewall, a per-host and/or per-subscriber quota will likely exist for both implicit dynamic mappings and explicit dynamic mappings. A host might make an excessive number of implicit or explicit dynamic mappings, consuming an inordinate number of ports, causing a denial of service to other hosts. Thus, Section 17.2 recommends that hosts be limited to a reasonable number of explicit dynamic mappings.

An attacker, on the path between the PCP client and PCP server, can drop PCP requests, drop PCP responses, or spoof a PCP error, all of which will effectively deny service. Through such actions, the PCP client might not be aware the PCP server might have actually processed the PCP request. An attacker sending a NO_RESOURCES error can cause the PCP client to not send messages to that server for a while. There is no mitigation to this on-path attacker.

18.3.2. Ingress Filtering

It is important to prevent a host from fraudulently creating, deleting, or refreshing a mapping (or filtering) for another host, because this can expose the other host to unwanted traffic, prevent it from receiving wanted traffic, or consume the other host’s mapping quota. Both implicit and explicit dynamic mappings are created based on the source IP address in the packet, and hence depend on ingress filtering to guard against spoof source IP addresses.

18.3.3. Mapping Theft

In the time between when a PCP server loses state and the PCP client notices the lower-than-expected Epoch Time value, it is possible that the PCP client’s mapping will be acquired by another host (via an explicit dynamic mapping or implicit dynamic mapping). This means incoming traffic will be sent to a different host ("theft"). Rapid Recovery reduces this interval, but would not completely eliminate this threat. The PCP client can reduce this interval by using a relatively short lifetime; however, this increases the amount of PCP chatter. This threat is reduced by using persistent storage of explicit dynamic mappings in the PCP server (so it does not lose explicit dynamic mapping state), or by ensuring the previous external IP address, protocol, and port cannot be used by another host (e.g.,

by using a different IP address pool).

18.3.4. Attacks Against Server Discovery

This document does not specify server discovery, beyond contacting the default gateway.

19. IANA Considerations

IANA is requested to perform the following actions:

19.1. Port Number

PCP will use ports 5350 and 5351 (currently assigned by IANA to NAT-PMP [I-D.cheshire-nat-pmp]). We request that IANA re-assign those ports to PCP, and relinquish UDP port 44323.

[Note to RFC Editor: Please remove the text about relinquishing port 44323 prior to publication.]

19.2. Opcodes

IANA shall create a new protocol registry for PCP Opcodes, numbered 0-127, initially populated with the values:

value	Opcode
0	ANNOUNCE
1	MAP
2	PEER
3-31	Standards Action [RFC5226]
32-63	Specification Required [RFC5226]
96-126	Private Use [RFC5226]
127	Reserved, Standards Action [RFC5226]

The value 127 is Reserved and may be assigned via Standards Action [RFC5226]. The values in the range 3-31 can be assigned via Standards Action [RFC5226], 32-63 via Specification Required [RFC5226], and 96-126 is for Private Use [RFC5226].

19.3. Result Codes

IANA shall create a new registry for PCP result codes, numbered 0-255, initially populated with the result codes from Section 7.4. The value 255 is Reserved and may be assigned via Standards Action [RFC5226].

The values in the range 14-127 can be assigned via Standards Action [RFC5226], 128-191 via Specification Required [RFC5226], and 191-254 is for Private Use [RFC5226].

19.4. Options

IANA shall create a new registry for PCP Options, numbered 0-255, each with an associated mnemonic. The values 0-127 are mandatory-to-process, and 128-255 are optional to process. The initial registry contains the Options described in Section 13. The Option values 0, 127 and 255 are Reserved and may be assigned via Standards Action [RFC5226].

Additional PCP Option codes in the ranges 4-63 and 128-191 can be created via Standards Action [RFC5226], the ranges 64-95 and 192-223 are for Specification Required [RFC5226] and the ranges 96-126 and 224-254 are for Private Use [RFC5226].

Documents describing an Option should describe if the processing for both the PCP client and server and the information below:

```
Option Name: <mnemonic>
Number: <value>
Purpose: <textual description>
Valid for Opcodes: <list of Opcodes>
Length: <rules for length>
May appear in: <requests/responses/both>
Maximum occurrences: <count>
```

20. Acknowledgments

Thanks to Xiaohong Deng, Alain Durand, Christian Jacquenet, Jacni Qin, Simon Perreault, James Yu, Tina TSOU (Ting ZOU), Felipe Miranda Costa, James Woodyatt, Dave Thaler, Masataka Ohta, Vijay K. Gurbani, Loa Andersson, Richard Barnes, Russ Housley, Adrian Farrel, Pete Resnick, Pasi Sarolahti, Robert Sparks, Wesley Eddy, Dan Harkins, Peter Saint-Andre, Stephen Farrell, Ralph Droms, Felipe Miranda Costa, Amit Jain, and Wim Henderickx for their comments and review.

Thanks to Simon Perreault for highlighting the interaction of dynamic connections with PCP-created mappings.

Thanks to Francis Dupont for his several thorough reviews of the specification, which improved the protocol significantly.

Thanks to T. S. Ranganathan for the state diagram.

Thanks to Peter Lothberg for clock skew information.

Thanks to Margaret Wasserman and Sam Hartman for writing the Security Considerations section.

Thanks to authors of DHCPv6 for retransmission text.

21. References

21.1. Normative References

[RFC0768] Postel, J., "User Datagram Protocol", STD 6, RFC 768, August 1980.

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

[RFC2827] Ferguson, P. and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing", BCP 38, RFC 2827, May 2000.

[RFC4086] Eastlake, D., Schiller, J., and S. Crocker, "Randomness Requirements for Security", BCP 106, RFC 4086, June 2005.

[RFC4193] Hinden, R. and B. Haberman, "Unique Local IPv6 Unicast Addresses", RFC 4193, October 2005.

[RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, February 2006.

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.

[RFC6056] Larsen, M. and F. Gont, "Recommendations for Transport-Protocol Port Randomization", BCP 156, RFC 6056, January 2011.

[proto_numbers]
IANA, "Protocol Numbers", 2011, <http://www.iana.org/assignments/protocol-numbers/protocol-numbers.xml>.

21.2. Informative References

[I-D.boucadair-pcp-failure]
Boucadair, M., Dupont, F., and R. Penno, "Port Control Protocol (PCP) Failure Scenarios", draft-boucadair-pcp-failure-04 (work in progress), August 2012.

[I-D.cheshire-dnsext-dns-sd]
Cheshire, S. and M. Krochmal, "DNS-Based Service Discovery", draft-cheshire-dnsext-dns-sd-11 (work in progress), December 2011.

[I-D.cheshire-nat-pmp]
Cheshire, S. and M. Krochmal, "NAT Port Mapping Protocol (NAT-PMP)", draft-cheshire-nat-pmp-05 (work in progress), September 2012.

[I-D.ietf-behave-lsn-requirements]
Perreault, S., Yamagata, I., Miyakawa, S., Nakagawa, A., and H. Ashida, "Common requirements for Carrier Grade NATs (CGNs)", draft-ietf-behave-lsn-requirements-09 (work in progress), August 2012.

[I-D.ietf-behave-sctpnat]
Stewart, R., Tuexen, M., and I. Ruengeler, "Stream Control Transmission Protocol (SCTP) Network Address Translation", draft-ietf-behave-sctpnat-07 (work in progress), October 2012.

[I-D.ietf-pcp-upnp-igd-interworking]
Boucadair, M., Dupont, F., Penno, R., and D. Wing, "Universal Plug and Play (UPnP) Internet Gateway Device (IGD)-Port Control Protocol (PCP) Interworking Function", draft-ietf-pcp-upnp-igd-interworking-04 (work in progress), September 2012.

[I-D.miles-behave-l2nat]
Miles, D. and M. Townsley, "Layer2-Aware NAT", draft-miles-behave-l2nat-00 (work in progress), March 2009.

[IGDv1] UPnP Gateway Committee, "WANIPConnection:1", November 2001, <http://upnp.org/specs/gw/UPnP-gw-WANIPConnection-v1-Service.pdf>.

[RFC0793] Postel, J., "Transmission Control Protocol", STD 7, RFC 793, September 1981.

[RFC1918] Rekhter, Y., Moskowitz, R., Karrenberg, D., Groot, G., and E. Lear, "Address Allocation for Private Internets", BCP 5, RFC 1918, February 1996.

[RFC2136] Vixie, P., Thomson, S., Rekhter, Y., and J. Bound, "Dynamic Updates in the Domain Name System (DNS UPDATE)", RFC 2136, April 1997.

[RFC3007] Wellington, B., "Secure Domain Name System (DNS) Dynamic Update", RFC 3007, November 2000.

[RFC3022] Srisuresh, P. and K. Egevang, "Traditional IP Network Address Translator (Traditional NAT)", RFC 3022, January 2001.

[RFC3581] Rosenberg, J. and H. Schulzrinne, "An Extension to the Session Initiation Protocol (SIP) for Symmetric Response Routing", RFC 3581, August 2003.

[RFC3587] Hinden, R., Deering, S., and E. Nordmark, "IPv6 Global Unicast Address Format", RFC 3587, August 2003.

[RFC4303] Kent, S., "IP Encapsulating Security Payload (ESP)", RFC 4303, December 2005.

[RFC4340] Kohler, E., Handley, M., and S. Floyd, "Datagram Congestion Control Protocol (DCCP)", RFC 4340, March 2006.

[RFC4787] Audet, F. and C. Jennings, "Network Address Translation (NAT) Behavioral Requirements for Unicast UDP", BCP 127, RFC 4787, January 2007.

[RFC4941] Narten, T., Draves, R., and S. Krishnan, "Privacy Extensions for Stateless Address Autoconfiguration in IPv6", RFC 4941, September 2007.

[RFC4960] Stewart, R., "Stream Control Transmission Protocol", RFC 4960, September 2007.

[RFC4961] Wing, D., "Symmetric RTP / RTP Control Protocol (RTCP)", BCP 131, RFC 4961, July 2007.

[RFC5382] Guha, S., Biswas, K., Ford, B., Sivakumar, S., and P. Srisuresh, "NAT Behavioral Requirements for TCP", BCP 142, RFC 5382, October 2008.

[RFC6092] Woodyatt, J., "Recommended Simple Security Capabilities in Customer Premises Equipment (CPE) for Providing Residential IPv6 Internet Service", RFC 6092, January 2011.

[RFC6145] Li, X., Bao, C., and F. Baker, "IP/ICMP Translation Algorithm", RFC 6145, April 2011.

[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.

[RFC6296] Wasserman, M. and F. Baker, "IPv6-to-IPv6 Network Prefix Translation", RFC 6296, June 2011.

[RFC6333] Durand, A., Droms, R., Woodyatt, J., and Y. Lee, "Dual-Stack Lite Broadband Deployments Following IPv4 Exhaustion", RFC 6333, August 2011.

[RFC6619] Arkko, J., Eggert, L., and M. Townsley, "Scalable Operation of Address Translators with Per-Interface Bindings", RFC 6619, June 2012.

Appendix A. NAT-PMP Transition

The Port Control Protocol (PCP) is a successor to the NAT Port Mapping Protocol, NAT-PMP [I-D.cheshire-nat-pmp], and shares similar semantics, concepts, and packet formats. Because of this NAT-PMP and PCP both use the same port, and use NAT-PMP and PCP’s version negotiation capabilities to determine which version to use. This section describes how an orderly transition may be achieved.

A client supporting both NAT-PMP and PCP SHOULD send its request using the PCP packet format. This will be received by a NAT-PMP server or a PCP server. If received by a NAT-PMP server, the response will be as indicated by the NAT-PMP specification [I-D.cheshire-nat-pmp], which will cause the client to downgrade to NAT-PMP and re-send its request in NAT-PMP format. If received by a PCP server, the response will be as described by this document and processing continues as expected.

A PCP server supporting both NAT-PMP and PCP can handle requests in either format. The first octet of the packet indicates if it is NAT-PMP (first octet zero) or PCP (first octet non-zero).

A PCP-only gateway receiving a NAT-PMP request (identified by the first octet being zero) will interpret the request as a version mismatch. Normal PCP processing will emit a PCP response that is compatible with NAT-PMP, without any special handling by the PCP server.

Appendix B. Change History

[Note to RFC Editor: Please remove this section prior to publication.]

B.1. Changes from draft-ietf-pcp-base-28 to -29

- o Removed text suggesting PCP client can remove old mappings when it acquires a new IP address.

B.2. Changes from draft-ietf-pcp-base-27 to -28

- o When processing MAP request or processing PEER request, Mapping Nonce validation only applies to Basic Threat Model, and not to THIRD_PARTY.

- o A maximum payload size of 1100 keeps PCP packets below IPv6's 1280 MTU limit while still allowing some room for encapsulation. This accomodates EAP over PANA over PCP (EAP needs 1020 octets, per RFC3748), should PCP authentication decide to use EAP over PANA over PCP.

- o Both MAP and PEER-created mappings cannot have their lifetimes reduced beyond normal UDP/TCP timesouts.

- o Disallow re-assigning External Port to same internal host.

B.3. Changes from draft-ietf-pcp-base-26 to -27

- o For table, reverted the NAT64 remote peer to IPv6 -- because from the IPv6 PCP client's perspective, the remote peer really is IPv6.

- o "list of PCP server addresses" changed to "longer list of PCP server addresses"

- o Clarify that unsolicited ANNOUNCE messages are sent from the PCP server IP address and PCP port.

- o "1024 bytes" changed to "1024 octets".

- o Clarify that re-transmitted requests must use same Mapping Nonce value (beginning of Section 8.1.1).

- o Describe that de-synchronization that can occur (end of Section 8.1.1).

- o For devices that lose state or expect IP renumbering, Rapid Recovery is now a MUST, with SHOULD for implementing both multicast Announce mechanism and unicast mechanisms.

- o For refreshing MAP or PEER, Mapping Nonce has to match the previous MAP or PEER. This protects from off-path attackers stealing MAP or shortening PEER mappings.

- o With the Mapping Nonce change, we now allow PEER to reduce mapping lifetime to same lifetime as implicit mapping lifetime (but not shorter). Changes for this are in both PEER section and Security Considerations.

- o With Mapping Nonce change, can no longer delete a 'set of mappings' (because we cannot send multiple Mapping Nonce values), so removed text that allowed that.

- o Send Mapping Update only 3 times (used to be 10 times).

- o General PCP processing now requires validating Mapping Nonce, if the opcode uses a Mapping Nonce Section 8.3.

- o Moved text describing NO_RESOURCES handling from General Processing section to MAP and PEER processing sections, as it NO_RESOURCES processing should be done after validating Mapping Nonce.

- o Clarified SCTP NAT behavior (port numbers stay the same, causing grief).

- o added EIM definition.

- o Clarified Mapping Type definitions.

- o PCP Client definition simplified to no longer obliquely and erroneously reference UPnP IGD.

- o Clarified using network-byte order.

- o Epoch time comparison now allows slight packet re-ordering.

- o Encourage that when new address is assigned (e.g., DHCP) that PCP as well as non-PCP mappings be cleaned up.

- o Simplified formatting of retransmission, but no normative change.

- o Clarified how server chooses ports and how Suggested External Port can gently influence that decision.

- o Described how PCP client can use PCP Client Address with a non-PCP-aware inner NAT (Section 8.1.)

- o Clarified 1024 octet length applies to UDP payload itself, and that error responses copy 1024 of UDP payload.

- o Lifetime for both MAP and PEER should not exceed the remaining IP address lifetime of the PCP client (if known) or half the typical IP address lifetime (if the remaining lifetime is unknown).

- o Lifetime section was (mistakenly) a subsection of the MAP section, but referenced by both MAP and PEER. It is now a top-level section.

- o Clarified that PEER cannot reduce lifetime beyond normal implicit mapping lifetime, no matter what. This restriction prevents malicious or accidental deletion of a quiescent connection that was not using PCP.

- o Clarified port re-use of PCP-created mappings should follow same port re-use algorithm used by the NAT for implicitly-created mappings (likely maximum segment lifetime).

- o Other minor text changes; consult diffs.

B.4. Changes from draft-ietf-pcp-base-25 to -26

- o Changed "internal address and port" to "internal address, protocol, and port" in several more places.

- o Improved wording of THIRD_PARTY restrictions.

- o Bump version number from 1 to 2, to accommodate pre-RFC PCP client implementations without needing a heuristic.

B.5. Changes from draft-ietf-pcp-base-24 to -25

- o Clarified the port used by the PCP server when sending unsolicited unicast ANNOUNCE.

- o Removed parenthetical comment implying ANNOUNCE was not a normal Opcode; it is a normal Opcode.

- o Explain that non-PCP-speaking host-based and network-based firewalls need to allow incoming connections for MAP to work.

- o For race condition with PREFER_FAILURE, clarified that it is the PCP client’s responsibility to delete the mapping if the PCP client doesn’t need the mapping.

- o For table, the NAT64 remote peer is IPv4 (was IPv6).

- o Added a Mapping Nonce field to both MAP and PEER requests and responses, to protect from off-path attackers spoofing the PCP

server’s IP address.

- o Security considerations: added ’PCP is secure against off-path attackers who can spoof the PCP server’s IP address’, because of the addition of the Mapping Nonce.

- o Removed reference to DS-Lite from Security Considerations, as part of the changes to THIRD_PARTY from IESG review.

- o Rapid Recovery is now a SHOULD implement.

- o Clarify behavior of PREFER_FAILURE with zeros in Suggested External Port or Address fields.

- o PCP server is now more robust and insistent about informing PCP client of state changes.

- o When PCP server sends Mapping Update to a specific PCP client, and gets an update for a particular mapping, it doesn’t need to send reminders about that mapping any more.

- o THIRD_PARTY is now prohibited on subscriber PCP clients.

B.6. Changes from draft-ietf-pcp-base-23 to -24

- o Explained common questions regarding PCP’s design, such as lack of transction identifiers and its request/response semantics and operation (Protocol Design Note (Section 6)).

- o added MUST for all-zeros IPv6 and IPv4 address formats.

- o included field definitions for Opcode-specific information and PCP options under both Figure 2 and Figure 3.

- o adopted retransmission mechanism from DHCPv6.

- o 1024 message size limit described in PCP message restriction.

- o Explained PCP server list, with example of host with IPv4 and IPv6 addresses having two PCP servers (one IPv4 PCP server for IPv4 mappings and one IPv6 PCP server for IPv6 mappings).

- o mention PCP client needs to expect unsolicited PCP responses from previous incarnations of itself (on the same host) or of this host (using same IP address as another PCP client).

- o eliminated overuse of ’packet format’ when it was ’opcode format’.

- o for IANA registries, added code points assignable via Standards Action (previously was just Specification Required).

- o Version negotiation, added explanation that retrying after 30 minutes makes the protocol self-healing if the PCP server is upgraded.

- o Version negotiation now accomodates non-contiguous version numbers.

- o Tweaked definition of VERSION field (that "1" is for this version, but other values could of course appear in the future).

- o when receiving unsolicited ANNOUNCE, PCP client now waits random 0-5 seconds.

- o Removed ’interworking function’ from list of terminology because we no longer use the term in this document.

- o tightened definitions of ’PCP client’ and ’PCP server’.

- o For ’Requested Lifetime’ definitions, removed text requiring its value be 0 for not-yet-defined opcodes.

- o Removed some unnecessary text suggesting logging (is an implementation detail).

- o Added active-mode FTP as example protocol that can break with mappings to different IP addresses.

- o Clarified that if PCP request contains a Suggested External Address, the PCP server should try to create a mapping to that address even if other mappings already exist to a different external address.

- o Changed "internal address and port" to "internal address, protocol, and port" in several places.

- o Clarified which 96 bits are copied into error response. Clarified that only error responses are copied verbatim from request.

- o a single PCP server can control multiple NATs or multiple firewalls (Section 4).

- o Clarified that sending unsolicited multicast ANNOUNCE is not always available on all networks.

- o Clarified option length error example is when option length exceeds UDP length

- o Explained that an on-path attacker that can spoof packets can redirect traffic to a host of their choosing.

- o Instead of saying IPv4-mapped addresses won't appear on the wire, say they aren't used for mappings.

- o THIRD_PARTY is useful for management device (e.g., in a network operations center).

- o Clarified PCP responses have fields updated as indicated with 'set by the server' from field definitions.

- o Disallow using MAP to the PCP ports themselves and encourage implementations have policy control for other ports.

- o Instead of 'idempotent', now says 'identical requests generate identical response'.

- o Described which Options are included when sending Mapping Update (unsolicited responses), Section 14.2.

- o Dropped [RFC2136] and [RFC3007] to informative references.

- o Updated from 'should' to 'SHOULD' in Section 17.1.

- o Described 'hairpin' in terminology section.

B.7. Changes from draft-ietf-pcp-base-22 to -23

- o Instead of returning error NO_RESOURCES when requesting a MAP for all protocols or for all ports, return UNSUPP_PROTOCOL.

- o Clarify that PEER-created mappings are treated as if it was implicit dynamic outbound mapping (Section 12.3).

- o Point out that PEER-created mappings may be very short until bidirectional traffic is seen by the PCP-managed device.

- o Clairification that an existing implicit mapping (created e.g., by TCP SYN) can become managed by a MAP request (Section 11.3.

- o Clarified the ANNOUNCE Opcode is being defined in Section 14.1, and that the length of requests (as well as responses) is zero.

- o Clarify that ANNOUNCE has Lifetime=0 for requests and responses.
- o Clarify ANNOUNCE can be sent unicast by the client (to solicit a response), or can be multicasted (unsolicited) by the server.
- o Allow ANNOUNCE to be sent unicast by the server, to accomodate case where PCP server fails but knows the IP address of a PCP client (e.g., web portal).
- o Clarified ports used for unicast and multicast unsolicited ANNOUNCE.
- o Tweaked NO_RESOURCES handling, to just disallow *new* mappings.
- o State diagram is now non-normative, because it overly simplifies that implicit mappings become MAP (when they actually still retain their previous behavior when the MAP expires).
- o In section Section 15, clarified that PEER cannot delete or shorten any lifetime, and that MAP can only shorten or delete lifetimes of MAP-created mappings.
- o Clarified handling of MAP when mapping already exists (4 steps).
- o 2^32-1
- o Randomize retry interval (1.5-2.5), and maximum retry interval is now 1024 seconds (was 15 minutes).
- o Remove MUST be 0 for Reserved field when sending error responses for un-parseable message.
- o Whenever PCP client includes Suggested IP Address (in MAP or PEER), the PCP server should try to fulfill that request, even if creating a mapping on that IP address means the internal host will have mappings on different IP addresses and ports.
- o For NO_RESOURCES error, the PCP client can attempt to renew and attempt to delete mappings (as they can help shed load) -- it just can’t try to create new ones.
- o Removed the overly simplistic normative text regarding honoring Suggested External Address from Section 10 in favor of the text in Section 11.3 which has significantly more detail.

B.8. Changes from draft-ietf-pcp-base-21 to -22

- o Removed paragraph discussing multiple addresses on the same (physical) interface; those will work with PCP.
- o The FILTER Option's Prefix Length field redefined to simply be a count of the relevant bits (rather than 0-32 for IPv4-mapped addresses).
- o Point out NO_RESOURCES attack vector in security considerations.
- o Tighten up recommendation for client handling long Lifetimes, and moved from the MAP-specific section to the General PCP Processing section. Client should normalize to 24 hours maximum for success and 30 minute maximum for errors.

B.9. Changes from draft-ietf-pcp-base-20 to -21

- o To delete all mappings using THIRD_PARTY, use the all-zeros IP address (rather than previous text which used length=0).
- o added normative text for what PCP server does when it receives all-zeros IP address in THIRD_PARTY option.
- o PREFER_FAILURE allowed for use by web portal.
- o clarifications to mandatory option processing.
- o cleanup and wordsmithing of the THIRD_PARTY text.

B.10. Changes from draft-ietf-pcp-base-19 to -20

- o clarify if Options are included in responses.
- o clarify when External Address can be ignored by the PCP server / PCP-controlled device
- o added 'Transition from state M to state I is implementation dependent' to state diagram

B.11. Changes from draft-ietf-pcp-base-18 to -19

- o Described race condition with MAP containing PREFER_FAILURE and Mapping Update.
- o Added state machine (Section 16.5).

- o Fully integrated Rapid Recovery, with a separate Opcode having its own processing description.

- o Clarified that due to Mapping Update, a single MAP or PEER request can receive multiple responses, each updating the previous request, and that the PCP client needs to handle MAP updates or PEER updates accordingly.

B.12. Changes from draft-ietf-pcp-base-17 to -18

- o Removed UNPROCESSED option. Instead, unprocessed options are simply not included in responses.

- o Updated terminology section for Implicit/Explicit and Outbound/ Inbound.

- o PEER requests cannot delete or shorten the lifetime of a mapping.

- o Clarified that PCP clients only retransmit mapping requests for as long as they actually want the mapping.

- o Revised Epoch time calculations and explanation.

- o Renamed the announcement opcode from No-Op to ANNOUNCE.

B.13. Changes from draft-ietf-pcp-base-16 to -17

- o suggest acquiring a mapping to the Discard port if there is a desire to show the user their external address (Section 11.6).

- o Added Restart Announcement.

- o Tweaked terminology.

- o Detailed how error responses are generated.

B.14. Changes from draft-ietf-pcp-base-15 to -16

- o fixed mistake in PCP request format (had 32 bits of extraneous fields)

- o Allow MAP to request all ports (port=0) for a specific protocol (protocol!=0), for the same reason we added support for all ports (port=0) and all protocols (protocol=0) in -15

- o corrected text on Client Processing a Response related to receiving ADDRESS_MISMATCH error.

- o updated Epoch text.

- o Added text that MALFORMED_REQUEST is generated for MAP if Protocol is zero but Internal Port is non-zero.

B.15. Changes from draft-ietf-pcp-base-14 to -15

- o Softened and removed text that was normatively explaining how PEER is implemented within a NAT.

- o Allow a MAP request for protocol=0, which means "all protocols". This can work for an IPv6 or IPv4 firewall. Its use with a NAPT is undefined.

- o combined SERVER_OVERLOADED and NO_RESOURCES into one error code, NO_RESOURCES.

- o SCTP mappings have to use same internal and suggested external ports, and have implied PREFER_FAILURE semantics.

- o Re-instated ADDRESS_MISMATCH error, which only checks the client address (not its port).

B.16. Changes from draft-ietf-pcp-base-13 to -14

- o Moved discussion of socket operations for PCP source address into Implementation Considerations section.

- o Integrated numerous WGLC comments.

- o NPTv6 in scope.

- o Re-written security considerations section. Thanks, Margaret!

- o Reduced PEER4 and PEER6 Opcodes to just a single Opcode, PEER.

- o Reduced MAP4 and MAP6 Opcodes to just a single Opcode, MAP.

- o Rearranged the PEER packet formats to align with MAP.

- o Removed discussion of the "O" bit for Options, which was confusing. Now the text just discusses the most significant bit of the Option code which indicates mandatory/optional, so it is clearer the field is 8 bits.

- o The THIRD_PARTY Option from an unauthorized host generates UNSUPP_OPTION, so the PCP server doesn’t disclose it knows how to process THIRD_PARTY Option.

- o Added table to show which fields of MAP or PEER need IPv6/IPv4 addresses for IPv4 firewall, DS-Lite, NAT64, NAT44, etc.

- o Accommodate the server's Epoch going up or down, to better detect switching to a different PCP server.

- o Removed ADDRESS_MISMATCH; the server always includes its idea of the Client's IP Address and Port, and it's up to the client to detect a mismatch (and rectify it).

B.17. Changes from draft-ietf-pcp-base-12 to -13

- o All addresses are 128 bits. IPv4 addresses are represented by IPv4-mapped IPv6 addresses (::FFFF/96)

- o PCP request header now includes PCP client's port (in addition to the client's IP address, which was in -12).

- o new ADDRESS_MISMATCH error.

- o removed PROCESSING_ERROR error, which was too similar to MALFORMED_REQUEST.

- o Tweaked text describing how PCP client deals with multiple PCP server addresses (Section 8.1)

- o clarified that when overloaded, the server can send SERVER_OVERLOADED (and drop requests) or simply drop requests.

- o Clarified how PCP client chooses MAP4 or MAP6, depending on the presence of its own IPv6 or IPv4 interfaces (Section 10).

- o compliant PCP server MUST support MAPx and PEERx, SHOULD support ability to disable support.

- o clarified that MAP-created mappings have no filtering, and PEER-created mappings have whatever filtering and mapping behavior is normal for that particular NAT / firewall.

- o Integrated WGLC feedback (small changes to abstract, definitions, and small edits throughout the document)

- o allow new Options to be defined with a specification (rather than standards action)

B.18. Changes from draft-ietf-pcp-base-11 to -12

- o added implementation note that MAP and implicit dynamic mappings have independent mapping lifetimes.

B.19. Changes from draft-ietf-pcp-base-10 to -11

- o clarified what can cause CANNOT_PROVIDE_EXTERNAL error to be generated.

B.20. Changes from draft-ietf-pcp-base-09 to -10

- o Added External_AF field to PEER requests. Made PEER's Suggested External IP Address and Assigned External IP Address always be 128 bits long.

B.21. Changes from draft-ietf-pcp-base-08 to -09

- o Clarified in PEER Opcode introduction (Section 12) that they can also create mappings.
- o More clearly explained how PEER can re-create an implicit dynamic mapping, for purposes of rebuilding state to maintain an existing session (e.g., long-lived TCP connection to a server).
- o Added Suggested External IP Address to the PEER Opcodes, to allow more robust rebuilding of connections. Added related text to the PEER server processing section.
- o Removed text encouraging PCP server to statefully remember its mappings from Section 16.3.1, as it didn't belong there. Text in Security Considerations already encourages persistent storage.
- o More clearly discussed how PEER is used to re-establish TCP mapping state. Moved it to a new section, as well (it is now Section 10.4).
- o MAP errors now copy the Suggested Address (and port) fields to Assigned IP Address (and port), to allow PCP client to distinguish among many outstanding requests when using PREFER_FAILURE.
- o Mapping theft can also be mitigated by ensuring hosts can't re-use same IP address or port after state loss.
- o the UNPROCESSED option is renumbered to 0 (zero), which ensures no other option will be given 0 and be unable to be expressed by the UNPROCESSED option (due to its 0 padding).

- o created new Implementation Considerations section (Section 16) which discusses non-normative things that might be useful to implementers. Some new text is in here, and the Failure Scenarios text (Section 16.3) has been moved to here.

- o Tweaked wording of EDM NATs in Section 16.1 to clarify the problem occurs both inside->outside and outside->inside.

- o removed "Interference by Other Applications on Same Host" section from security considerations.

- o fixed zero/non-zero text in Section 15.

- o removed duplicate text saying MAP is allowed to delete an implicit dynamic mapping. It is still allowed to do that, but it didn’t need to be said twice in the same paragraph.

- o Renamed error from UNAUTH_TARGET_ADDRESS to UNAUTH_THIRD_PARTY_INTERNAL_ADDRESS.

- o for FILTER option, removed unnecessary detail on how FILTER would be bad for PEER, as it is only allowed for MAP anyway.

- o In Security Considerations, explain that PEER can create a mapping which makes its security considerations the same as MAP.

B.22. Changes from draft-ietf-pcp-base-07 to -08

- o moved all MAP4-, MAP6-, and PEER-specific options into a single section.

- o discussed NAT port-overloading and its impact on MAP (new section Section 16.1), which allowed removing the IMPLICIT_MAPPING_EXISTS error.

- o eliminated NONEXIST_PEER error (which was returned if a PEER request was received without an implicit dynamic mapping already being created), and adjusted PEER so that it creates an implicit dynamic mapping.

- o Removed Deployment Scenarios section (which detailed NAT64, NAT44, Dual-Stack Lite, etc.).

- o Added Client’s IP Address to PCP common header. This allows server to refuse a PCP request if there is a mismatch with the source IP address, such as when a non-PCP-aware NAT was on the path. This should reduce failure situations where PCP is deployed in conjunction with a non-PCP-aware NAT. This addition was

consensus at IETF80.

- Changed UNSPECIFIED_ERROR to PROCESSING_ERROR. Clarified that MALFORMED_REQUEST is for malformed requests (and not related to failed attempts to process the request).

- Removed MISORDERED_OPTIONS. Consensus of IETF80.

- SERVER_OVERLOADED is now a common PCP error (instead of specific to MAP).

- Tweaked PCP retransmit/retry algorithm again, to allow more aggressive PCP discovery if an implementation wants to do that.

- Version negotiation text tweaked to soften NAT-PMP reference, and more clearly explain exactly what UNSUPP_VERSION should return.

- PCP now uses NAT-PMP’s UDP port, 5351. There are no normative changes to NAT-PMP or PCP to allow them both to use the same port number.

- New Appendix A to discuss NAT-PMP / PCP interworking.

- improved pseudocode to be non-blocking.

- clarified that PCP cannot delete a static mapping (i.e., a mapping created by CLI or other non-PCP means).

- moved theft of mapping discussion from Epoch section to Security Considerations.

B.23. Changes from draft-ietf-pcp-base-06 to -07

- tightened up THIRD_PARTY security discussion. Removed "highest numbered address", and left it as simply "the CPE’s IP address".

- removed UNABLE_TO_DELETE_ALL error.

- renumbered Opcodes

- renumbered some error codes

- assigned value to IMPLICIT_MAPPING_EXISTS.

- UNPROCESSED can include arbitrary number of option codes.

- Moved lifetime fields into common request/response headers

- o We've noticed we're having to repeatedly explain to people that the "requested port" is merely a hint, and the NAT gateway is free to ignore it. Changed name to "suggested port" to better convey this intention.

- o Added NAT-PMP transition section

- o Separated Internal Address, External Address, Remote Peer Address definition

- o Unified Mapping, Port Mapping, Port Forwarding definition

- o adjusted so DHCP configuration is non-normative.

- o mentioned PCP refreshes need to be sent over the same interface.

- o renamed the REMOTE_PEER_FILTER option to FILTER.

- o Clarified FILTER option to allow sending an ICMP error if policy allows.

- o for MAP, clarified that if the PCP client changed its IP address and still wants to receive traffic, it needs to send a new MAP request.

- o clarified that PEER requests have to be sent from same interface as the connection itself.

- o for MAP opcode, text now requires mapping be deleted when lifetime expires (per consensus on 8-Mar interim meeting)

- o PEER Opcode: better description of remote peer's IP address, specifically that it does not control or establish any filtering, and explaining why it is 'from the PCP client's perspective'.

- o Removed latent text allowing DMZ for 'all protocols' (protocol=0). Which wouldn't have been legal, anyway, as protocol 0 is assigned by IANA to HOPOPT (thanks to James Yu for catching that one).

- o clarified that PCP server only listens on its internal interface.

- o abandoned 'target' term and reverted to simplier 'internal' term.

B.24. Changes from draft-ietf-pcp-base-05 to -06

- o Dual-Stack Lite: consensus was encapsulation mode. Included a suggestion that the B4 will need to proxy PCP-to-PCP and UPnP-to-PCP.

- o defined THIRD_PARTY Option to work with the PEER Opcode, too. This meant moving it to its own section, and having both MAP and PEER Opcodes reference that common section.

- o used "target" instead of "internal", in the hopes that clarifies internal address used by PCP itself (for sending its packets) versus the address for MAPpings.

- o Options are now required to be ordered in requests, and ordering has to be validated by the server. Intent is to ease server processing of mandatory-to-implement options.

- o Swapped Option values for the mandatory- and optional-to-process Options, so we can have a simple lowest..highest ordering.

- o added MISORDERED_OPTIONS error.

- o re-ordered some error messages to cause MALFORMED_REQUEST (which is PCP’s most general error response) to be error 1, instead of buried in the middle of the error numbers.

- o clarified that, after successfully using a PCP server, that PCP server is declared to be non-responsive after 5 failed retransmissions.

- o tightened up text (which was inaccurate) about how long general PCP processing is to delay when receiving an error and if it should honor Opcode-specific error lifetime. Useful for MAP errors which have an error lifetime. (This all feels awkward to have only some errors with a lifetime.)

- o Added better discussion of multiple interfaces, including highlighting Wi-Fi+Ethernet. Added discussion of using IPv6 Privacy Addresses and RFC1918 as source addresses for PCP requests. This should finish the section on multi-interface issues.

- o added some text about why server might send SERVER_OVERLOADED, or might simply discard packets.

- o Dis-allow internal-port=0, which means we dis-allow using PCP as a DMZ-like function. Instead, ports have to be mapped individually.

- o Text describing server’s processing of PEER is tightened up.

- o Server’s processing of PEER now says it is implementation-specific if a PCP server continues to allow the mapping to exist after a PEER message. Client’s processing of PEER says that if client

wants mapping to continue to exist, client has to continue to send recurring PEER messages.

B.25. Changes from draft-ietf-pcp-base-04 to -05

- o tweaked PCP common header packet layout.
- o Re-added port=0 (all ports).
- o minimum size is 12 octets (missed that change in -04).
- o removed Lifetime from PCP common header.
- o for MAP error responses, the lifetime indicates how long the server wants the client to avoid retrying the request.
- o More clearly indicated which fields are filled by the server on success responses and error responses.
- o Removed UPnP interworking section from this document. It will appear in [I-D.ietf-pcp-upnp-igd-interworking].

B.26. Changes from draft-ietf-pcp-base-03 to -04

- o "Pinhole" and "PIN" changed to "mapping" and "MAP".
- o Reduced from four MAP Opcodes to two. This was done by implicitly using the address family of the PCP message itself.
- o New option THIRD_PARTY, to more carefully split out the case where a mapping is created to a different host within the home.
- o Integrated a lot of editorial changes from Stuart and Francis.
- o Removed nested NAT text into another document, including the IANA-registered IP addresses for the PCP server.
- o Removed suggestion (MAY) that PCP server reserve UDP when it maps TCP. Nobody seems to need that.
- o Clearly added NAT and NAPT, such as in residential NATs, as within scope for PCP.
- o HONOR_EXTERNAL_PORT renamed to PREFER_FAILURE
- o Added ’Lifetime’ field to the common PCP header, which replaces the functions of the ’temporary’ and ’permanent’ error types of the previous version.

- o Allow arbitrary Options to be included in PCP response, so that PCP server can indicate un-supported PCP Options. Satisfies PCP Issue #19

- o Reduced scope to only deal with mapping protocols that have port numbers.

- o Reduced scope to not support DMZ-style forwarding.

- o Clarified version negotiation.

B.27. Changes from draft-ietf-pcp-base-02 to -03

- o Adjusted abstract and introduction to make it clear PCP is intended to forward ports and intended to reduce application keepalives.

- o First bit in PCP common header is set. This allows DTLS and non-DTLS to be multiplexed on same port, should a future update to this specification add DTLS support.

- o Moved subscriber identity from common PCP section to MAP* section.

- o made clearer that PCP client can reduce mapping lifetime if it wishes.

- o Added discussion of host running a server, client, or symmetric client+server.

- o Introduced PEER4 and PEER6 Opcodes.

- o Removed REMOTE_PEER Option, as its function has been replaced by the new PEER Opcodes.

- o IANA assigned port 44323 to PCP.

- o Removed AMBIGUOUS error code, which is no longer needed.

B.28. Changes from draft-ietf-pcp-base-01 to -02

- o more error codes

- o PCP client source port number should be random

- o PCP message minimum 8 octets, maximum 1024 octets.

- o tweaked a lot of text in section 7.4, "Opcode-Specific Server Operation".

- o opening a mapping also allows ICMP messages associated with that mapping.

- o PREFER_FAILURE value changed to the mandatory-to-process range.

- o added text recommending applications that are crashing obtain short lifetimes, to avoid consuming subscriber's port quota.

B.29. Changes from draft-ietf-pcp-base-00 to -01

- o Significant document reorganization, primarily to split base PCP operation from Opcode operation.

- o packet format changed to move 'protocol' outside of PCP common header and into the MAP* opcodes

- o Renamed Informational Elements (IE) to Options.

- o Added REMOTE_PEER (for disambiguation with dynamic ports), REMOTE_PEER_FILTER (for simple packet filtering), and PREFER_FAILURE (to optimize UPnP IGDv1 interworking) options.

- o Is NAT or router behind B4 in scope?

- o PCP option MAY be included in a request, in which case it MUST appear in a response. It MUST NOT appear in a response if it was not in the request.

- o Result code most significant bit now indicates permanent/temporary error

- o PCP Options are split into mandatory-to-process ("P" bit), and into Specification Required and Private Use.

- o Epoch discussion simplified.

Authors' Addresses

Dan Wing (editor)
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, California 95134
USA

Email: dwing@cisco.com

Stuart Cheshire
Apple Inc.
1 Infinite Loop
Cupertino, California 95014
USA

Phone: +1 408 974 3207
Email: cheshire@apple.com

Mohamed Boucadair
France Telecom
Rennes, 35000
France

Email: mohamed.boucadair@orange.com

Reinaldo Penno
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, California 95134
USA

Email: repenno@cisco.com

Paul Selkirk
Internet Systems Consortium
950 Charter Street
Redwood City, California 94063
USA

Email: pselkirk@isc.org

PCP Working Group
Internet-Draft
Intended status: Standards Track
Expires: October 16, 2014

M. Boucadair
France Telecom
R. Penno
D. Wing
Cisco
April 14, 2014

DHCP Options for the Port Control Protocol (PCP)
draft-ietf-pcp-dhcp-13

Abstract

This document specifies DHCP (IPv4 and IPv6) options to configure hosts with Port Control Protocol (PCP) server IP addresses. The use of DHCPv4 or DHCPv6 depends on the PCP deployment scenarios. The set of deployment scenarios to which use of DHCPv4 or DHCPv6 apply are outside the scope of this document.

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 RFC 2119 [RFC2119].

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 October 16, 2014.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction 2
2. Terminology 3
3. DHCPv6 PCP Server Option 3
3.1. Format 3
3.2. DHCPv6 Client Behavior 4
4. DHCPv4 PCP Option 5
4.1. Format 5
4.2. DHCPv4 Client Behavior 6
5. DHCP Server Configuration Guidelines 6
6. Dual-Stack Hosts 8
7. Hosts with Multiple Interfaces 8
8. Security Considerations 8
9. IANA Considerations 8
9.1. DHCPv6 Option 8
9.2. DHCPv4 Option 8
10. Acknowledgements 9
11. References 9
11.1. Normative References 9
11.2. Informative References 10
Authors' Addresses 10

1. Introduction

This document defines DHCPv4 [RFC2131] and DHCPv6 [RFC3315] options that can be used to configure hosts with PCP server [RFC6887] IP addresses.

This specification assumes a PCP server is reachable with one or multiple IP addresses. As such, a list of IP addresses can be returned in the DHCP PCP server option.

This specification allows returning one or multiple lists of PCP server IP addresses. This is used as a hint to guide the PCP client when determining whether to send PCP requests to one or multiple PCP servers. Concretely, the PCP client needs an indication to decide whether entries need to be instantiated in all PCP servers (e.g.,

multi-homing, multiple PCP-controlled devices providing distinct services , etc.) or using one IP address from the list (e.g., redundancy group scenario, proxy-based model, etc.). Refer to [I-D.boucadair-pcp-deployment-cases] for a discussion on PCP deployment scenarios.

For guidelines on how a PCP client can use multiple IP addresses and multiple PCP servers, see [I-D.ietf-pcp-server-selection].

2. Terminology

This document makes use of the following terms:

- o PCP server denotes a functional element that receives and processes PCP requests from a PCP client. A PCP server can be co-located with or be separated from the function (e.g., NAT, Firewall) it controls. Refer to [RFC6887].
- o PCP client denotes a PCP software instance responsible for issuing PCP requests to a PCP server. Refer to [RFC6887].
- o DHCP refers to both DHCPv4 [RFC2131] and DHCPv6 [RFC3315].
- o DHCP client denotes a node that initiates requests to obtain configuration parameters from one or more DHCP servers.
- o DHCP server refers to a node that responds to requests from DHCP clients.

3. DHCPv6 PCP Server Option

3.1. Format

The DHCPv6 PCP server option can be used to configure a list of IPv6 addresses of a PCP server.

The format of this option is shown in Figure 1.

```
 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_V6_PCP_SERVER      |         Option-length         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                         ipv6-address                          |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                         ipv6-address                          |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             ...                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 1: DHCPv6 PCP server option

The fields of the option shown in Figure 1 are as follows:

- o Option-code: OPTION_V6_PCP_SERVER (TBA, see Section 9.1)
- o Option-length: Length of the ’PCP server IP Address(es)’ field in octets. MUST be a multiple of 16.
- o PCP server IPv6 Addresses: Includes one or more IPv6 addresses [RFC4291] of the PCP server to be used by the PCP client. Note, IPv4-mapped IPv6 addresses (Section 2.5.5.2 of [RFC4291]) are allowed to be included in this option.

To return more than one PCP server to the DHCPv6 client (as opposed to more than one address for a single PCP server), the DHCPv6 server returns multiple instances of OPTION_V6_PCP_SERVER.

3.2. DHCPv6 Client Behavior

To discover one or more PCP servers, the DHCPv6 client requests PCP server IP addresses by including OPTION_V6_PCP_SERVER in an Option Request Option (ORO), as described in Section 22.7 of [RFC3315].

The DHCPv6 client MUST be prepared to receive multiple instances of OPTION_V6_PCP_SERVER; each instance is to be treated as a separate PCP server.

If an IPv4-mapped IPv6 address is received in OPTION_V6_PCP_SERVER, it indicates that the PCP server has the corresponding IPv4 address.

Note: When presented with the IPv4-mapped prefix, current versions of Windows and Mac OS generate IPv4 packets, but will not send IPv6 packets [RFC6052]. Representing IPv4 addresses as IPv4-mapped IPv6 addresses follows the same logic as in section 5 of [RFC6887].

The DHCPv6 client MUST silently discard multicast and host loopback addresses [RFC6890] conveyed in OPTION_V6_PCP_SERVER.

4. DHCPv4 PCP Option

4.1. Format

The DHCPv4 PCP server option can be used to configure a list of IPv4 addresses of a PCP server. The format of this option is illustrated in Figure 2.

```
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Code         |     Length    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| List-Length   |   List of     |
+-+-+-+-+-+-+-+-+  PCP server   |
/              IPv4 Addresses   /
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+   ---
| List-Length   |   List of     |    |
+-+-+-+-+-+-+-+-+  PCP server   |    |
/              IPv4 Addresses   /    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+    |
.              ...               . optional
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+    |
| List-Length   |   List of     |    |
+-+-+-+-+-+-+-+-+  PCP server   |    |
/              IPv4 Addresses   /    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+   ---
```

Figure 2: DHCPv4 PCP server option

The description of the fields is as follows:

- o Code: OPTION_V4_PCP_SERVER (TBA, see Section 9.2);
- o Length: Length of all included data in octets. The minimum length is 5.
- o List-Length: Length of the "List of PCP server IPv4 Addresses" field in octets; MUST be a multiple of 4.

- o List of PCP server IPv4 Addresses: Contains one or more IPv4 addresses of the PCP server to be used by the PCP client. The format of this field is shown in Figure 3.
- o OPTION_V4_PCP_SERVER can include multiple lists of PCP server IPv4 addresses; each list is treated as a separate PCP server. When several lists of PCP server IPv4 addresses are to be included, "List-Length" and "PCP server IPv4 Addresses" fields are repeated.

```
0     8     16    24    32    40    48
+-----+-----+-----+-----+-----+-----+--
|  a1 |  a2 |  a3 |  a4 |  a1 |  a2 | ...
+-----+-----+-----+-----+-----+-----+--
     IPv4 Address 1          IPv4 Address 2 ...
```

This format assumes that an IPv4 address is encoded as a1.a2.a3.a4.

Figure 3: Format of the List of PCP server IPv4 Addresses

OPTION_V4_PCP_SERVER is a concatenation-requiring option. As such, the mechanism specified in [RFC3396] MUST be used if OPTION_V4_PCP_SERVER exceeds the maximum DHCPv4 option size of 255 octets.

4.2. DHCPv4 Client Behavior

To discover one or more PCP servers, the DHCPv4 client requests PCP server IP addresses by including OPTION_V4_PCP_SERVER in a Parameter Request List Option [RFC2132].

The DHCPv4 client MUST be prepared to receive multiple lists of PCP server IPv4 addresses in the same DHCPv4 PCP server option; each list is to be treated as a separate PCP server.

The DHCPv4 client MUST silently discard multicast and host loopback addresses [RFC6890] conveyed in OPTION_V4_PCP_SERVER.

5. DHCP Server Configuration Guidelines

DHCP servers supporting the DHCP PCP server option can be configured with a list of IP addresses of the PCP server(s). If multiple IP addresses are configured, the DHCP server MUST be explicitly configured whether all or some of these addresses refer to:

1. the same PCP server: the DHCP server returns multiple addresses in the same instance of the DHCP PCP server option.

2. distinct PCP servers: the DHCP server returns multiple lists of PCP server IP addresses to the requesting DHCP client (encoded as

multiple OPTION_V6_PCP_SERVER or in the same OPTION_V4_PCP_SERVER); each list is referring to a distinct PCP server. For example, multiple PCP servers may be configured to a PCP client in some deployment contexts such as multi-homing. It is out of scope of this document to enumerate all deployment scenarios that require multiple PCP servers to be returned.

Precisely how DHCP servers are configured to separate lists of IP addresses according to which PCP server they address is out of scope for this document. However, DHCP servers MUST NOT combine the IP addresses of multiple PCP servers and return them to the DHCP client as if they belong to a single PCP server, and DHCP servers MUST NOT separate the addresses of a single PCP server and return them as if they belonged to distinct PCP servers. For example, if an administrator configures the DHCP server by providing a Fully Qualified Domain Name (FQDN) for a PCP server, even if that FQDN resolves to multiple addresses, the DHCP server MUST deliver them within a single server address block.

DHCPv6 servers that implement this option and that can populate the option by resolving FQDNs will need a mechanism for indicating whether to query for A records or only AAAA records. When a query returns A records, the IP addresses in those records are returned in the DHCPv6 response as IPv4-mapped IPv6 addresses.

Discussion: The motivation for this design is to accommodate deployment cases where an IPv4 connectivity service is provided while only DHCPv6 is in use (e.g., an IPv4-only PCP server in a DS-Lite context [RFC6333]).

Since this option requires support for IPv4-mapped IPv6 addresses, a DHCPv6 server implementation will not be complete if it does not query for A records and represent any that are returned as IPv4-mapped IPv6 addresses in DHCPv6 responses. This behavior is neither required nor suggested for DHCPv6 options in general: it is specific to OPTION_V6_PCP_SERVER. The mechanism whereby DHCPv6 implementations provide this functionality is beyond the scope of this document.

For guidelines on providing context-specific configuration information (e.g., returning a regional-based configuration), and information on how a DHCP server might be configured with FQDNs that get resolved on demand, see [I-D.ietf-dhc-topo-conf].

6. Dual-Stack Hosts

A Dual-Stack host might receive PCP server option via both DHCPv4 and DHCPv6. For guidance on how a DHCP client can handle PCP server IP lists for the same network but obtained via different mechanisms, see [I-D.ietf-pcp-server-selection].

7. Hosts with Multiple Interfaces

A host may have multiple network interfaces (e.g, 3G, IEEE 802.11, etc.); each configured differently. Each PCP server learned MUST be associated with the interface via which it was learned.

Refer to [I-D.ietf-pcp-server-selection] and Section 8.4 of [RFC6887] for more discussion on multi-interface considerations.

8. Security Considerations

The security considerations in [RFC2131] and [RFC3315] are to be considered. PCP-related security considerations are discussed in [RFC6887].

The PCP Server option targets mainly the simple threat model (Section 18.1 of [RFC6887]). It is out of scope of this document to discuss potential implications of the use of this option in the advanced threat model (Section 18.2 of [RFC6887]).

9. IANA Considerations

9.1. DHCPv6 Option

IANA is requested to assign the following new DHCPv6 Option Code in the registry maintained in http://www.iana.org/assignments/dhcpv6-parameters:

Option Name	Value
OPTION_V6_PCP_SERVER	TBA

9.2. DHCPv4 Option

IANA is requested to assign the following new DHCPv4 Option Code in the registry maintained in http://www.iana.org/assignments/bootp-dhcp-parameters/:

Option Name	Value	Data length	Meaning
OPTION_V4_PCP_SERVER	TBA	Variable; the minimum length is 5.	Includes one or multiple lists of PCP server IP addresses; each list is treated as a separate PCP server.

10. Acknowledgements

Many thanks to C. Jacquenet, R. Maglione, D. Thaler, T. Mrugalski, T. Reddy, S. Cheshire, M. Wasserman, C. Holmberg, A. Farrel, S. Farrel, B. Haberman, and P. Resnick for their review and comments.

Special thanks to T. Lemon and B. Volz for the review and their effort to enhance this specification.

11. References

11.1. Normative References

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

[RFC2131] Droms, R., "Dynamic Host Configuration Protocol", RFC 2131, March 1997.

[RFC2132] Alexander, S. and R. Droms, "DHCP Options and BOOTP Vendor Extensions", RFC 2132, March 1997.

[RFC3315] Droms, R., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, July 2003.

[RFC3396] Lemon, T. and S. Cheshire, "Encoding Long Options in the Dynamic Host Configuration Protocol (DHCPv4)", RFC 3396, November 2002.

[RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, February 2006.

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

[RFC6890] Cotton, M., Vegoda, L., Bonica, R., and B. Haberman, "Special-Purpose IP Address Registries", BCP 153, RFC 6890, April 2013.

11.2. Informative References

[I-D.boucadair-pcp-deployment-cases]
Boucadair, M., "PCP Deployment Models", draft-boucadair-pcp-deployment-cases-01 (work in progress), December 2013.

[I-D.ietf-dhc-topo-conf]
Lemon, T. and T. Mrugalski, "Customizing DHCP Configuration on the Basis of Network Topology", draft-ietf-dhc-topo-conf-01 (work in progress), February 2014.

[I-D.ietf-pcp-server-selection]
Boucadair, M., Penno, R., Wing, D., Patil, P., and T. Reddy, "PCP Server Selection", draft-ietf-pcp-server-selection-02 (work in progress), January 2014.

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

[RFC6333] Durand, A., Droms, R., Woodyatt, J., and Y. Lee, "Dual-Stack Lite Broadband Deployments Following IPv4 Exhaustion", RFC 6333, August 2011.

Authors' Addresses

Mohamed Boucadair
France Telecom
Rennes 35000
France

Email: mohamed.boucadair@orange.com

Reinaldo Penno
Cisco
USA

Email: repenno@cisco.com

Dan Wing
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, California 95134
USA

Email: dwing@cisco.com

PCP WG
Internet-Draft
Intended status: Standards Track
Expires: November 25, 2013

R. Maglione
Cisco Systems
D. Cheng
Huawei Technologies
M. Boucadair
France Telecom

May 24, 2013

RADIUS Extensions for Port Control Protocol (PCP)
draft-maglione-pcp-radius-ext-08

Abstract

This document specifies a new Remote Authentication Dial In User Service (RADIUS) attribute to carry a Port Control Protocol (PCP) Server Names. This attribute can be configured on a RADIUS server so that the information can be conveyed to Network Access Server (NAS) via RADIUS protocol, and the co-located Dynamic Host Configuration Protocol (DHCP/DHCPv6) server can then populate the information to PCP client.

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 November 25, 2013.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents

carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction 2
2. Terminology 3
3. PCP Server Configuration using RADIUS and DHCPv4/DHCPv6 . . . 4
4. PCP-Server-Name RADIUS Attribute 7
5. Table of attributes 9
6. Security Considerations 9
7. IANA Considerations 9
8. Acknowledgments 9
9. References 9
9.1. Normative References 9
9.2. Informative References 10
Authors' Addresses 10

1. Introduction

Port Control Protocol (PCP) [RFC6887] provides a mechanism to control how incoming packets are forwarded by upstream devices such as NATs and firewalls. PCP is a client/server protocol where a PCP client may reside on a host, a Customer Premises Equipment (CPE), etc., which communicates with a PCP server that may reside anywhere in a network.

[RFC6887] defines a procedure for the PCP client to communicate with its PCP Server. The IP address of the PCP Server(s) can be configured to the PCP client; if not the PCP client assumes its default router as being its PCP server.

[I-D.ietf-pcp-dhcp] defines DHCPv6 and DHCPv4 options which are meant to be used by a PCP client to discover a PCP server name. However, provisioning for name of the PCP server is required on a DHCPv4/ DHCPv6 server before it can populate this information.

Auto-configuration on a DHCPv4/DHCPv6 is possible in a broadband network, where typically, user profile is maintained on a Remote Authentication Dial In User Service (RADIUS) server and RADIUS protocol [RFC2865] is used to convey user-related information to other network elements including a host and CPE. [RFC6911] describes a typical broadband network scenario in which the Network Access Server (NAS) acts as the access gateway for the users (hosts or CPEs) and the NAS embeds a DHCPv6 Server function that allows it to locally handle any DHCPv6 requests issued by the clients.

In such environment, PCP server’s name can be configured on a RADIUS server, which then passes the information to a NAS that co-locates with the DHCPv4/DHCPv6 server, which in turn populates the location of the PCP server.

This document defines a new RADIUS attribute that can be used to carry a PCP server name. As defined in [I-D.ietf-pcp-dhcp], a PCP Server Name can be a DNS name, IP literals strings, etc. This document is designed to allow for configuring PCP Server name which can be a DNS name, IP literals or any strings which may be passed to a local name resolution library on the PCP client side. Multiple occurrences of the PCP server name RADIUS attribute is supported.

The proposed RADIUS attribute is designed to accommodate various deployment contexts (e.g., dedicated option per IP connectivity context, single option for dual-stack access, etc.).

The approach described above is already used for providing the FQDN of the AFTR in the DS-Lite scenario [RFC6333] and the equivalent RADIUS attribute for the DS-Lite Tunnel Name is defined [RFC6519].

2. Terminology

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].

The following terms are defined in [RFC6887]:

- Port forwarding
- PCP
- PCP client
- PCP Server

The following term is defined in [I-D.ietf-pcp-dhcp]:

- PCP Server Name

3. PCP Server Configuration using RADIUS and DHCPv4/DHCPv6

Figure 1 illustrates an example of how RADIUS protocol works together with DHCPv6, to allow a host to learn automatically the name of a PCP server in case of a PPP session that carries IPv6 traffic.

The Network Access Server (NAS) operates as a client of RADIUS and co-locates with a DHCPv6 Server for DHCPv6. The NAS initially sends a RADIUS Access Request message to the RADIUS server, requesting authentication. Once the RADIUS server receives the request, it validates the sending client and if the request is approved, the RADIUS server replies with an Access Accept message including a list of attribute-value pairs that describe the parameters to be used for this session. This list MAY also contain the name of a PCP server. When the co-located DHCPv6 server receives a DHCPv6 message from a client containing the PCP Server Option, it SHALL use the name returned in the RADIUS attribute as defined in this memo to populate the DHCPv6 PCP Server option defined in [I-D.ietf-pcp-dhcp].

```
   PCP/DHCPv6                          NAS                      AAA
  Client                            DHCPv6 Server              Server
    |                                  |                         |
    |----PPP LCP Config Request------> |                         |
    |                                  |----Access-Request ---->|
    |                                  |                         |
    |                                  |<-Access-Accept---------|
    |                                  | (PCP-Server-Name)       |
    |<-----PPP LCP Config ACK  -----   |                         |
    |                                  |                         |
    |------ PPP IPv6CP Config Req ---->|                         |
    |                                  |                         |
    |<----- PPP IPv6CP Config ACK -----|                         |
    |                                  |                         |
    |-------  DHCPv6 Solicit  -------->|                         |
    |                                  |                         |
    |<-------DHCPv6 Advertisement------|                         |
    |  (PCP server Name DHCPv6 Option) |                         |
    |                                  |                         |
    |-------  DHCPv6 Request  -------->|                         |
    |  (PCP server Name DHCPv6 Option) |                         |
    |                                  |                         |
    |<-------- DHCPv6 Reply ---------  |                         |
    |  (PCP server Name DHCPv6 Option) |                         |

               DHCPv6                          RADIUS
```

Figure 1: RADIUS and DHCPv6 Message Flow for a PPP Session

The Figure 2 illustrates how the RADIUS protocol and DHCPv6 work together to accomplish PCP client configuration when DHCPv6 is used to provide connectivity to a requesting host.

The difference between this message flow and previous one is that in this scenario the interaction between NAS and AAA/ RADIUS Server is triggered by the DHCPv6 Solicit message received by the NAS from the DHCPv6 client, while in case of a PPP Session the trigger is the PPP LCP Config Request message received by the NAS.

```
 PCP/DHCPv6                            NAS                     AAA
  Client                            DHCPv6 Server             Server
    |                                  |                        |
    |------ DHCPv6 Solicit --------->  |                        |
    |                                  |                        |
    |                                  |----Access-Request ---->|
    |                                  |                        |
    |                                  |<-Access-Accept---------|
    |                                  | (PCP-Server-Name)      |
    |                                  |                        |
    |<-------DHCPv6 Advertisement------|                        |
    |  (PCP Server Name DHCPv6 Option) |                        |
    |                                  |                        |
    |-------  DHCPv6 Request  -------->|                        |
    |  (PCP Server Name DHCPv6 Option) |                        |
    |                                  |                        |
    | <-------- DHCPv6 Reply --------- |                        |
    | (PCP Server Name DHCPv6 Option)  |                        |

                DHCPv6                          RADIUS
```

Figure 2: RADIUS and DHCPv6 Message Flow for an IP Session

In the scenario depicted in Figure 2 the Access-Request packet SHOULD contain a Service-Type attribute (6) with the value Authorize Only (17); thus, according to [RFC5080], the Access-Request packet MUST contain a State attribute that it obtains from the previous authentication process.

In both scenaiors mentioned above, Message-Authenticator (type 80) according to [RFC2869] SHOULD be used to protect both Access-Request and Access-Accept Messages.

In case that the PCP server name is re-configured, the RADIUS server must send a RADIUS CoA message [RFC5176] that carries the RADIUS PCP server name attribute to the NAS, which once accepts and sends back a RADIUS CoA ACK message, the new PCP server name replaces the original one and is then re-propulated by the DHCPv6 server.

A similar message flow also applies to the IPv4 scenario when DHCPv4 is used to provide connectivity to the user (Figure 3).

```
 PCP/DHCPv4                              NAS                     AAA
  Client                            DHCPv4 Server              Server
    |                                 |                         |
    |-------- DHCP Discovery --------> |                         |
    |                                 |                         |
    |                                 |----Access-Request ---->|
    |                                 |                         |
    |                                 |<-Access-Accept---------|
    |                                 | (PCP-Server-Name)       |
    |                                 |                         |
    |<--------- DHCP Offer ------------|                         |
    |    (PCP server Name Sub-Option)  |                         |
    |                                 |                         |
    |---------  DHCP Request  -------->|                         |
    |   (PCP server Name Sub-Option)   |                         |
    |                                 |                         |
    | <--------- DHCP Ack -------------|                         |
    |    (PCP server Name Sub-Option)  |                         |

                DHCPv4                        RADIUS
```

Figure 3: RADIUS and DHCPv4 Message Flow for an IP Session

After receiving the PCP server name in the initial Access-Accept the NAS MUST store the received PCP Server Name locally. When the PCP Client sends a DHCPv4 message to request an extension of the lifetimes for the assigned address or prefix, the NAS does not have to initiate a new Access-Request towards the AAA server to request the PCP server name. The NAS retrieves the previously stored PCP Server name and uses it in its reply.

If the DHCPv4 server to which the DHCP Renew message was sent at time T1 has not responded, the DHCPv4 client initiates a Rebind/Reply exchange with any available server. In this scenario the NAS MUST initiate a new Access-Request towards the AAA server, after the co-located DHCPv4 server receives the DHCP message. The NAS MAY include the PCP Server Name attribute in its Access-Request.

If the NAS does not receive the PCP server name attribute in the Access-Accept it MAY fallback to a pre-configured default tunnel name, if any. If the NAS does not have any pre-configured default tunnel name or if the NAS receives an Access-Reject, the PCP client can not be configured by the NAS.

The handling when the PCP server name is re-configured on the RADIUS server is similar to that in IPv6 case, i.e., the new PCP server name is conveyed to the NAS in a RADIUS CoA message, which if accepted, the new PCP server name replaces the original one and is then re-propulated by the DHCPv4 server.

The scenario with PPP Session and IPv4 only connectivity does not require DHCPv4: the whole configuration of the client is performed by PPP. This case is out of scope of this document because in order to complete the configuration of the PCP client a new PPP IPCP option would be required.

4. PCP-Server-Name RADIUS Attribute

A new RADIUS attribute, called PCP-Server-Name, along with its format is defined below.

The PCP-Server-Name attribute contains a name that refers to a PCP server the client requests to establish a connection to for PCP related service. The NAS shall use the name(s) returned in the RADIUS PCP-Server-Name attribute instance(s) to populate the PCP Server Name DHCP Sub-Option in IPv4 addressing context, or the PCP Server Name DHCPv6 Option in IPv6 addressing context, as determined by the DHCP server [I-D.ietf-pcp-dhcp]. The same or distinct PCP Server Names MAY be configured; it is out of scope of this document to elaborate on this point. Nevertheless, the PCP-Server-Name attribute conveys an indication for the deployment context.

The PCP-Server-Name attribute MAY appear in an Access-Accept packet. This attribute MAY be used in Access-Request packets as a hint to the RADIUS server; for example if the NAS is pre-configured with a default PCP server name, this name MAY be inserted in the attribute. The RADIUS server MAY ignore the hint sent by the NAS and it MAY assign a different PCP Server name. If the NAS includes the PCP Server Name attribute, but the AAA server does not recognize it, this attribute MUST be ignored by the AAA Server. If the NAS does not receive PCP Server Name attribute in the Access-Accept it MAY fallback to a pre-configured default PCP server name, if any. If the NAS is pre-provisioned with a default PCP server name and the PCP server name received in Access-Accept is different from the configured default, then the PCP server name received in the Access-Accept message MUST be used for the session.

The PCP server Name RADIUS attribute MAY be present in Accounting-Request records where the Acct-Status-Type is set to Start, Stop or Interim-Update.

The PCP server name RADIUS attribute MAY be present in an CoA-Request packet, when the PCP server name is re-configured.

The PCP Server Name RADIUS attribute MAY appear more than once in a message.

A summary of the PCP-Server-Name RADIUS attribute format is shown below. The fields are transmitted from left to right.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |    Length     |             Context           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      PCP-Server-Name  ....
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The description of the fields is as follows:

Type:

TBA1 for PCP-Server-Name.

Length:

This field indicates the total length in octets of this attribute including the Type, the Length fields.

Context:

This field indicates the IP connectivity context:

0: Dual-Stack. The same option is provided for both DHCPv4 and DHCPv6 requesting hosts.
1: This option is provided for DHCPv4 requesting hosts.
2: This option is provided for DHCPv6 requesting hosts.

PCP-Server-Name:

Includes a PCP Server Name. As defined in , PCP Server Name is a UTF-8 [RFC3629] string that can be passed to getaddrinfo(), such as a DNS name, address literals, etc. The name MUST NOT contain spaces or nulls.

This attribute is type of complex [RFC6158].

5. Table of attributes

The following table provides a guide to which attributes may be found in which kinds of packets, and in what quantity.

Request	Accept	Reject	Challenge	Accounting Request	#	Attribute
0+	0+	0	0	0+	TBA1	PCP-Server-Name
0-1	0-1	0	0	0-1	6	Service-Type
0-1	0-1	0-1	0-1	0-1	80	Message-Authenticator

The following table defines the meaning of the above table entries.

0 This attribute MUST NOT be present in packet.
0+ Zero or more instances of this attribute MAY be present in packet.
0-1 Zero or one instance of this attribute MAY be present in packet.

6. Security Considerations

This document has no additional security considerations beyond those already identified in [RFC2865].

7. IANA Considerations

This document requests the allocation of a new Radius attribute types from the IANA registry "Radius Attribute Types" located at http://www.iana.org/assignments/radius-types:

PCP-Server-Name - TBA1

8. Acknowledgments

The authors would like to thank Mario Ullio, Alan Dekok, Sheng Jiang and Tassos Chatzithomaoglou for their valuable comments and assistance.

9. References

9.1. Normative References

[I-D.ietf-pcp-dhcp]
Boucadair, M., Penno, R., and D. Wing, "DHCP Options for the Port Control Protocol (PCP)", draft-ietf-pcp-dhcp-07 (work in progress), March 2013.

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

[RFC2865] Rigney, C., Willens, S., Rubens, A., and W. Simpson, "Remote Authentication Dial In User Service (RADIUS)", RFC 2865, June 2000.

[RFC3629] Yergeau, F., "UTF-8, a transformation format of ISO 10646", STD 63, RFC 3629, November 2003.

[RFC5080] Nelson, D. and A. DeKok, "Common Remote Authentication Dial In User Service (RADIUS) Implementation Issues and Suggested Fixes", RFC 5080, December 2007.

[RFC6158] DeKok, A. and G. Weber, "RADIUS Design Guidelines", BCP 158, RFC 6158, March 2011.

[RFC6519] Maglione, R. and A. Durand, "RADIUS Extensions for Dual-Stack Lite", RFC 6519, February 2012.

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

9.2. Informative References

[RFC2869] Rigney, C., Willats, W., and P. Calhoun, "RADIUS Extensions", RFC 2869, June 2000.

[RFC5176] Chiba, M., Dommety, G., Eklund, M., Mitton, D., and B. Aboba, "Dynamic Authorization Extensions to Remote Authentication Dial In User Service (RADIUS)", RFC 5176, January 2008.

[RFC6333] Durand, A., Droms, R., Woodyatt, J., and Y. Lee, "Dual-Stack Lite Broadband Deployments Following IPv4 Exhaustion", RFC 6333, August 2011.

[RFC6911] Dec, W., Sarikaya, B., Zorn, G., Miles, D., and B. Lourdelet, "RADIUS Attributes for IPv6 Access Networks", RFC 6911, April 2013.

Authors' Addresses

Roberta Maglione
Cisco Systems
181 Bay Street
Toronto, ON M5J 2T3
Canada

Email: 'robmgl@cisco.com'

Dean Cheng
Huawei Technologies
2330 Central Expressway
Santa Clara, CA 95050
USA

Phone: +1 408 330 4754
Email: dean.cheng@huawei.com

Mohamed Boucadair
France Telecom
Rennes 35000
France

Email: mohamed.boucadair@orange.com

Network Working Group
Internet-Draft
Updates: 5191 (if approved)
Intended status: Standards Track
Expires: January 04, 2014

Y. Ohba
Y. Tanaka
Toshiba
S. Das
ACS
A. Yegin
Samsung
T. Tsou
Huawei

July 03, 2013

Provisioning Message Authentication Key for PCP using PANA (Side-by-Side Approach)
draft-ohba-pcp-pana-04

Abstract

This document specifies a mechanism for provisioning PCP (Port Control Protocol) message authentication key using PANA (Protocol for carrying Authentication for Network Access).

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 January 04, 2014.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents

carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction 2
1.1. Specification of Requirements 2
2. Establishing a PCP SA 3
3. Authentication Capablity Discovery 5
4. Security Considerations 6
5. IANA Considerations 6
6. Acknowledgments 6
7. Normative References 6
Appendix A. Change History 7
Authors' Addresses 8

1. Introduction

PCP (Port Control Protocol) [I-D.ietf-pcp-base] is used for an IPv6 or IPv4 host to control how incoming IPv6 or IPv4 packets are translated and forwarded by a network address translator (NAT) or by a simple firewall. It also allows a host to optimize its outgoing NAT keepalive messages.

In order to provide integrity protection for PCP messages, a message authentication mechanism for PCP is defined in [I-D.ietf-pcp-authentication]. Three components are defined in [I-D.ietf-pcp-authentication]: (1) PCP options for providing per-packet origin authentication, integrity and replay protection, (2) PCP Security Association (SA) for generating the aforementioned options, and (3) PCP options for generating PCP SA from execution of EAP authentication.

The third component seems to define a new EAP lower-layer within PCP. In this document, PANA (Protocol for carrying Authentication for Network Access) [RFC5191] is proposed instead of defining a new EAP lower-layer. This draft along with other two components described in [I-D.ietf-pcp-authentication] provides a complete solution which otherwise will duplicate the work of transporting EAP over UDP. The proposed solution can run over a single PCP port.

1.1. Specification of Requirements

In this document, several words are used to signify the requirements of the specification. These words are often capitalized. 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 RFC 2119 [RFC2119].

2. Establishing a PCP SA

A PCP client should know the authentication capability of the PCP server before deciding to use PANA with it. PCP client can obtain this information either via an out-of band scheme (e.g., manual configuration, DHCP), or via an in-band scheme (e.g., trial-and-error, PCP ANNOUNCE Opcode). In trial-and-error scheme the PCP client tests the PCP server by sending its first request without any authentication. If the PCP server returns AUTHENTICATION_REQUIRED error message, then the PCP client concludes that the PCP server is mandating use of authentication. Otherwise the PCP client concludes that the PCP server is allowing unauthenticated PCP. See Section 3 for the details of ANNOUNCE-based discovery.

A PaC (PANA Client) on a PCP client node initiates PANA authentication over the PCP port number (To be assigned) prior to sending an authenticated PCP message. The initiation may be requested by the PCP client. We assume that a PAA (PANA Authentication Agent) is implemented on each PCP server that supports authenticated PCP messages. Therefore, the PCP server’s IP address is used as the address of the PAA. The PANA authentication for establishing a PCP SA is dedicated to the PCP usage only.

In order to distinguish PANA and PCP messages that are multiplexed over the PCP port number (To be assigned), bit 0 of Reserved field of PANA header is used and whose value is 1. In PCP, the corresponding bit is part of Version field and whose value is 0, as shown in Figure 1. For this scheme to work, PCP Version values less than 128 MUST be used.

```
 0
 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+
|    Reserved     ...
+-+-+-+-+-+-+-+-+
    The first 8 bits of PANA header (bit 0 value is 1)

 0
 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+-
|    Version    | ...
+-+-+-+-+-+-+-+-+-
     The first 8 bits of PCP header (bit 0 value is 0)
```

Figure 1: The First 8 bits of PANA and PCP Headers

When a PANA message is carried over the PCP port number (To be assigned), the sender MUST set bit 0 of Reserved field. Other Reserved bits and bit 0 when used over port numbers other than the PCP port number (To be assigned) are still governed by [RFC5191].

Upon successful PANA authentication, the message authentication key for PCP message is derived from the EAP MSK as follows:

PCP_AUTH_KEY = prf+(MSK, "IETF PCP" | SID | KID)

where where | denotes concatenation.

- o The prf+ function is defined in IKEv2 [RFC5996]. The pseudo-random function to be used for the prf+ function is negotiated using PRF-Algorithm AVP in the initial PANA-Auth-Request and PANA-Auth-Answer exchange with 'S' (Start) bit set, as defined in [RFC5191].

- o "IETF PCP" is the ASCII code representation of the non-NULL terminated string (excluding the double quotes around it).

- o SID is a four-octet PANA Session Identifier [RFC5191].

- o KID is the content of the Key-ID AVP [RFC5191] associated with the MSK.

The same integrity algorithm used for the PANA session MUST be used for PCP message authentication.

The PCP_AUTH_KEY and its associated parameters (i.e., the IP addresses of the PCP client and PCP server, PANA Session ID, Key ID, message authentication algorithm and lifetime) are passed from the

PAA application to the PCP server application on the same PCP server device, and also passed from the PaC application to the PCP client application on the same PCP client node, using an API. The API can be implementation-specific, and therefore is not specified in this document. The PANA Session ID and Key ID are used in the corresponding fields (Session ID, Key ID) of the Authentication Tag Option.

Once a PCP SA is established, any PCP message that does not contain a valid Authentication Tag and a fresh Nonce under the current PCP SA MUST be silently discarded.

The PCP SA MUST be immediately deleted when the corresponding PANA SA is deleted. The PCP SA SHALL remain as long as the corresponding PANA SA exists.

If the PCP server that requires authenticated PCP message receives an unauthenticated PCP request, it returns an "AUTHENTICATION_REQUIRED" result code.

If a PCP SA needs to be updated, the PCP client or the PCP server SHALL initiate PANA re-authentication phase. If a PCP SA needs to be re-established after expiration or loss of the SA for an existing PCP mapping state, the PCP client or the PCP server SHALL initiate PANA authentication and authorization phase.

3. Authentication Capablity Discovery

A PCP client supporting PCP authentication MAY send an ANNOUNCE request with an AUTH_CAPABILITY option prior to initiating PANA in order to know whether a PCP server supports PCP authentication. A PCP server supporting PCP authentication SHALL return an ANNOUNCE response with "SUCCESS" result code and an AUTH_CAPABILITY option.

The AUTH_CAPABILITY Option is formatted in Figure 2.

```
 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=TBA|  Reserved     |        Option Length=0        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2: AUTH_CAPABILITY Option Format

The fields are described below:

Option Name: AUTH_CAPABILITY
Number: To be assigned by IANA

Purpose: To indicate the sender’s authentication capability
Valid for Opcodes: ANNOUNCE
Length: 0
May appear in: requests, responses
Maximum occurrences: 1

4. Security Considerations

The key provisioning mechanism described in this document provides a cryptographic binding between a PANA session and a PCP SA based on using the PANA session identifier and key identifier in the PCP_AUTH_KEY derivation function.

For EAP channel binding [RFC6677], it is required for a PAA to distinguish whether PANA authentication is conducted for network access authentication or PCP authentication. Such a distinction can be made using the assigned port number over which the PANA authentication is conducted, namely, the PANA authentication is conducted for PCP authentication when the port number is the PCP port number (to be assigned), and it is for network access authentication when the port number is the PANA port number (716). How the corresponding information is conveyed from the PAA to the authentication server is outside the scope of this document.

5. IANA Considerations

A new result code for "AUTHENTICATION_REQUIRED" needs to be allocated. The usage of the "AUTHENTICATION_REQUIRED" result code is described in Section 2.

A new PCP Option for AUTH_CAPABILITY needs to be allocated. The usage of AUTH_CAPABILITY Option is described in Section 3.

6. Acknowledgments

Authors would like to acknowledge Dave Thaler for his suggestion on the use of ANNOUNCE Opcode for capability discovery, and Richard Martija, Pedro Moreno Sanchez and Rafa Marin-Lopez for fully implementing the mechanism described in this document.

7. Normative References

[I-D.ietf-pcp-authentication]
Wasserman, M., Hartman, S., and D. Zhang, "Port Control Protocol (PCP) Authentication Mechanism", draft-ietf-pcp-authentication-01 (work in progress), October 2012.

[I-D.ietf-pcp-base]
Wing, D., Cheshire, S., Boucadair, M., Penno, R., and P. Selkirk, "Port Control Protocol (PCP)", draft-ietf-pcp-base-29 (work in progress), November 2012.

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

[RFC5191] Forsberg, D., Ohba, Y., Patil, B., Tschofenig, H., and A. Yegin, "Protocol for Carrying Authentication for Network Access (PANA)", RFC 5191, May 2008.

[RFC5996] Kaufman, C., Hoffman, P., Nir, Y., and P. Eronen, "Internet Key Exchange Protocol Version 2 (IKEv2)", RFC 5996, September 2010.

[RFC6677] Hartman, S., Clancy, T., and K. Hoeper, "Channel-Binding Support for Extensible Authentication Protocol (EAP) Methods", RFC 6677, July 2012.

Appendix A. Change History

Changes from -00 to -01 :

- o Added Alper to authors.
- o Changed to use demultiplexing approach from seperate key management.
- o Removed PCP server id from key derivation algorithm.
- o Added EAP channel binding discussion in Security Considerations section.

Changes from -01 to -02 :

- o Added Editor’s Note in Section 2.

Changes from -02 to -03 :

- o Changed document title
- o Added Tina to authors.
- o Used Bit 0 instead of Bits 5-6-7 to consider PCP Version 0 used by NAT-PCP.
- o Added ANNOUNCE-based authentication capability discovery.

o Moved RFC 2119 to Normative Reference.

Changes from -03 to -04 :

o Added text for SA revnew and re-establishment.

Authors' Addresses

Yoshihiro Ohba
Toshiba Corporate Research and Development Center
1 Komukai-Toshiba-cho
Saiwai-ku, Kawasaki, Kanagawa 212-8582
Japan

Phone: +81 44 549 2127
Email: yoshihiro.ohba@toshiba.co.jp

Yasuyuki Tanaka
Toshiba Corporate Research and Development Center
1 Komukai-Toshiba-cho
Saiwai-ku, Kawasaki, Kanagawa 212-8582
Japan

Phone: +81 44 549 2127
Email: yatch@isl.rdc.toshiba.co.jp

Subir Das
Applied Communication Sciences
1 Telcordia Drive
Piscataway, NJ 08854
USA

Email: sdas@appcomsci.com

Alper Yegin
Samsung
Istanbul
Turkey

Email: alper.yegin@yegin.org

Tina Tsou
Huawei Technologies (USA)
2330 Central Expressway
Santa Clara, CA 95050
USA

Email: Tina.Tsou.Zouting@huawei.com
URI: http://tinatsou.weebly.com/contact.html

Network Working Group
Internet-Draft
Updates: 5191 (if approved)
Intended status: Standards Track
Expires: January 04, 2014

Y. Ohba
Toshiba
A. Yegin
Samsung
S. Das
ACS

July 03, 2013

Provisioning Message Authentication Key for PCP using PANA (Encapsulation Approach)

draft-ohba-pcp-pana-encap-01

Abstract

This document specifies a mechanism for provisioning PCP (Port Control Protocol) message authentication key by encapsulating PANA (Protocol for carrying Authentication for Network Access) in PCP.

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 January 04, 2014.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of

the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction 2
1.1. Specification of Requirements 2
2. Establishing a PCP SA 3
3. Security Considerations 4
4. IANA Considerations 4
5. Acknowledgments 5
6. Normative References 5
Authors' Addresses 5

1. Introduction

PCP (Port Control Protocol) [I-D.ietf-pcp-base] is used for an IPv6 or IPv4 host to control how incoming IPv6 or IPv4 packets are translated and forwarded by a network address translator (NAT) or by a simple firewall. It also allows a host to optimize its outgoing NAT keepalive messages.

In order to provide integrity protection for PCP messages, a message authentication mechanism for PCP is defined in [I-D.ietf-pcp-authentication]. Three components are defined in [I-D.ietf-pcp-authentication]: (1) PCP options for providing per-packet origin authentication, integrity and replay protection, (2) PCP Security Association (SA) for generating the aforementioned options, and (3) PCP options for generating PCP SA from execution of EAP authentication.

The third component seems to define a new EAP lower-layer within PCP. In this document, PANA (Protocol for carrying Authentication for Network Access) [RFC5191] is proposed instead of defining a new EAP lower-layer. This draft along with other two components described in [I-D.ietf-pcp-authentication] provides a complete solution which otherwise will duplicate the work of transporting EAP over UDP. The proposed solution can run over a single PCP port.

1.1. Specification of Requirements

In this document, several words are used to signify the requirements of the specification. These words are often capitalized. 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 RFC 2119 [RFC2119].

2. Establishing a PCP SA

A PaC (PANA Client) on the PCP client node initiates PANA authentication over the PCP port number (To be assigned for PCP [I-D.ietf-pcp-base]) prior to sending an authenticated PCP message. The initiation is requested by the PCP client. We assume that a PAA (PANA Authentication Agent) is implemented on each PCP server that supports authenticated PCP messages. Therefore, the PCP server's IP address is used as the address of the PAA. The PANA authentication for establishing a PCP SA is dedicated to the PCP usage only.

PANA authentication for establishing a PCP SA is conducted using PCP AUTH Opcode. An AUTH request or response carries a PANA PDU (PANA header and payload) in its Opcode-specific information [I-D.ietf-pcp-base]. Any PANA message sent by the PCP client is carried in an AUTH request, and any PANA message sent by the PCP server is carried in an AUTH response. The PCP retransmission mechanism SHOULD NOT be used for Auth Opcode to avoid double-layer retransmission.

If the PCP server does not support AUTH Opcode but it receives an AUTH request, it returns an "UNSUPP_OPCODE" result code.

If the PCP server that requires authenticated PCP message receives an unauthenticated PCP request other than an AUTH request, it returns an "AUTHENTICATION_REQUIRED" result code.

Upon successful PANA authentication, the message authentication key for PCP message is derived from the EAP MSK as follows:

PCP_AUTH_KEY = prf+(MSK, "IETF PCP" | SID | KID)

where where | denotes concatenation.

- o The prf+ function is defined in IKEv2 [RFC5996]. The pseudo-random function to be used for the prf+ function is negotiated using PRF-Algorithm AVP in the initial PANA-Auth-Request and PANA-Auth-Answer exchange with 'S' (Start) bit set, as defined in [RFC5191].

- o "IETF PCP" is the ASCII code representation of the non-NULL terminated string (excluding the double quotes around it).

- o SID is a four-octet PANA Session Identifier [RFC5191].

- o KID is the content of the Key-ID AVP [RFC5191] associated with the MSK.

The same integrity algorithm used for the PANA session MUST be used for PCP message authentication.

The PCP_AUTH_KEY and its associated parameters (i.e., the IP addresses of the PCP client and PCP server, PANA Session ID, Key ID, message authentication algorithm and lifetime) are passed from the PAA application to the PCP server application on the same PCP server device, and also passed from the PaC application to the PCP client application on the same PCP client node, using an API. The API can be implementation-specific, and therefore is not specified in this document. The PANA Session ID and Key ID are used in the corresponding fields (Session ID, Key ID) of the Authentication Tag Option.

Once a PCP SA is established, any PCP message that does not contain a valid Authentication Tag and a fresh Nonce under the current PCP SA MUST be silently discarded.

The PCP SA MUST be immediately deleted when the corresponding PANA SA is deleted. The PCP SA SHALL remain as long as the corresponding PANA SA exists.

If a PCP SA needs to be updated, the PCP client or the PCP server SHALL initiate PANA re-authentication phase. If a PCP SA needs to be re-established after expiration or loss of the SA for an existing PCP mapping state, the PCP client or the PCP server SHALL initiate PANA authentication and authorization phase.

3. Security Considerations

The key provisioning mechanism described in this document provides a cryptographic binding between a PANA session and a PCP SA based on using the PANA session identifier and key identifier in the PCP_AUTH_KEY derivation function.

For EAP channel binding [RFC6677], it is required for a PAA to distinguish whether PANA authentication is conducted for network access authentication or PCP authentication. Such a distinction can be made using the assigned port number over which the PANA authentication is conducted, namely, the PANA authentication is conducted for PCP authentication when the port number is the PCP port number (to be assigned), and it is for network access authentication when the port number is the PANA port number (716). How the corresponding information is conveyed from the PAA to the authentication server is outside the scope of this document.

4. IANA Considerations

A new PCP Opcode for AUTH needs to be allocated. The usage of the AUTH Opcode is described in Section 2.

A new result code for "AUTHENTICATION_REQUIRED" needs to be allocated. The usage of the "AUTHENTICATION_REQUIRED" result code is described in Section 2.

5. Acknowledgments

TBD.

6. Normative References

[I-D.ietf-pcp-authentication]
Wasserman, M., Hartman, S., and D. Zhang, "Port Control Protocol (PCP) Authentication Mechanism", draft-ietf-pcp-authentication-01 (work in progress), October 2012.

[I-D.ietf-pcp-base]
Wing, D., Cheshire, S., Boucadair, M., Penno, R., and P. Selkirk, "Port Control Protocol (PCP)", draft-ietf-pcp-base-29 (work in progress), November 2012.

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

[RFC5191] Forsberg, D., Ohba, Y., Patil, B., Tschofenig, H., and A. Yegin, "Protocol for Carrying Authentication for Network Access (PANA)", RFC 5191, May 2008.

[RFC5996] Kaufman, C., Hoffman, P., Nir, Y., and P. Eronen, "Internet Key Exchange Protocol Version 2 (IKEv2)", RFC 5996, September 2010.

[RFC6677] Hartman, S., Clancy, T., and K. Hoeper, "Channel-Binding Support for Extensible Authentication Protocol (EAP) Methods", RFC 6677, July 2012.

Authors' Addresses

Yoshihiro Ohba
Toshiba Corporate Research and Development Center
1 Komukai-Toshiba-cho
Saiwai-ku, Kawasaki, Kanagawa 212-8582
Japan

Phone: +81 44 549 2127
Email: yoshihiro.ohba@toshiba.co.jp

Alper Yegin
Samsung
Istanbul
Turkey

Email: alper.yegin@yegin.org

Subir Das
Applied Communication Sciences
1 Telcordia Drive
Piscataway, NJ 08854
USA

Email: sdas@appcomsci.com

PCP Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 14, 2013

P. Patil
T. Reddy
R. Penno
D. Wing
Cisco

October 11, 2012

Using PCP to control NAT and Firewalls in Multihoming
draft-patil-pcp-multihoming-00

Abstract

This note describes how Port Control Protocol (PCP) can be used to control NATs and Firewalls in multihoming deployments.

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 April 14, 2013.

Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction 3
2. Problem Statement 3
3. IPv6 Multihoming 4
4. IPv4 Multihoming 4
5. Other Multihoming use cases 5
5.1. IPv6 Network-Managed Firewall 6
5.2. IPv4 Policy based Routing 7
6. Multiple interfaces and Servers 7
7. Security Considerations 8
8. IANA Considerations 8
9. References 8
9.1. Normative References 8
9.2. Informative References 9
Authors' Addresses 9

1. Introduction

A host can use the Port Control Protocol (PCP) to flexibly manage the IP address and port mapping information on Network Address Translators (NATs) or firewalls, to facilitate communications with remote hosts. In a multihomed network, there may be multiple PCP servers providing Firewall or prefix translation functions to hosts in the network.

This document covers PCP related considerations in IPv4 and IPv6 multihomed networks.

2. Problem Statement

The main problem of a PCP multihoming situation can be succintly described as 'one client, multiple servers'. PCP-base [I-D.ietf-pcp-base] does not address how a PCP Client should behave in a situation when it discovers multiple PCP Servers and therefore many questions are open to standardization. For example, if multiple PCP Servers are discovered through the same interface, should the client send PCP requests be sent to all of them? Are there significant differences between a multihoming and high-availability scenarios? If yes, how can a PCP Client determine one versus the other. These are just a few questions related to the problem.

In this document we make the following simplifying assumption:

- o Whenever a PCP Client discovers multiple PCP Servers, it will send requests to all of them in parallel as described in [I-D.boucadair-pcp-server-selection].

- o There is no requirement that multiple PCP Servers have the same capabilities.

- o PCP Requests to different servers are independent, meaning that the result of a PCP request to one server does not influence another.

- o If PCP Servers provides NAT, it is out of scope how the client manages ports across PCP Servers. For example, whether PCP Client requires all external ports to be the same or whether there are ports available at all.

In all scenarios below PCP client has a single interface unless explicitly noted otherwise.

3. IPv6 Multihoming

In an IPv6 multihomed network, two or more routers co-located with firewalls are present on a single link shared with the host(s). Each router is in turn connected to a different service provider network and the host in this environment would be offered multiple prefixes and advertised multiple DNS/NTP servers. Consider a scenario in which firewalls within an IPv6 multihoming environment also implement a PCP Server. PCP client learns of the available PCP servers by using DHCP [I-D.ietf-pcp-dhcp] or any other PCP server discovery technique defined in future specifications. The PCP client will send PCP requests in parallel to each of the PCP Servers as described in [I-D.boucadair-pcp-server-selection].

```
                        ==================
                        |    Internet    |
                        ==================
                           |          |
                           |          |
                       +----+-+      +-+----+
                       | ISP1 |      | ISP2 |
                       +----+-+      +-+----+      ISP Network
                           |          |
    ..........................................................
                           |          |
                           |          |        Subscriber Network
                   +-------+---+ +----+------+
                   | rtr1 with | | rtr2 with |
                   |   FW1     | |    FW2    |
                   +-------+---+ +----+------+
                           |          |
                           |          |
                           |          |
                    -----+-+-----+------
                           |
                         +-+-----+
                         | Hosts |
                         +-------+
```

Figure 1: IPv6 Multihoming

4. IPv4 Multihoming

In an IPv4 multihomed network, the gateway router is connected to different service provider networks. The host is connected to the gateway router and is given a private IPv4 address oblivious to the fact that there are multiple service providers. The Gateway router

will be configured with multiple PCP proxy servers, each corresponding to an upstream PCP server. Each PCP Server is announced independently since it is within a different ISP. PCP client can learn these multiple PCP proxy addresses using DHCP or any other PCP server discovery technique. The PCP client, by sending PCP requests in parallel to both the PCP proxies, will learn the external IP addresses and ports allocated by each of the upstream PCP servers. The Gateway router which implements a PCP Proxy [I-D.bpw-pcp-proxy] , creates local NAT state, modifies the PCP request and forwards it to the PCP server. The incoming PCP response will be updated by the PCP Proxy and forwarded to the PCP client.

```
                      ====================
                      |    Internet       |
                      =====================
                         |              |
                         |              |
                    +----+--------+   +-+------------+
                    | ISP1        |   | ISP2         |
                    | PCP ServerA |   | PCP ServerB  |
                    +----+--------+   +-+------------+ ISP Network
                         |              |
                         |              |
        ..............................................................
                         |              |
                         | Port1        | Port2    Subscriber Network
                         |              |
                    +----+-----------------+
                    |   NAPT & PCP Proxy   |
                    |       GW Router      |
                    +----+-----------------+
                         |
                         |
                         |
                    -----+-+-----+------
                         |
                       +-+-----+
                       | Hosts |  (private address space)
                       +-------+
```

Figure 2: IPv4 Multihomed environment with Gateway Router performing NAPT

5. Other Multihoming use cases

5.1. IPv6 Network-Managed Firewall

A network-managed Firewall uses the same techniques as the premises-based firewall, but the firewall service is delivered using a security appliance positioned in the ISP. The requesting router in customer premises may obtain the PCP server addresses from the ISP delegating router, and then pass that configuration information on to the PCP clients through a DHCP server in the requesting router in the customer premises. The PCP client can also learn PCP servers using other PCP server discovery techniques. Each PCP Server is announced independently since it is within a different ISP.

```
                        ====================
                        |    Internet       |
                        =====================
                           |              |
                           |              |
                      +----+------+     +-+---------+
                      |           |     |           |
                      | ISP1 with |     | ISP2 with |
                      | FW1       |     | FW2       |
                      +----+------+     +-+---------+ ISP Network
                           |              |
     ............................................................
                           |              |
                           | Port1        | Port2   Subscriber Network
                           |              |
                      +----+-----------------+
                      |      Gateway         |
                      |       Router         |
                      +----+-----------------+
                           |
                           |
                           |
                      -----+-+-----+------
                           |
                         +-+-----+
                         | Hosts |
                         +-------+
```

Figure 3: Network-Managed Firewall

When the PCP client sends a PCP request to the PCP server deployed in the ISP and the source address of the PCP request is not the one that is delegated by the upstream ISP, then that PCP request will be dropped at the ISP by its ingress filter rule. Ingress filtering is becoming more popular among ISPs to mitigate the damage of denial-of-service (DoS) attacks as explained in section 2.1.2 of [RFC5220]. In IPv6 multihoming the PCP client will eventually learn that the PCP

server responds to only PCP requests with specific source address after few attempts and hence can discard sending PCP requests with wrong source address to the PCP server provided by the ISP.

5.2. IPv4 Policy based Routing

Policy based Routing (PBR) with multi-homing is typically used in enterprises to route packets from the same source IP address to different ISP based on configuration policies and match conditions based on source IP address, destination IP address, destination port, DSCP value(s), L4 and L7 protocols (e.g., SIP, RTP, RTSP) etc. For e.g. a site with Dual WAN connections Gold-ISP, Bronze-ISP and uses Gold-ISP for certain traffic only (e.g. Media). In such a environment NAT has different NAT pools and would rely on pre-configured PBR policy to determine which NAT address pool to use when an IP packet comes from an internal host. PCP allows a host to interact with a PCP-controlled NAT device and request an external IP and port. Therefore a PCP Server that controls the NAT device with PBR and receives a PCP request from a PCP client needs to know from which NAT pool to allocate an external IP address and port.

The PCP PEER request would contain the destination IP address, destination port and transport protocol of the remote peer that the PCP client will be trying to communicate with. The PCP MAP request with FILTER option would also contain the destination IP address, transport port but the destination port could be all ports. The NAT device based on the information present in the PCP request can possibly select the NAT pool, create mapping and return the external IP address and port in PCP response.

There is also a possibility that PBR is determined based on other information like L7 protocol, DSCP value(s) that is not conveyed by default in the PCP PEER or MAP with FILTER option. Further In case of PCP MAP request with just the 3-tuple information (internal port, protocol and source IP address), the NAT device does not know which NAT pool to use. Hence if the information conveyed in PCP request is not sufficient to execute the policy then the PCP server will return a new error code (PROVIDE_MORE_DATA) in the PCP response to the PCP client asking it to provide additional information in subsequent PCP requests. The PCP client can then convey more information like DESCRIPTION, DSCP_POLICY using the PCP extensions defined in [I-D.boucadair-pcp-extensions].

6. Multiple interfaces and Servers

One interesting case for PCP multi-homing is when a end host such as a mobile terminal has multiple interfaces concurrently active, for

example, Wi-Fi and 3G. In this case PCP client would discover different PCP Servers over different interfaces. Although multiple interfaces are available, an application might choose to use just one based on, for example, bandwidth requirements, and therefore would need to send PCP requests to just one PCP Server.

This scenario requires further discussion. TBD

7. Security Considerations

Security considerations in [I-D.ietf-pcp-base] apply to this use.

8. IANA Considerations

The following PCP result code is to be allocated : PROVIDE_MORE_DATA

9. References

9.1. Normative References

[I-D.boucadair-pcp-extensions]
Boucadair, M., Penno, R., and D. Wing, "Some Extensions to Port Control Protocol (PCP)", draft-boucadair-pcp-extensions-03 (work in progress), April 2012.

[I-D.boucadair-pcp-server-selection]
Boucadair, M., Penno, R., and D. Wing, "PCP Server Selection", draft-boucadair-pcp-server-selection-00 (work in progress), September 2012.

[I-D.bpw-pcp-proxy]
Boucadair, M., Penno, R., Wing, D., and F. Dupont, "Port Control Protocol (PCP) Proxy Function", draft-bpw-pcp-proxy-02 (work in progress), September 2011.

[I-D.ietf-pcp-base]
Wing, D., Cheshire, S., Boucadair, M., Penno, R., and P. Selkirk, "Port Control Protocol (PCP)", draft-ietf-pcp-base-28 (work in progress), October 2012.

[I-D.ietf-pcp-dhcp]
Boucadair, M., Penno, R., and D. Wing, "DHCP Options for the Port Control Protocol (PCP)", draft-ietf-pcp-dhcp-05 (work in progress), September 2012.

9.2. Informative References

[RFC5220] Matsumoto, A., Fujisaki, T., Hiromi, R., and K. Kanayama, "Problem Statement for Default Address Selection in Multi-Prefix Environments: Operational Issues of RFC 3484 Default Rules", RFC 5220, July 2008.

Authors' Addresses

Prashanth Patil
Cisco Systems, Inc.
Cessna Business Park, Varthur Hobli
Sarjapur Marthalli Outer Ring Road
Bangalore, Karnataka 560103
India

Email: praspati@cisco.com

Tirumaleswar Reddy
Cisco Systems, Inc.
Cessna Business Park, Varthur Hobli
Sarjapur Marathalli Outer Ring Road
Bangalore, Karnataka 560103
India

Email: tireddy@cisco.com

Reinaldo Penno
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, California 95134
USA

Email: repenno@cisco.com

Dan Wing
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, California 95134
USA

Email: dwing@cisco.com

PCP
Internet-Draft
Intended status: Standards Track
Expires: July 25, 2013

T. Reddy
Cisco
M. Isomaki
Nokia
D. Wing
P. Patil
Cisco

January 21, 2013

Optimizing NAT and Firewall Keepalives Using Port Control Protocol (PCP)
draft-reddy-pcp-optimize-keepalives-01

Abstract

This document describes how Port Control Protocol is useful to reduce NAT and firewall keepalive messages for a variety of applications.

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 July 25, 2013.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as

described in the Simplified BSD License.

Table of Contents

1. Introduction 3
2. Notational Conventions 3
3. Overview of Operation 3
3.1. Application Scenarios 3
3.2. NAT and Firewall Topologies and Detection 5
3.3. Detect PCP Unaware Firewalls 7
3.4. Keepalive Optimization 7
4. Keepalive Interval Determination Procedure when PCP unaware Firewall or NAT is detected 7
5. Application-Specific Operation 9
5.1. SIP 9
5.2. HTTP 9
5.3. Media and data channels with ICE 10
5.4. Detecting Flow Failure 11
5.5. Firewalls 11
5.5.1. IPv6 Network with Firewalls 11
5.5.2. Mobile Network with Firewalls 12
6. IANA Considerations 12
7. Security Considerations 12
8. Acknowledgements 12
9. Change History 12
9.1. Changes from draft-reddy-pcp-optimize-keepalives-00 . . . 12
10. References 13
10.1. Normative References 13
10.2. Informative References 13
Appendix A. Example PHP script 14
Authors' Addresses 14

1. Introduction

Many types of applications need to keep their Network Address Translator (NAT) and Firewall (FW) mappings alive for long periods of time, even when they are otherwise not sending or receiving any traffic. This is typically done by sending periodic keep-alive messages just to prevent the mappings from expiring. As NAT/FW mapping timers may be short and unknown to the endpoint, the frequency of these keep-alives may be high. An IPv4 or IPv6 host can use the Port Control Protocol (PCP)[I-D.ietf-pcp-base] to flexibly manage the IP address and port mapping information on NATs and FWs to facilitate communications with remote hosts. This document describes how PCP can be used to reduce keep-alive messages for both client-server and peer-to-peer type of communication.

The mechanism described in this document is especially useful in cellular mobile networks, where frequent keep-alive messages make the radio transition between active and power-save states causing signaling congestion. The excessive time spent on the active state due to keep-alives also greatly reduces the battery life of the cellular connected devices such as smartphones or tablets. Requirement #14 in [I-D.binet-v6ops-cellular-host-reqs-rfc3316update] explains that cellular host SHOULD support of PCP as a driver to save battery consumption exacerbated by keepalive messages.

2. Notational Conventions

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].

This note uses terminology defined in [RFC5245] and [I-D.ietf-pcp-base] .

3. Overview of Operation

3.1. Application Scenarios

PCP can help both client-server and peer-to-peer applications to reduce their keep-alive rate. The relevant applications are the ones that need to keep their NAT/FW mappings alive for long periods of time, for instance to be able to send or receive application messages in both directions at any time.

A typical client-server scenario is depicted in Figure 1. A client, who may reside behind one or multiple layers of NATs/FWs, opens a

connection to a globally reachable server, and keeps it open to be able to receive messages from the server at any time. The connection may be a connection-oriented transport protocol such as TCP or SCTP or connection-less transport protocol such as UDP. Protocols operating in this manner include Session Initiation Protocol (SIP) [RFC3261], Extensible Messaging and Presence Protocol (XMPP) [RFC3921], Internet Mail Application Protocol (IMAP) [RFC2177] with its IDLE command, the WebSocket protocol and the various HTTP long-polling protocols. There are also a number of proprietary instant messaging, Voice over IP, e-mail and notification delivery protocols that belong in this category. All of these protocols aim to keep the client-server connection alive for as long as the application is running. When the application has otherwise no traffic to send, specific keep-alive messages are sent periodically to ensure that the NAT/FW state in the middle does not expire. The client can use PCP to keep the required mapping at the NAT/FW and use application keep-alives to keep the state on the Application Server/Peer as mentioned in Section 3.4.

```
     PCP          PCP
    Client       Server       __________
+-----------+   +------+    /          \   +-----------+
|Application|___| NAT/ |____| Internet |___|Application|
|  Client   |   |  FW  |    |          |   |   Server  |
+-----------+   +------+    \__________/   +-----------+
               (multiple
                layers)

       ------------> PCP

       ------------------------------------------>
               Application keep-alive
```

Figure 1: PCP with Client-Server applications

There are also scenarios where the long-term communication association is between two peers, both of whom may reside behind one or more layers of NAT/FW. This is depicted in Figure 2. The initiation of the association may have happened using mechanisms such as Interactive Communications Establishment (ICE), perhaps first triggered by a "signaling" protocol such as SIP or XMPP or RTCWeb. Examples of the peer-to-peer protocols include RTP and RTCWeb data channel. A number of proprietary VoIP or video call or streaming or file transfer protocols also exist in this category. Typically the communication is based on UDP, but TCP or SCTP may be used. Unless

there is no traffic flowing otherwise, the peers have to inject periodic keep-alive packets to keep the NAT/FW mappings on both sides of the communication active. Instead of application keep-alives, both peers can use PCP to control the mappings on the NAT/FWs to reduce the keep-alive frequency as explained in Section 3.4.

```
     PCP          PCP                        PCP          PCP
    Client       Server       __________    Server       Client
+-----------+   +------+    /          \   +------+   +-----------+
|Application|___| NAT/ |____| Internet |___| NAT/ |___|Application|
|   Peer    |   |  FW  |    |          |   |  FW  |   |    Peer   |
+-----------+   +------+    \__________/   +------+   +-----------+
               (multiple                  (multiple
                layers)                    layers)

       ------------> PCP                   PCP <------------

       <--------------------------------------------------->
                      Application keep-alive
```

Figure 2: PCP with Peer-to-Peer applications

3.2. NAT and Firewall Topologies and Detection

Before an application can reduce its keep-alive rate, it has to make sure it has all of the NATs and Firewalls on its path under control. This means it has to detect the presence of any PCP-unaware NATs and Firewalls on its path. PCP itself is able to detect unexpected NATs between the PCP client and server as depicted in Figure 3. The PCP client includes its own IP address and UDP port within the PCP request. The PCP server compares them to the source IP address and UDP port it sees on the packet. If they differ, there are one or more additional NATs between the PCP client and server, and the server will return an error. Unless the application has some other means to control these PCP unaware NATs, it has to fall back to its default keep-alive mechanism.

```
        PCP           PCP       PCP
       Client       Unaware    Aware        __________
   +-----------+   +------+   +------+    /          \   +-----------+
   |Application|___| NAT  |___| NAT/ |____| Internet |___|Application|
   |  Client   |   |      |   |  FW  |    |          |   |   Server  |
   +-----------+   +------+   +------+    \__________/   +-----------+

         <-----------///---------->
             PCP based detection
```

Figure 3: PCP unaware NAT between PCP client and server

Figure 4 shows a topology where one or more PCP unaware NATs are deployed on the exterior of the PCP capable NAT/FWs. To detect this, the application must have the capability to request from its server or peer what IP and transport address it sees. If those differ from the IP and transport address given to the application by the out most PCP aware NAT/FW, the application can detect that there is at least one more PCP unaware NAT on the path. In this case, the application has to fall back to its default keep-alive mechanism.

```
        PCP          PCP        PCP
       Client       Aware     Unaware      __________
   +-----------+   +------+   +------+    /          \   +-----------+
   |Application|___| NAT/ |___| NAT  |____| Internet |___|Application|
   |  Client   |   |  FW  |   |      |    |          |   |   Server  |
   +-----------+   +------+   +------+    \__________/   +-----------+

         <------------>
               PCP

         <---------------------///--------------------------->
                    Application based detection
```

Figure 4: PCP unaware NAT external to the last PCP aware NAT

Section 5 describes how the detection works in a number of real application protocols.

The caveat is that Firewalls can not be detected this way. The client will have to use the alternative procedure explained in Section 3.3 to detect PCP unaware Firewalls.

3.3. Detect PCP Unaware Firewalls

The client sends a STUN Binding Request to the STUN server. STUN server will return its alternate IP address and alternate port in OTHER-ADDRESS in the binding response [RFC5780]. The client then sends MAP request with FILTER option to PCP server to permit STUN server to reach the client using the STUN servers alternate IP address and alternate port. The client then sends a binding request to the primary address of the STUN server with the CHANGE-REQUEST attribute set to change-port and change-IP. This will cause the server to send its response from its alternate IP address and alternate port. If the client receives a response then the client is aware that on path Firewall devices are PCP aware. If the client does not receive a response then the client is aware that could be one or more on path PCP unaware Firewall devices. PCP client will perform the tests separately for each transport protocol. If no response is received, the client will then repeat the test atmost three times for connectionless transport protocols.

If the STUN server does not support OTHER-ADDRESS then this test cannot be run. This procedure can be adopted by other protocols to detect PCP unaware Firewalls.

3.4. Keepalive Optimization

If the application determines that all NATs and Firewalls on its path to the Internet support PCP, it can start using PCP instead of its default keep-alives to maintain the NAT/FW state. It can use PCP PEER Request with the Requested Lifetime set to an appropriate value. The application may still send some application-specific heartbeat messages end-to-end.

Processing the lifetime value of the PEER Opcode is described in Section 15 of [I-D.ietf-pcp-base]. Sending a PEER request with a very short Requested Lifetime can be used to query the lifetime of an existing mapping. PCP recommends that lifetimes of mapping created or lengthened with PEER be longer than the lifetimes of implicitly-created NAT and Firewall mappings. Thus PCP can be used to save battery consumption by making PCP PEER message interval longer than what the application would normally use the keep middle box state alive, and strictly shorter than the server state refresh interval.

4. Keepalive Interval Determination Procedure when PCP unaware Firewall or NAT is detected

If PCP unaware NAT/Firewall is detected then a client can use the following heuristics method to determine the keepalive interval :

1. The client sends a STUN Binding Request to the STUN server. This connection is called the Primary Channel. STUN server will return its alternate IP address and alternate port in OTHER-ADDRESS in the binding response [RFC5780].

2. The client then sends STUN Binding Request to the STUN server using alternate IP address and alternate port. This connection is called the Secondary Channel.

3. The Client will initially set the default keepalive interval for NAT/FW mappings to 60 seconds (FWa).

4. After FWa seconds the Client will send a binding request to the STUN server using the Primary Channel with the CHANGE-REQUEST attribute set to change-port and change-IP. This will cause the STUN server to send its response from the Secondary channel.

5. If the client receives response from the server then it will increase the keepalive interval value FWa = (old FWa) + (old FWa)/2. This indicates that NAT/FW mappings are alive.

6. Steps 4 and 5 will be repeated until there is no response from the STUN server. If there is no response from the STUN server then the client will use the FWa value as Keepalive interval to refresh FW/NAT mappings.

The above procedure will be done separately for each transport protocol. For connectionless transport protocols like UDP if timer of 2 seconds elapses without response from the STUN server then the client will repeat step 4 atmost three times to handle packet loss.

This procedure can be adopted by other protocols to use Primary and Secondary channels, so that the client can determine the keepalive interval to refresh FW/NAT mapping. This procedure only serves as a guideline and if applications already use some other heuristics method to determine keepalive, they can continue with the existing logic. For example Teredo determines Refresh interval using the procedure in "Optional Refresh Interval Determination Procedure" (Section 5.2.7 of [RFC4380]).

To improve reliability, applications SHOULD continue to use PCP to lengthen the FW/NAT mappings even if the above described mechanism is used to detect PCP unaware NAT/Firewall. This ensures that PCP aware FW/NAT do not close old mappings with no packet exchange when there is a resource-crunch situation.

5. Application-Specific Operation

This section describes how PCP is used with specific application protocols.

5.1. SIP

For connection-less transports the User Agent (UA) sends a STUN Binding Request over the SIP flow as described in section 4.4.2 of [RFC5626]. The UA then learns the External IP Address and Port using a PEER request/response. If the XOR-MAPPED-ADDRESS in the STUN Binding Response matches the external address and port provided by PCP PEER response then the UA optimizes the keepalive traffic as described in Section 3.4. There is no further need to send STUN Binding Requests over the SIP flow to keep the NAT binding alive.

If the XOR-MAPPED-ADDRESS in the STUN Binding Response does not match the external address and port provided by the PCP PEER response then PCP will not be used to keep the NAT bindings alive for the flow that is being used for the SIP traffic. This means that multiple layers of NAT are involved and intermediate NATs are not PCP aware. In this case the UA will continue to use the technique in section 4.4.2 of [RFC5626].

For connection-oriented transports, the UA sends a STUN Binding Request multiplexed with SIP over the TCP connection. STUN multiplexed with other data over a TCP or TLS-over-TCP connection is explained in section 7.2.2 of [RFC5389]. The UA then learns the External IP address and port using a PEER request/response. If the XOR-MAPPED-ADDRESS in the STUN Binding Response matches the external address and port provided by PCP PEER response then the UA optimizes the keepalive traffic as described in Section 3.4.

If the XOR-MAPPED-ADDRESS in the STUN Binding Response does not match the external address and port provided by PCP PEER response then PCP will not be used to keep the NAT bindings alive. In this case the UA performs a keep-alive check by sending a double-CRLF (the "ping") then waits to receive a single CRLF (the "pong") using the technique in section 4.4.1 of [RFC5626].

5.2. HTTP

Web Applications that require persistent connections use techniques such as HTTP long polling and Websockets for session keep alive as explained in section 3.1 of [I-D.isomaki-rtcweb-mobile]. In such scenarios, after the client establishes a connection with the HTTP server, it can execute server side scripts such as PHP residing on the server to provide the transport address and port of the HTTP

client seen at the HTTP server. In addition, the HTTP client also learns the external IP Address and port using the PCP PEER request/ response.

If the IP address and port learned from the server matches the external address and port provided by PCP PEER response then the HTTP client optimizes keepalive traffic as described in Section 3.4.

If the IP address and port do not match then PCP will not be used to keep the NAT bindings alive for the flow that is being used for the HTTP traffic. This means that there are NATs or HTTP proxies between the PCP server and the HTTP server. The HTTP client will have to resort to use existing techniques for keep alive. Please see Appendix A for an example server side PHP script to obtain the client source IP address.

HTTP protocol allows intermediaries like transparent proxies to be involved and there is no way for the client to know that request/ response is relayed through a proxy.

5.3. Media and data channels with ICE

The ICE agent learns the External IP Address and Port using a MAP request/response. This candidate learnt through PCP is encoded in the ICE offer and answer just like the server reflexive candidate, If the server reflexive candidate and External IP address learnt using PCP are different. When using the Recommended Formula in section 4.1.2.1 of [RFC5245] to compute priority for the candidates learnt through PCP, the ICE agent can use a preference value greater than or equal to the server reflexive candidates.

The ICE agent in addition to ICE connectivity checks and performs the following :

The ICE agent checks if the XOR-MAPPED-ADDRESS from the STUN [RFC5389] Binding response received as part of ICE connectivity check matches the external address and port provided by PCP MAP response.

1. If the match is successful then PCP will be used to keep the NAT bindings alive. The ICE agent optimizes keepalive traffic by refreshing the mapping via a new PCP MAP request containing information from the earlier PCP response.

2. If the match is not successful then PCP will not be used for keep NAT binding alive. The ICE agent will use the technique in section 4.4 of [RFC6263] to keep NAT bindings alive. This means that multiple layers of NAT are involved and intermediate NATs are not PCP aware.

Some network operators deploying a PCP Server may allow PEER but not MAP. In such cases the ICE agent learns the external IP address and port using a STUN binding request/response during ICE connectivity checks. The ICE agent also learns the external IP Address and port using a PCP PEER request/response. If the IP address and port learned from the STUN binding response matches the external address and port provided by the PCP PEER response then the ICE agent optimizes keepalive traffic as described in Section 3.4.

5.4. Detecting Flow Failure

Using the Rapid Recovery technique in section 14 of [I-D.ietf-pcp-base] PCP client upon receiving a PCP ANNOUNCE from a PCP server becomes aware that PCP server has rebooted or lost its mapping state. The PCP client issues new PCP requests to recreate any lost mapping state and thus reconstructs lost mappings fast enough that existing media, HTTP and SIP flows do not break. If the NAT state cannot be recovered the endpoint will find the new external address and port as part of the Rapid Recovery technique in PCP itself and reestablish a connection with the peer.

In lieu of this mechanism if a PCP server reboots and loses its mapping state or when a NAT gateway has its external IP address changed so that its current mapping state becomes invalid, it may take some time before the endpoints realize that the connectivity is lost.

5.5. Firewalls

PCP allows applications to communicate with Firewall devices with PCP functionality to create mappings for incoming connections. In such cases PCP can be used by the endpoint to create an explicit mapping on Firewall to permit inbound traffic and further use PCP to send keep-alives to keep the Firewall mappings alive.

5.5.1. IPv6 Network with Firewalls

As part of the call setup, the endpoint would gather its host candidates and relayed candidate from a TURN server, send the candidates in the offer to the peer endpoint. On receiving the answer from the peer endpoint, the PCP client sends a PCP MAP request with FILTER opcode to create a dynamic mapping in Firewall to permit ICE connectivity checks and subsequent media traffic from the remote peer.

5.5.2. Mobile Network with Firewalls

Mobile Networks are also making use of a Firewall to protect their customers from various attacks like downloading malicious content. The Firewall is usually configured to block all unknown inbound connections as explained in section 2.1 of [I-D.chen-pcp-mobile-deployment]. In such cases PCP can be used by Mobile devices to create an explicit mapping on the Firewall to permit inbound traffic and optimize the keepalive traffic as described in Section 3.4. This would result in saving of radio and power consumption of the Mobile device while protecting it from attacks.

6. IANA Considerations

None

7. Security Considerations

The security considerations in [RFC5245]and [I-D.ietf-pcp-base] apply to this use.

8. Acknowledgements

Authors would like to thank Dave Thaler, Basavaraj Patil for valuable inputs to the document.

9. Change History

[Note to RFC Editor: Please remove this section prior to publication.]

9.1. Changes from draft-reddy-pcp-optimize-keepalives-00

- o Added sections 3.3, 4
- o Updated section 3 an 3.4 and Introduction

10. References

10.1. Normative References

[I-D.ietf-pcp-base]
Wing, D., Cheshire, S., Boucadair, M., Penno, R., and P. Selkirk, "Port Control Protocol (PCP)", draft-ietf-pcp-base-29 (work in progress), November 2012.

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

[RFC5245] Rosenberg, J., "Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols", RFC 5245, April 2010.

[RFC5389] Rosenberg, J., Mahy, R., Matthews, P., and D. Wing, "Session Traversal Utilities for NAT (STUN)", RFC 5389, October 2008.

[RFC5626] Jennings, C., Mahy, R., and F. Audet, "Managing Client-Initiated Connections in the Session Initiation Protocol (SIP)", RFC 5626, October 2009.

[RFC5780] MacDonald, D. and B. Lowekamp, "NAT Behavior Discovery Using Session Traversal Utilities for NAT (STUN)", RFC 5780, May 2010.

[RFC6263] Marjou, X. and A. Sollaud, "Application Mechanism for Keeping Alive the NAT Mappings Associated with RTP / RTP Control Protocol (RTCP) Flows", RFC 6263, June 2011.

10.2. Informative References

[I-D.binet-v6ops-cellular-host-reqs-rfc3316update]
Binet, D., Boucadair, M., Ales, V., Byrne, C., and G. Chen, "Internet Protocol Version 6 (IPv6) for Cellular Hosts", draft-binet-v6ops-cellular-host-reqs-rfc3316update-03 (work in progress), October 2012.

[I-D.chen-pcp-mobile-deployment]
Chen, G., Cao, Z., Boucadair, M., Ales, V., and L. Thiebaut, "Analysis of Port Control Protocol in Mobile Network", draft-chen-pcp-mobile-deployment-02 (work in progress), October 2012.

[I-D.isomaki-rtcweb-mobile]
Isomaki, M., "RTCweb Considerations for Mobile Devices",

draft-isomaki-rtcweb-mobile-00 (work in progress), July 2012.

[RFC2177] Leiba, B., "IMAP4 IDLE command", RFC 2177, June 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.

[RFC3921] Saint-Andre, P., Ed., "Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence", RFC 3921, October 2004.

[RFC4380] Huitema, C., "Teredo: Tunneling IPv6 over UDP through Network Address Translations (NATs)", RFC 4380, February 2006.

Appendix A. Example PHP script

```
<html>
Connected to <?PHP echo gethostname(); ?> on port <?PHP echo
getenv(SERVER_PORT) ?> on <?PHP echo date("d-M-Y H:i:s"); ?> Pacific Time
<p>
Your IP address is: <?PHP echo getenv(REMOTE_ADDR); ?>,
port <?PHP echo getenv(REMOTE_PORT); ?>
</p>;
</html>
```

Authors' Addresses

Tirumaleswar Reddy
Cisco Systems, Inc.
Cessna Business Park, Varthur Hobli
Sarjapur Marathalli Outer Ring Road
Bangalore, Karnataka 560103
India

Email: tireddy@cisco.com

Markus Isomaki
Nokia
Keilalahdentie 2-4
FI-02150 Espoo
Finland

Email: markus.isomaki@nokia.com

Dan Wing
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, California 95134
USA

Email: dwing@cisco.com

Prashanth Patil
Cisco Systems, Inc.
Cessna Business Park, Varthur Hobli
Sarjapur Marthalli Outer Ring Road
Bangalore, Karnataka 560103
India

Email: praspati@cisco.com

Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: April 25, 2013

Q. Sun
China Telecom
M. Boucadair
X. Deng
France Telecom
C. Zhou
Huawei Technologies
T. Tsou
Huawei Technologies (USA)
October 22, 2012

Lightweight 4over6 Port-set Allocation: Using PCP To Coordinate Between the CGN and Home Gateway
draft-tsou-pcp-natcoord-08

Abstract

This document defines an extension to the base PCP. New OpCode is defined to enhance PCP with the ability to reserve port sets for internal hosts.

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 April 25, 2013.

Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents

carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Application Scenario 3
2. MAP_PORT_SET Opcode 3
2.1. MAP_PORT_SET Operation Packet Formats 4
2.2. MAP_PORT_SET Mapping Table Example 8
3. MAP_PORT_SET Operation 8
3.1. Generating a MAP_PORT_SET Request 8
3.2. Renewing a MAP_PORT_SET Mapping 8
3.3. Processing a MAP_PORT_SET Request 9
3.4. Processing a MAP_PORT_SET Response 10
4. Mapping Lifetime and Deletion 10
5. PREFER_FAILURE Option for MAP_PORT_SET Opcode 10
6. Coexistence with MAP OpCode 10
7. MAP_PORT_SET Failover 11
8. Security Considerations 11
9. IANA Considerations 11
10. Authors List 11
11. References 12
11.1. Normative References 12
11.2. informative References 12
Authors' Addresses 13

1. Application Scenario

PCP can be used to control an upstream device to achieve the following goals:

1. A plain IP address (i.e., a non-shared) can be assigned to a given subscriber because it subscribed to a service which uses a protocol that don’t embed a transport number or because the NAT is the only deployed platform to manage IP addresses.

2. An application (e.g., sensor) does not need to listen to a whole range of ports available on a given IP address. Only a limited set of ports are used to bind its running services. For such devices, the external port(s) and IP address can be delegated to that application and therefore avoid enforcing NAT in the network side for its associated flows. The NAT in the PCP- controlled device should be bypassed.

3. A device able to restrict its source ports can be delegated an external port restricted IP address. The PCP- controlled device should be instructed to by-pass the NAT when handling flows destined/issued to that device.

This document extends PCP with the ability to reserve port sets instead of individual ports. This is motivated by the need to offload to a port-restricted device in lightweight 4over6 [I-D.cui-softwire-b4-translated-ds-lite], reduce the logging and enhance the performance of the CGN.

Using individual MAP requests to reserve all individual ports of a given port set can not achieve this goal because an additional indication is needed to instruct the PCP-controlled device to not enforce a NAT for packets matching these ports. A candidate solution is to define a new Option to request for this feature be enforced by the PCP-controlled device. Nevertheless, this solution is not efficient when large port sets are assigned (e.g., address sharing ratio of 1:2 or 1:8). Another issue, is when no NAT is enforced in the PCP-controlled device but only a Port Range Router (PRR) function, the request has not to indicate the internal ports.

For those reasons, a new PCP OpCode is defined in this document.

2. MAP_PORT_SET Opcode

This section defines a new Opcode to request a port set from a PCP-controlled device.

The format of MAP_PORT_SET is designed to be close to the MAP message format. The port set is encoded using a port mask to convey a contiguous port range.

By analogy, a port set binding can be seen as an aggregate of MAP mappings. When assigning a port set to a PCP Client, the PCP-controlled device maintains a binding between the source IP address of the PCP request, the assigned external IP address and the assigned port set. Allocating port sets can greatly reduce individual MAP requests for a PCP client when requesting a bulk of ports at one time. This mechanism can be applied for lightweight 4over6 [I-D.cui-softwire-b4-translated-ds-lite] in port-set allocation process. It can also be applied to stateless PCP-controlled device, in which the Internal address, External address and Port set is determined algorithmically.

MAP_PORT_SET: Create an explicit dynamic mapping between an Internal IP Address and an External IP Address + Port set

It is totally up to the PCP server to determine the port-set quota for each PCP client. In addition, when the PCP-controlled device supports multiple port-sets delegation for a given PCP client, the PCP client MAY re-initiate a PCP request to get another port set when it has exhausted all the ports within the port-set.

PCP-controlled device SHOULD provide a configuration option to allow administrators to configure the size of each individual port set (denoted as MAX_REQUEST_QUOTA) to be assigned and the size of the total ports for a PCP client (denoted as MAX_USER_QUOTA).

2.1. MAP_PORT_SET Operation Packet Formats

The MAP_PORT_SET Opcode has a similar packet layout for both requests and response. Figure 1 shows the format of the MAP_PORT_SET request.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                 PORT_SET_Nonce (96bits)                       |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Protocol    |              Reserved (24 bits)               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Suggested Port Set Index     |  Suggested Port Set Mask       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|             Suggested External IP Address (128 bits)          |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 0                   1                   2                   3
```

Figure 1: MAP_PORT_SET Opcode Request format

These fields are described below:

- o Requested lifetime (in common header): Requested lifetime of this port set mapping, in seconds. The value 0 indicates "delete".

- o PORT_SET_Nonce: Random value chosen by the PCP client which SHOULD be different for individual PCP requests. But the same value MUST be kept in one request re-transmission. See Section 11.2 of [I-D.ietf-pcp-base].

- o Protocol: the default value is zero (to indicate all transport protocols).

- o Reserved bits: 24 bits MUST be set to 0.

- o Suggested Port Set Index (PSI): The PSI indicates the value of the significant bits of the Port Mask. By default, PSI is set to 0 in a request. It can also convey Suggested Port Set Index if the client has a hint on it. The first k bits on the left of the 2-octet field is the Port Set Index value, with the rest of the field right padding zeros.

- o Suggested Port Set Mask (PSM): The PSM indicates the position of the bits that are used to build the Port Set Index. The 1 values in the Port Set Mask indicate by their position the significant bits of the Port Set Value. By default, PSM is set to 0 in a request. It can also convey Suggested Port Set Mask if the client

has a hint on it. The first k bits on the left is padding ones while the remained (16-k) bits of the 2-octet field on the right is padding zeros.

- o Suggested External IP Address: Suggested external IPv4 or IPv6 address. Same as Section 10.1 of [I-D.ietf-pcp-base].

In the context of Port Set Option, the port number should consist of port set prefix and port number suffix. The port set prefix can be got from Port Set Index and Port Set Mask, while port number suffix can change continuously. The format of port number is shown below.

```
0                                                       15
+-----------------------+-----------------------------+
|    port set prefix    |      port number suffix     |
+-----------------------+-----------------------------+
|<-------k bits-------->|<--------(16-k) bits-------->|
```

In order to exclude the system ports ([I-D.ietf-tsvwg-iana-ports]) or ports saved by SPs, the former port-sets that contains well-known ports SHOULD NOT be assigned.

Figure 2 shows the format of Opcode-specific information in a response packet for the MAP_PORT_SET Opcode:

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                 PORT_SET_Nonce (96bits)                       |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Protocol    |              Reserved (24 bits)               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Suggested Port Set Index     |  Suggested Port Set Mask       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|           Assigned External IP Address (128 bits)             |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 0                   1                   2                   3
```

Figure 2: MAP_PORT_SET Opcode format of Response

These fields are described below:

- o Lifetime (in common header): On an error response, this indicates how long clients should assume they'll get the same error response from the PCP server if they repeat the same request. On a success response, this indicates the lifetime for this mapping, in seconds.

- o PORT_SET_Nonce: MUST be copied from the request.

- o Protocol: MUST be copied from the request.

- o Reserved bits: 16 bits MUST be set to 0.

- o Assigned Port Set Index (PSI): The PSI indicates the value of the significant bits of the Port Mask.

- o Assigned Port Set Mask (PSM): The Port Set Mask indicates the position of the bits that are used to build the Port Set Index. The 1 values in the Port Set Mask indicate by their position the significant bits of the Port Range Value.

- o Assigned External IP Address (128 bits): This field conveys the assigned external IPv4 (encoded using IPv4-mapped IPv6 address) or IPv6 address for the mapping. On an error response, the Assigned External IP Address is copied from the request.

2.2. MAP_PORT_SET Mapping Table Example

The following table depicts an example of the mapping table in the PCP Server enabling MAP_PORT_SET OpCode.

Internal Address	External Address	Port Range	Protocol	NONCE
2001:db8::1	192.0.2.33	5120-6143	0	nonce1
2001:db8::2	192.0.2.33	6144-7167	0	nonce2
2001:db8::3	192.0.2.33	7168-8191	0	nonce3
2001:db8::4	192.0.2.33	8192-9215	0	nonce4

Figure 3: Mapping table example in MAP_PORT_SET

3. MAP_PORT_SET Operation

3.1. Generating a MAP_PORT_SET Request

The MAP_PORT_SET request MUST contain values in the Suggested IP Address field, Suggested Port Set Index and Suggested Port Mask. However, this port set indicated in the request of the PCP Client is only a hint; it is up to the PCP Server to assign a port set.

If a PCP Client fails to receive an expected response from a server, the PCP client follows the same retransmission procedure defined for MAP in the base PCP specification (section 8.1.1 of [I-D.ietf-pcp-base]). The PORT_SET_Nonce should be copied from the previous MAP_PORT_SET request.

If a PCP Client uses out all the ports in the current assigned port set, it MAY generate a new MAP_PORT_SET Request to get another delegated port-set. The Client MUST use a different Mapping Nonce for different MAP_PORT_SET requests. If USER_EX_QUOTA error is received from the server, the PCP client SHOULD NOT request for another new port set.

3.2. Renewing a MAP_PORT_SET Mapping

Port Set mapping renewal for MAP_PORT_SET MUST follow the same procedure for an individual MAP mapping (section 11.2.1 of [I-D.ietf-pcp-base]) except for considerations related to the internal port (which is included in a MAP request but not present in a MAP_PORT_SET).

The MAP_PORT_SET request MUST include the currently assigned IP address and port-set in the Suggested IP address,Suggested Port Set Index and Suggested Port Set Mask.

3.3. Processing a MAP_PORT_SET Request

The PCP server SHOULD take exactly the same order as in (section 11.3 of [I-D.ietf-pcp-base]). In particular, as there is no Internal Port in MAP_PORT_SET anymore, all the processes regarding to Internal Port should be neglected accordingly.

Considerations related to the assignment of the external IP Address are the same as what is defined in (section 11.3 of [I-D.ietf-pcp-base]).

The procedures regarding to the port set are similar to the external port processes in MAP Opcode (section 11.3 of [I-D.ietf-pcp-base]), except that the whole port-set should be treated consistently in MAP_PORT_SET Opcode. The same operations for handling the Suggested external port for a MAP request are applied on the Suggested Port Set.

The procedures for PORT_SET_Nonce is exactly the same as the Mapping Nonce field defined in (section 11.3 of [I-D.ietf-pcp-base]). The PCP server only needs to remember ONE PORT_SET_Nonce for each mapping (Internal IP Address, External IP address and Port Set).

The error codes in MAP_PORT_SET Response mainly have the following possibilities:

- If the PCP server or PCP-controlled device does not support MAP_PORT_SET Opcode, the error UNSUPP_OPCODE MUST be returned.

- If an option does not make sense, (e.g., the PREFER_FAILURE Option is included in a request with lifetime=0, etc.,), the request is invalid and generates a MALFORMED_OPTION error. This procedure is the same with section 10.3 of [I-D.ietf-pcp-base].

If the requested lifetime is zero, it indicates a request to delete an existing mapping.

A PCP server SHOULD maintain MAX_USER_QUOTA and MAX_REQUEST_QUOTA. MAX_USER_QUOTA is to indicate the maximum number of ports a subscriber may get in total, and MAX_REQUEST_QUOTA is to indicate the maximum number of ports in each request. Therefore, one PCP Client will have up to N mappings, in which N SHOULD NOT be larger than floor(MAX_USER_QUOTA/MAX_REQUEST_QUOTA). The specific mechanism to configure the quotas is out of scope.

If the PCP server is configured to allocate multiple port-set allocation for one subscriber, the same External address SHOULD be assigned to one subscriber in multiple port-set requests to guarantee the consistency.

To optimize the number of mapping entries maintained by the PCP server, it is RECOMMENDED to configure the server to assign the maximum allowed port set in a single response. This policy SHOULD be configurable.

When MAP_PORT_SET is applied to stateless PCP-controlled device, the PCP server returns an answer indicating the external IP address and port-set as seen by remote peers.

3.4. Processing a MAP_PORT_SET Response

On receiving a MAP_PORT_SET Response, the same procedure as the one for individual mapping [section 10.4 of [I-D.ietf-pcp-base]] MUST be followed by the PCP Client to validate the response (except the considerations related to the internal port).

4. Mapping Lifetime and Deletion

The procedure for port-set mapping lifetime and deletion is also the same with individual mapping [section 10.5 of [I-D.ietf-pcp-base]].

5. PREFER_FAILURE Option for MAP_PORT_SET Opcode

This option [section 10.2 of [I-D.ietf-pcp-base]] can be applied to MAP_PORT_SET Opcode indicating that if the PCP server cannot map the suggested External Address and port-set, the PCP server should not create a mapping.

6. Coexistence with MAP OpCode

Normally, the PCP server for MAP_PORT_SET will not run NAT. So there is no NAT binding in PCP and the PCP server will not run MAP OpCode for the same subscriber. In the case when the PCP client is embedded in the host and the PCP server keeps the NAT bindings for some special-purpose applications, the external address and the port allocated to the subscriber should be consistent with the ones in MAP_PORT_SET response.

7. MAP_PORT_SET Failover

The failover mechanism in MAP [section 14 in [I-D.ietf-pcp-base]] and [I-D.boucadair-pcp-failure] can also be applied to MAP_PORT_SET.

The only difference compared to MAP is the amount of Mapping entries in MAP_PORT_SET PCP server is much less than MAP. Therefore, the cost of state synchronization has been greatly reduced in MAP_PORT_SET.

8. Security Considerations

The same security considerations discussed in [I-D.ietf-pcp-base] have to be taken into account.

9. IANA Considerations

The authors request the following new OpCode: MAP_PORT_SET

10. Authors List

The following are extended authors who contributed to the effort:

Yunqing Chen

China Telecom

Room 502, No.118, Xizhimennei Street

Beijing 100035

P.R.China

Chongfeng Xie

China Telecom

Room 502, No.118, Xizhimennei Street

Beijing 100035

P.R.China

Yong Cui

Tsinghua University

Beijing 100084

P.R.China

Phone: +86-10-62603059

Email: yong@csnet1.cs.tsinghua.edu.cn

Qi Sun

Tsinghua University

Beijing 100084

P.R.China

Phone: +86-10-62785822

Email: sunqibupt@gmail.com

Gabor Bajko

Nokia

Email: gabor.bajko@nokia.com

11. References

11.1. Normative References

[I-D.ietf-pcp-base]
Wing, D., "Port Control Protocol (PCP)", October 2012.

11.2. informative References

[I-D.boucadair-pcp-failure]
Boucadair, M., Dupont, F., and R. Penno, "Port Control Protocol (PCP) Failure Scenarios", August 2012.

[I-D.cui-softwire-b4-translated-ds-lite]
Cui, Y., Sun, Q., Boucadair, M., Tsou, T., and Y. Lee, "Lightweight 4over6: An Extension to DS-Lite Architecture", Feb 2012.

Authors' Addresses

Qiong Sun
China Telecom
P.R.China

Phone: 86 10 58552936
Email: sunqiong@ctbri.com.cn

Mohamed Boucadair
France Telecom
Rennes, 35000
France

Email: mohamed.boucadair@orange.com

Xiaohong Deng
France Telecom

Email: xiaohong.deng@orange-ftgroup.com

Cathy Zhou
Huawei Technologies
Bantian, Longgang District
Shenzhen 518129
P.R. China

Phone:
Email: cathy.zhou@huawei.com

Tina Tsou
Huawei Technologies (USA)
2330 Central Expressway
Santa Clara, CA 95050
USA

Phone: +1 408 330 4424
Email: Tina.Tsou.Zouting@huawei.com