Using PCP to Find an External Address in an NPTv6 Network

draft-baker-pcp-nptv6-search-00

Abstract

This note describes an approach to finding the set of External Addresses associated with an Internal Address.

Requirements

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

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 23, 2012.

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. Using Multicast PCP ............................................ 3
   2.1. PCP Client: Generating a Request ...................... 3
   2.2. PCP Server: Processing a Request ..................... 4
   2.3. PCP Client: Processing Responses ..................... 4
   2.4. Recovery .................................................. 5
3. An implementation approach ..................................... 5
4. IANA Considerations ........................................... 5
5. Operational Considerations .................................... 5
6. Security Considerations ....................................... 6
7. Acknowledgements .............................................. 6
8. Change Log ..................................................... 6
9. References ..................................................... 7
   9.1. Normative References .................................... 7
   9.2. Informative References ................................... 7
Authors’ Addresses .................................................. 7
1. Introduction

This note uses terminology defined in [I-D.ietf-pcp-base].

Section 5 of [RFC6296] points out that there can be issues when an application refers a session initiated by a peer to a third party application running on the same or a different system; it must identify the system the third party application is running on. This is often done by citing an IP address or a DNS name that maps to one or more IP addresses.

In a network that uses Network Address Translation or Network Prefix Translation technology, referrals using IP addresses imply that the application must be able to identify both the Internal and External Addresses of the third party. Similarly, when a peer system queries DNS to find an address (either for the initial access or because a referral used a DNS name), DNS must supply it with an address appropriate to its domain. If the two are both in the same network, that would be the Internal Address, and otherwise all External Addresses.

This note describes an approach to finding the External Addresses associated with an Internal Address.

2. Using Multicast PCP

Consider a scenario in which the firewall or other system implementing the NPTv6 Translator also implements a Port Control Protocol (PCP) [I-D.ietf-pcp-base] Server, and that PCP Server joins a multicast group ALL-NPTv6-TRANSLATORS. A PCP Client could send PCP Requests to the multicast group, and get responses from every NPTv6 Translator in the domain.

2.1. PCP Client: Generating a Request

The PCP client sends a MAP Request to that multicast group address, with Internal Port=0 and Protocol=0 (which means ‘all ports for all protocols’). To accommodate packet loss, the request SHOULD be transmitted several times with a random jitter between them. It is RECOMMENDED to transmit the MAP Request a total of three times with the first retransmission after 5 seconds plus a random value between 0-2.5 seconds, and again at 10 seconds plus a random value between 0-5 seconds.
2.2. PCP Server: Processing a Request

The PCP server embedded in the NPTv6 device first verifies that the PCP message conforms to the requirements of a PCP MAP request as described in [I-D.ietf-pcp-base]. Then it checks that the MAP request field’s Protocol and Internal Port are both zero; if not, a MALFORMED_REQUEST error is generated.

If the MAP request contains the THIRD_PARTY option, it MUST contain an IPv6 address, otherwise it results in a MALFORMED_OPTION error.

Then, depending on the IPv6 prefix of the PCP MAP request (or the IPv6 prefix of the THIRD_PARTY option, if present):

1. If no translation applies to that IPv6 address (i.e., the address is not within a prefix that is translated) the Assigned External Address of the MAP response is set to the same as the IP address from the IP header of the PCP request (unless THIRD_PARTY was used, in which case it is set to the IP address of the THIRD_PARTY option).

2. If a translation would occur, the external address is returned in the Assigned External Address field.

If the NPTv6 device itself is multihomed (i.e., it contains multiple NPTv6 translation functions), a separate MAP Response is sent for each NPTv6 instance -- as if they were separate devices. These MAY be sent from the same unicast source address.

It is RECOMMENDED that the Assigned Lifetime of the MAP response be the remaining lifetime of the ISP-assigned address. In this way, PCP clients receive timely updates to the IPv6 address assigned by the ISP.

2.3. PCP Client: Processing Responses

Each MAP request sent to the multicast group will result in zero, one, or more responses (from each NPTv6 listening to that multicast group).

If the network contains multiple NPTv6 instances, multiple MAP responses will normally be received. If multiple responses are received, the shortest PCP Assigned Lifetime should be used when sending renewal multicast PCP requests.

Renewals should follow the procedure described in Section 10.2.1 of [I-D.ietf-pcp-base].
2.4. Recovery

An NPTv6 device may join or leave a network unexpectedly (e.g., device failure, link failure, or link recovery). To accommodate these situations, the NPTv6 devices SHOULD implement PCP Rapid Recovery, as described in Section 13 of [I-D.ietf-pcp-base].

3. An implementation approach

A practical implementation of the PCP client in this case would be in a DNS Server [RFC1034][RFC1035]. Whenever it learns of a mapping between a name and an Internal Address (which might happen only at startup in a static system, or might happen frequently if Dynamic DNS [RFC2136][RFC3007] is used with IPv6 Privacy Addresses [RFC4941]), the DNS Server queries ALL-NPTv6-TRANSLATORS for the list of relevant addresses to create AAAA Resource Records for. It may get no response (although if there are no such translators one would hope for an ICMP Host Unreachable response rather than letting it time out), one response, or many. It always makes a Resource Record for the Internal Address; it also makes Resource Records for any External Addresses reported. Such translations come with lifetimes; the DNS Server is responsible to re-request as lifetimes expire, and to not grant longer Resource Record lifetimes than it has address lifetimes.

Any system needing to know its own External Addresses or those of another party could then obtain them by resolving the relevant DNS name, or could follow the same process.

4. IANA Considerations

This note requests of the IANA the assignment of a set of multicast addresses as described in Section 2.7 of the IP Version 6 Addressing Architecture [RFC4291] from the registry [v6mult]. This set of addresses is referred to as "ALL-NPTv6-TRANSLATORS". One address should be assigned for each of the following scopes: Link-Local, Admin-Local, Site-Local, and Organization-Local.

5. Operational Considerations

This document defines a set of multicast addresses in several scopes. Operationally, the choice of which scope is appropriate is made by the administration. A reasonable default value in system configurations might be Organization-Local (e.g., all NPTv6 Translators operated by the organization). However, a large organization might well choose Site-Local or Admin-Local, and
consider that "site" or "administrative" domain to include the set of NPTv6 Translators advertising a default route into a specific part of its network.

6. Security Considerations

The principal security threat in this algorithm is a security threat inherent to IP multicast routing and any application that runs on it. A rogue system can join a multicast group, meaning it that it sees traffic sent to the multicast group that it was presumably not intended to see, and may originate responses that are not correct or infer information. Such a rogue system also in effect pulls traffic toward it, which may not have been planned for in capacity planning. In this scenario, the rogue system has the opportunity to learn the addresses of every system that has such a translation, and has the capability of adding an incorrect External Address to any list of External Addresses. This presents both privacy and security issues.

The obvious mitigation is authentication and authorization of responses returned; requesters should verify that responses are coming from systems that the administration thinks are legitimately NPTv6 Translators. PCP does not define an authentication model, and does not define the use of TLS/DTLS or others. Hence, this likely implies the use of IPSec, or at least a list of the addresses of authorized NPTv6 translators in the network, with administration-specific responses to rogue equipment such as ignoring such responses or some form of remediation. If the multicast routing technology supports it, refusing such rogue "joins" would be a good idea.

In addition, the security considerations in [I-D.ietf-pcp-base] also apply to this use.

7. Acknowledgements

This note resulted a conversation among the authors, Margaret Wasserman, Dave Thaler, and Ron Bonica, and from a separate conversation with Keith Moore.

8. Change Log

Initial Version: January 2012

9. References
9.1. Normative References


9.2. Informative References


Authors’ Addresses

Fred Baker
Cisco Systems
Santa Barbara, California  93117
USA

Email: fred@cisco.com

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

Email: dwing@cisco.com
Some Extensions to Port Control Protocol (PCP)
draft-boucadair-pcp-extensions-02

Abstract

This document extends Port Control Protocol (PCP) with new functionalities.

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 23, 2012.

Copyright Notice

Copyright (c) 2011 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.
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. DESCRIPTION ..................................................... 3
3. DSCP_POLICY ....................................................... 4
4. CAPABILITY ......................................................... 5
5. REPORT ............................................................. 8
6. CLIENT_IDENTIFIER .................................................. 9
7. Security Considerations ............................................ 10
8. IANA Considerations ............................................... 11
9. Normative References .............................................. 11

Authors’ Addresses ................................................... 11
1. Introduction

This document extends the base PCP [I-D.ietf-pcp-base] with various PCP Options.

Some of these options may be defined as new PCP OpCodes.

The main goal of this document is to kick-off discussions on the need to define some useful PCP options which are not part of base PCP.

2. DESCRIPTION

This option (Code TBA, Figure 1) MAY be included in a PCP MAP request to include a description associated with a requested mapping. This option is optional to be supported by PCP Servers and PCP Clients. The maximum length SHOULD be a configurable option 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.

This option can be used by a user to indicate a description associated with a given mapping such as "My mapping for my FTP server" or "My remote access to my CP router", etc. In addition, in some deployment scenarios, this field can be used for troubleshooting purposes and can be used to convey values as the ones listed hereafter:

- "This is the mapping for my specific IPsec implementation"
- "This is the mapping for subscriber bob@example.com"
- "This is the mapping for special subscriber adsl-line-1234@example.com"
- "This is the mapping that failed before due to XYZ"

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

Option Name: Description Option (DESCRIPTION)
Number: TBA (IANA)
Purpose: Used to associate a text description with a mapping
Valid for Opcodes: MAP
Length: Variable (multiple of 4)
May appear in: both request and response
Maximum occurrences: 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| DESCRIPTION   |  Reserved     |           Length              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           Description                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

Figure 1: Description Option

3. DSCP_POLICY

In some scenarios, the DSCP marking in the internal interface (i.e.,
customer-facing interface) and the external one (i.e., Internet-
facing interface) of the PCP-controlled device may be distinct. A
Service Provider MAY allow its customers to configure their DSCP
marking policies in an upstream device. Distinct DSCP marking
policies can be implemented in the internal and external sides of the
PCP-controlled device. A PCP Client MAY issue a PCP MAP request
indicating its internal DS code point and the external DSCP value.
Instructed forwarding policies are applied only for packets marked
with a given DSCP value.

A Service Provider may not support DSCP re-marking feature and adopt
a transparent scheme to QoS policy enforcement, that is, not
controllable by subscribers. Generic QoS enforcement policies can be
enforced for all customers: such as leave DSCP field values
unchanged.

This option is mandatory to process.

This option (Code TBA, Figure 2) allows to:

- Re-write any DSCP value to a specific value;
o Re-write a specific DSCP value to another specific value.

This Option:

Option Name: PCP DSCP Marking Policy Option (DSCP_POLICY)
Number: TBA (IANA); mandatory to process
Purpose: Associated a DSCP re-marking policy with a mapping
Valid for Opcodes: MAP, PEER
Length: 0x04
May appear in: both request and response
Maximum occurrences: 1

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
+---------------------------------------------|
| DSCP_POLICY | Reserved |            0x04               |
+---------------------------------------------|
| DIR| Int DSCP | Ext DSCP | 00...00                   |
+---------------------------------------------|
| +---------------------------------------------|

DIR  : Indicates the direction:
0  Inbound
1  Outbound
2  Both

Int DSCP: Indicates the DSCP value in the customer-faced interface.
0x3F is used to indicate ANY value.

Ext DSCP: Indicates the DSCP value in the Internet-faced interface.
0x3F is used to indicate ANY value.

Figure 2: DSCP Marking option

4. CAPABILITY

The CAPABILITY option (Code: TBA, Figure 3) is used by a PCP Server to indicate to a requesting PCP Client the capabilities it supports with regards to port forwarding operations. Several Capability options MAY be conveyed in the same PCP response message 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.
This Option:

Option Name: PCP Capabilities Option (CAPABILITY)
Number: TBA (IANA)
Purpose: Retrieve the capabilities of a PCP-controlled device
Valid for Opcodes: can be returned in a error message
Length: 0x04
May appear in: both request and response
Maximum occurrences: None

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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| CAPABILITY    |  Reserved     |            0x04               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|F T P A S C I O|              00...00                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

Figure 3: Capability option

Below is provided a description of the F, T, P, A, S, C, I and O bits:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>This bit indicates the address family of the source address issued by internal hosts</td>
</tr>
<tr>
<td>T</td>
<td>This bit indicates the address family of the source address of the packets forwarded in the external side of the PCP-controlled device</td>
</tr>
<tr>
<td>P</td>
<td>This bit indicates whether the source port number is translated or not.</td>
</tr>
<tr>
<td>A</td>
<td>This bit indicates whether the source IP address is translated or not.</td>
</tr>
<tr>
<td>S</td>
<td>This bit indicates whether the controlled device supports the ability to assign a set or ports</td>
</tr>
<tr>
<td>C</td>
<td>This bit indicates whether the PCP-controlled devices inspect the received packets and if it can block them</td>
</tr>
<tr>
<td>I</td>
<td>This bit indicates whether incoming packets are rejected unless an explicit rule is enforced in the PCP-controlled device</td>
</tr>
<tr>
<td>O</td>
<td>This bit indicates whether outbound packets are inspected or not before being granted to leave the internal realm.</td>
</tr>
</tbody>
</table>

The value of the F, T, P, A, S, C, I and O bits are as follows:
<table>
<thead>
<tr>
<th>Position</th>
<th>Name</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>From (F)</td>
<td>0=from IPv4, 1=from IPv6</td>
</tr>
<tr>
<td>2</td>
<td>To (T)</td>
<td>0=to IPv4, 1=to IPv6</td>
</tr>
<tr>
<td>3</td>
<td>Port-Xlate (P)</td>
<td>1=translated, 0=not translated</td>
</tr>
<tr>
<td>4</td>
<td>Addr-Xlate (A)</td>
<td>1=translated, 0=not translated</td>
</tr>
<tr>
<td>5</td>
<td>Port-Set (S)</td>
<td>1=enabled, 0=not supported</td>
</tr>
<tr>
<td>6</td>
<td>Packet-Control (C)</td>
<td>1=enabled, 0=not supported</td>
</tr>
<tr>
<td>7</td>
<td>Direction-Out (I)</td>
<td>1=enabled, 0=disabled</td>
</tr>
<tr>
<td>8</td>
<td>Direction-In (O)</td>
<td>1=enabled, 0=disabled</td>
</tr>
</tbody>
</table>

A stateless NAT64 [RFC6145] would have the following values:

- From=0 (IPv4)
- To=1 (IPv6)
- Port-Xlate=0 (No)
- Addr-Xlate=1 (Yes)
- Port-Set=0 (No)
- Packet-control=0 (No)
- Direction-out=0 (No)
- Direction-In=0 (No)

A stateful NAT64 [RFC6146] would have the following values:

- From=0 (IPv4)
- To=1 (IPv6)
- Port-Xlate=1 (Yes)
- Addr-Xlate=1 (Yes)
- Port-Set=0 (No)
- Packet-control=0 (No)
- Direction-out=0 (No)
- Direction-In=0 (No)

A NAT44 would be characterized as follows:
5. REPORT

The Report PCP Option (Code TBA, Figure 4) is used by a PCP Client to report a set of useful information to the PCP Server. Several Report Options with distinct Report Sub-Code values MAY be conveyed in the same PCP message. Only report data associated with the PCP Server to which this option is sent MUST be included in a Report Option.

This option can be used for troubleshooting or diagnose purposes.

This Option:

- Option Name: PCP Report Option (REPORT)
- Number: TBA (IANA)
- Purpose: Send a set of report data
- Valid for Opcodes: MAP
- Length: Variable
- May appear in: both request and response
- Maximum occurrences: Multiple

```
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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   SCOPE       |  Reserved     |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Report Sub-Code            |          00...00              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           Report Data                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 4: Report Option

The following Report Sub-Code values are defined:
Position | Meaning |
--- | --- |
0x00 | Time since last reboot/boot |
0x01 | Count of transmitted PCP messages to the PCP Server since last boot |
0x02 | Count of retransmitted PCP messages to the PCP Server since last boot |
0x03 | Count of received PCP Error messages from the PCP Server |

6. CLIENT_IDENTIFIER

PCP CLIENT_ID (Code TBA, Figure 5) is a token randomly [RFC4086] generated by the PCP Client. Only one CLIENT_ID Option MUST be present in a PCP message. The PCP Client and PCP Server MUST store the value included in this Option in a PCP MAP request.

- The CLIENT_ID MUST be generated by the PCP Client and not the PCP Server;
- Upon change of the IP address of the PCP Client (or a third party on behalf of which a mapping has been created), the CLIENT_ID is used to update related mappings (e.g., PCP MAP delete request and PCP MAP create request);
- The same CLIENT_ID MUST be used for all requested mappings, unless a new CLIENT_ID is generated by the PCP Client (e.g., reboot, OS crash, etc.); and
- The CLIENT_ID is stored by the PCP Server for all mappings (persistent storage);
This Option:

Option Name: PCP Client Identifier Option (CLIENT_ID)
Number: TBA (IANA); mandatory to process option
Purpose: Associate an identifier with the mappings
Valid for Opcodes: MAP
Length: Variable
May appear in: both request and response
Maximum occurrences: 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| CLIENT_ID     |  Reserved     |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Client Identifier                    |
:                                                               :
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

Figure 5: CLIENT_ID PCP Option

The length of the CLIENT_ID is encoded in the Length field in bytes. The length of the CLIENT_ID MUST be at least 4 bytes and MUST NOT exceed 16 bytes.

The RECOMMENDED value is 16 bytes so as to have a robust random CLIENT_ID. If a CLIENT_ID longer than 16 bytes or shorter than 4 bytes is received, the PCP Server MUST issue a PCP Error message with an error cause equal to "Invalid Client-ID".

For sanity checks, a PCP Server maintains the same CLIENT_ID value (which is used in the latest PCP request) for a given PCP Client for all mappings associated with the same internal IP address belonging to the same subscriber. Indeed, the PCP Server maintains an additional identifier denoted as subscriber-Id. A subscriber-is can be an IP address, IPv6 prefix or a subscriber identifier configured locally.

7. Security Considerations

Security considerations discussed in [I-D.ietf-pcp-base] must be considered. The use of CLIENT_ID option allows to soften issues related to stale mappings.
8. IANA Considerations

The following PCP Option Codes are to be allocated:

**DESCRIPTION**

DSCP_POLICY: The "O" bit MUST be set to 1.

**CAPABILITY**

**REPORT**

CLIENT_IDENTIFIER: The "O" bit MUST be set to 1.

9. Normative References


Authors’ Addresses

Mohamed Boucadair
France Telecom
Rennes, 35000
France

Email: mohamed.boucadair@orange.com
Reinaldo Penno
Juniper Networks
1194 N Mathilda Avenue
Sunnyvale, California 94089
USA

Email: rpenno@juniper.net

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

Email: dwing@cisco.com
Reserving N and N+1 Ports with PCP
draft-boucadair-pcp-rtp-rtcp-03

Abstract

This document defines a new PCP Option to reserve a pair of ports in a PCP-controlled device while preserving the parity and contiguity. This PCP Option eases the NAT traversal for RTP/RTCP flows.

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 20, 2012.

Copyright Notice

Copyright (c) 2011 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 this PCP 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 ............................ 4
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.

Note that the base PCP allows to retrieve the external IP address and port to be conveyed in the SIP signaling messages [RFC3261]. Therefore SIP Proxy Servers can be more lightweight and do not need to support means to ease the NAT traversal of SIP messages (e.g., [RFC5626], [RFC6223], etc.). The 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).

2. Why this PCP 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.

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.
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 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

---

The image contains a paragraph of text related to network protocols and options. The text describes the usage and behavior of the Port Reservation Option (PORT_RESRV_OPT) in the PCP (Packet-Centric Control Plane) protocol. The option is used to retrieve a pair of ports, and it is valid for MAP opcodes. The length of the option is 0, and it may appear in both request and response messages. The option is illustrated in a figure labeled Figure 1: Port Reservation Option (a.k.a., N/N+1 port). The client's behavior when retrieving a pair of ports is also detailed, including the process of checking if the Port Reservation Option is supported by the peer and the handling of RTP and RTCP flows.
procedure defined in Section 9.5 of [I-D.ietf-pcp-base] for each 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/answers for instance.

5. Server Behaviour

Upon receiving the Port Reservation Option, 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 6.3 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.
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.
Figure 3: Flow Example of a PCP Server supporting the option and 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.
Figure 5: Flow example to delete the mappings
7. IANA Considerations

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

- PORT RESERVATION OPTION.

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-14 (work in progress), October 2011.


10.2. Informative References


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
Port Control Protocol (PCP) Proxy Function
draft-bpw-pcp-proxy-01

Abstract

This document specifies the behavior of a PCP Proxy element, for instance embedded in Customer Premise routers.

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 September 15, 2011.

Copyright Notice

Copyright (c) 2011 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. PCP Server Discovery and Provisioning ....................... 4
3. PCP Proxy as a PCP Server ................................. 4
4. Control of the Firewall ..................................... 4
5. Embedded NAT in the CP Router ............................. 5
6. Simple PCP Proxy ........................................... 6
7. Smart Proxy .................................................. 7
   7.1. Multiple PCP Servers .................................. 7
   7.2. Epoch Handling ....................................... 8
   7.3. Request/Response Caching ............................. 8
   7.4. Retransmission Handling ............................... 8
   7.5. Full State ............................................ 9
8. IANA Considerations ......................................... 9
9. Security Considerations ..................................... 9
10. References ................................................ 10
    10.1. Normative References ................................. 10
    10.2. Informative References ............................... 10
Authors’ Addresses .............................................. 11
1. Introduction

PCP [I-D.ietf-pcp-base] discusses the implementation of NAT control features that rely upon Carrier Grade NAT (CGN) devices such as DS-Lite AFTR [I-D.ietf-softwire-dual-stack-lite].

The Customer Premise router, the B4 element in DS-Lite, is in charge to enforce some security controls on PCP requests so implements a PCP Proxy function: it acts as a PCP server receiving PCP requests on internal interfaces, and as a PCP client forwarding accepted PCP requests on an external interface to a CGN PCP server. The CGN PCP server in turn send replies (PCP responses) to the PCP Proxy external interface which are finally forwarded to PCP clients on internal hosts.

[Ed. Note: there is nothing about a requirement for the presence of a PCP Proxy even it is clear that only a PCP Proxy can perform some needed PCP message processing. BTW where should be such a requirement?]

The Proxy can be simple, i.e., implement as transparent/minimal processing as possible, or it can be smart, i.e., handle multiple CGN PCP servers, cache requests/responses, etc. A smart Proxy can be associated with UPnP IGD [I-D.bpw-pcp-upnp-igd-interworking] or/and NAT-PMP [I-D.bpw-pcp-nat-pmp-interworking] Interworking Function (IWF).

```plaintext
+------------+                       |
| PCP Client |-----+                 |
+--(Host 1)--+     |   +-----------+ |     +----------+
     +----------+     |   +-----------+ |     +----------+
| PCP Proxy |-------|PCP Server|
     +----------+     |   +-----------+ |     +----------+
| PCP Client |-----+                 |
+--(Host 2)--+               possible boundary
<- Home side | ISP side ->
```

Figure 1: Reference Architecture

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. PCP Server Discovery and Provisioning

The PCP Proxy MUST implement one of the discovery methods listed in [I-D.ietf-pcp-base] (e.g., DHCP [I-D.bpw-pcp-dhcp]).

[Ed. Note: keep this in phase with the base document.]

The address of the PCP Proxy is provisioned to local PCP Clients as their default PCP Server: If the PCP DHCP option is supported by an internal PCP Client, it will retrieve the PCP Server IP address to use from its local DHCP server (usually embedded on the CP router); otherwise internal PCP Clients will assume their default router being the PCP Server.

[Ed. Note: anything else to add about the PCP Server function of the PCP Proxy?]

3. PCP Proxy as a PCP Server

The PCP Proxy acts as a PCP server for internal hosts and accepts PCP requests on the interface(s) facing them, e.g., it creates servicing socket(s) and bound them to each address of this (these) interface(s) on UDP port 44323.

When the topology makes a routing loop possible, the PCP Proxy MAY check it is not the source of a PCP message it’s received.

4. Control of the Firewall

[Ed. Note: a firewall is usually embedded in the CP router. Even if packets coming from the Internet went first through the ISP NAT, some users still prefer to not disable the local firewall, so the PCP Proxy has to handle this case?]

A security policy to accept PCP messages from the provisioned PCP Server is to be enabled on the CP router. This policy can be for instance triggered by DHCP configuration or by outbound PCP requests issued from the PCP Proxy to the provisioned PCP Server.

In order to accept inbound and outbound traffic associated with PCP mappings instantiated in the upstream PCP Server, appropriate security policies are to be configured on the firewall.

For instance if the firewall rules have a lifetime, PCP response can be snooped in order to instantiate the corresponding firewall rules with the same lifetime. If they have no lifetime, an explicit dynamic mapping table can be kept in the PCP Proxy state in order to instantiate and remove corresponding firewall rules. This is in fact...
an easy sub-case of Section 5.

REMOTE_PEER_FILTER Options can be installed into the local firewall, forwarded to the PCP Server so installed into the remote NAT/firewall or both.

[Ed. Note: should we say the firewall function is already handled by the PCP controlled device so it is useless at the local level?]

5. Embedded NAT in the CP Router

When no NAT is embedded in the CP router, the port number included in received PCP messages (from the PCP Server or PCP Client(s)) are not altered by the PCP Proxy.

[Ed. Note: NAT444 seems to be the only exception?]

When the PCP Proxy is co-located with a NAT function in the CP router, it MUST update the content of received requested messages with the mapped port number and the address belonging to the external interface of the CP router (i.e., after the NAT operation) and not as initially positioned by the PCP Client. For the reverse path, PCP response messages MUST be updated by the PCP Proxy to replace the target port number to what has been initially positioned by the PCP Client. For this purpose the PCP Proxy has an access to the local NAT state. Note PCP messages with an unknown OpCode or Option can carry a hidden target address or internal port which will not be translated:

- a PCP Proxy co-located with a NAT SHOULD reject by an UNSUPP_OPCODE error response a received request with an unknown OpCode;
- a PCP Proxy co-located with a NAT SHOULD reject by an UNSUPP_OPTION error response a received request with a mandatory-to-process unknown Option;
- a PCP Proxy co-located with a NAT SHOULD remove any optional-to-process unknown Options from received requests before forwarding them.

When a PCP request is received and accepted by the PCP Proxy the corresponding mapping (explicit dynamic mapping for a MAP request, implicit dynamic mapping for a PEER request) is looked for in the local NAT state and temporary created if it does not exist. Temporary means it is deleted if no SUCCESS response is received, either explicitly or because of its short lifetime at creation.

If the local NAT associates explicit dynamic mappings to a lifetime, the requested lifetime in MAP requests SHOULD be adjusted to be in the accepted range of the local NAT, and the assigned lifetime copied
from MAP responses to the corresponding mapping in the local NAT. The same processing applies to implicit dynamic mappings and PEER requests/responses (but the valid requested lifetime range begins by zero in this case).

Otherwise explicit dynamic mappings have an undefined lifetime in the local NAT and the PCP Proxy SHOULD maintain an explicit dynamic mapping table and SHOULD delete corresponding explicit dynamic mappings in the local NAT when they expire or are deleted by the MAP request with a zero requested lifetime.

6. Simple PCP Proxy

A simple PCP Proxy performs minimal modifications to PCP requests and responses, in particular it does not change the Epoch value in responses. So it does not handle more than one PCP server.

The detailed behavior at the reception of a PCP request on an internal interface is as follows:
- check if the source IP address and the PCP target address are the same.
- apply security controls, including with the result of the previous item.
- if the request is rejected, build a synthetic error response and send it back to the PCP client.
- if the request is accepted, adjust it (e.g., adding a THIRD_PARTY Option, updating the internal address and port to their translated values as specified in Section 5) and forward it on a fresh UDP socket connected to the PCP server.
- Wait for the response during a reasonable delay.
- when the response is received from the PCP server, adjust it back (e.g., removing the THIRD_PARTY Option added previously, updating the internal address and port to their initial values as specified in Section 5), forward it to the source PCP client and close the socket to the PCP server.
  [Ed. Note: is there extra validation useful? The response comes from the PCP server and the PCP client will validated it anyway.]
- on a hard error on the UDP socket, build a synthetic ICMP error and send it to the source PCP client.

The reasonable delay minimum value is 20 seconds, request retransmission is handled by PCP clients.

For each pending request, the proxy MUST maintain in a data record:
- the request payload
- the interface where the request was received
- the source IP address of the request
- the source UDP port of the request
- the UDP socket connected to the PCP server
- an expire timeout

Receiving interfaces can be implemented by a set of servicing sockets, each socket bound to an address of an internal interface. Interface, source address and port are used to send back packets to the source PCP client. The request payload is used to generate synthetic ICMP. Responses are received on the UDP socket.

There is no (not yet) standardized way to build a synthetic error response, in particular no way to determine which Epoch value to put into it. This is why it is better to build a synthetic ICMP error than a synthetic error response with NETWORK_FAILURE on a socket hard error.

[Ed. Note: a specific document is supposed to solve this point and all bootstrap related issues.]

Too large requests SHOULD be forwarded to the PCP server in order to relay back the error response, i.e., the PCP Proxy is not in charge to enforce the message size limit and in general the PCP Proxy SHOULD NOT generate error response for a reason other than security controls. No behavior is specified in the case the PCP Proxy processing (e.g., adding a THIRD_PARTY Option) makes a valid request too large when it is sent to the PCP Server.

7. Smart Proxy

When a simple PCP Proxy uses as global variables only the CGN PCP server IP address, a set of servicing sockets and a list of pending request handlers, a smart PCP Proxy implements more services.

Even if most services rely on the Epoch handling one Section 7.2, services are described below in a natural order.

7.1. Multiple PCP Servers

A smart PCP Proxy MAY offer to handle multiple PCP servers at the same time, each PCP server is associated to each own handled Epoch value according to Section 7.2.

The only constraint is to maintain a reasonable coherency as PCP clients cannot be assumed to be prepared to this, i.e., this has to be transparent for / hidden to them.
7.2. Epoch Handling

With Epoch handling the Epoch value is related to internal timers and not blindly copied from PPC responses. There should be no advantages to have more than one managed Epoch per PCP server.

The Epoch MUST be reset when explicit dynamic mappings are lost, i.e.:
- at startup if the PCP proxy can’t recover the state.
  [Ed. Note: as it is very optional to manage state in the Proxy it should be the default.]
- when the WAN address is changed or any similar events which show any previous state is no longer valid.
- when the Epoch value in a PCP response is too small (cf. Epoch value validation rules in [I-D.ietf-pcp-base]).
- when the External Address has changed.

The last two rules are per PCP server, a PCP Proxy MAY check these conditions in all received responses for a PCP server, including when the PCP Proxy is a part of an IWF [I-D.bpw-pcp-upnp-igd-interworking] [I-D.bpw-pcp-nat-pmp-interworking].

7.3. Request/Response Caching

A PCP Proxy providing request/response caching checks each time it receives a PCP request if it has already seen the same request recently and got the corresponding PCP response. In this case, it sends back directly the cached response with the proper Epoch value and not forward the request to the PCP server.

[Ed. Note: this is an easy optimization, the only difficult point can be solved by the Epoch handling.]

7.4. Retransmission Handling

An extension of the previous service is to manage the retransmission of pending requests to the server internally, i.e., no longer driven by the PCP client. A cache entry SHOULD be expired after a delay short enough to keep it easy to distinguish it from a replay.

[Ed. Note: this allows smart retransmission scheduling as the Proxy "sees" all PCP exchanges with the PCP server.]
7.5. Full State

A smart PCP Proxy can keep the full state: an image of all active explicit dynamic mappings is kept in memory. This service is not interesting by itself but it can be necessary to support embedded firewall or NAT Section 5 and if the PCP Proxy is integrated in an IWF (e.g., to support UPnP IGD [I-D.bpw-pcp-upnp-igd-interworking]).

In conclusion this service MAY be supported. Note when it is supported the state SHOULD be recovered in case of failures according to [I-D.boucadair-pcp-failure].

8. IANA Considerations

This document makes no request of IANA.

Note to RFC Editor: this section may be removed on publication as an RFC.

9. Security Considerations

The security controls are applied on PCP requests and are about:
- authorized target addresses, in particular in case of a third party.
- authorized internal and external ports (note the external port is in general assigned by the CGN PCP server).

The default policy for requests for a third party when such a policy exists is to not allow them. The exact rule is: PCP requests including a THIRD_PARTY option enclosing an IP address distinct than the source IP address of the request MUST be rejected (by a NOT_AUTHORIZED error response).

When a PCP Proxy is at the boundary of two trust domains (named "internal" and "external" sides), it MUST provide at least these two security controls:
- split horizon anti-spoofing: requests from the external side and responses from the internal side MUST be dropped.
- a policy about requests on the behalf of a third party MUST be enforced.

A PCP Proxy MAY implement only the simple rule about third party: all received requests including a THIRD_PARTY option are rejected.

[Ed. Note: this is stricter than the default but keeps the minimal implementation as simple as possible.]

A received request carrying an unknown OpCode or Option SHOULD be dropped (or in the case of an unknown Option which is not mandatory-
to-process the Option be removed) if it is not a priori compatible with security controls or correct processing. This includes at least all cases where received requests are scanned for elements like the protocol, an address or a port.

[Ed. Note: magically a minimal implementation in favorable environments (no embedded NAT!) MAY accept unknown OpCodes and Options. There is no need for a similar rule for responses as the proxy can do nothing with a "bad" response anyway...]

10. References

10.1. Normative References

[I-D.bpw-pcp-dhcp]
Boucadair, M., Penno, R., and D. Wing, "DHCP and DHCPv6 Options for Port Control Protocol (PCP)", draft-bpw-pcp-dhcp-03 (work in progress), March 2011.

[I-D.ietf-pcp-base]


10.2. Informative References

[I-D.boucadair-pcp-failure]

[I-D.bpw-pcp-nat-pmp-interworking]

[I-D.bpw-pcp-upnp-igd-interworking]

[I-D.ietf-softwire-dual-stack-lite]

Authors' Addresses

Mohamed Boucadair
France Telecom
Rennes 35000
France

Email: mohamed.boucadair@orange-ftgroup.com

Reinaldo Penno
Juniper Networks
1194 N Mathilda Avenue
Sunnyvale, California 94089
USA

Email: rpenno@juniper.net

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

Email: dwing@cisco.com

Francis Dupont
Internet Systems Consortium

Email: fdupont@isc.org
Abstract

This memo provided two cases study, when PCP is deployed in mobile network. Some motivation of deployment PCP in mobile network was described and justification for each deployment case was stated. Corresponding to each mode, the document discussed some features to address operational issues. That might potentially cause some extension on PCP protocol level.

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 September 6, 2012.

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
Table of Contents

1. Introduction .................................................. 3
2. Problem Statements ........................................... 3
3. Case 1: PCP Client located on UE ............................ 5
   3.1. Deployment Description .................................. 5
   3.2. PCP API Design .......................................... 6
   3.3. Authentication Consideration ......................... 6
4. Case 2: PCP Proxy located on MGW ............................. 7
   4.1. Deployment Description .................................. 7
   4.2. PCP Whitelist/Blacklist Design ......................... 8
   4.3. Authentication Consideration ......................... 8
   4.4. PCP Policy Implementation ............................ 8
5. Conclusion ....................................................... 9
6. Security Considerations ...................................... 9
7. IANA Considerations .......................................... 9
8. Acknowledgements ............................................. 9
9. References ..................................................... 10
   9.1. Normative References .................................... 10
   9.2. Informative References ................................ 10
Authors' Addresses ............................................... 10
1. Introduction

This document provides a PCP deployment case study, which reflects demands from optimization on application traffic in mobile environment. Traffic in a mobile network is becoming a complex mix from protocols, different application and user behaviors. Some applications offer always-on characteristics to users, like instant message applications. These kinds of applications are normally TCP/IP-based. And, application flows require one long-lived TCP connection between clients and servers. The long-lived TCP connection has impacts on normal network behavior. Issues such as network resources and terminals battery consumption happened when there are frequent keepalive/notification message has been transmitted periodically. In order to mitigate the issues, PCP[I-D.ietf-pcp-base] could be a way to resolve the problems by managing how incoming packets are forwarded by upstream devices such as NAT64, NAT44, IPv6 and IPv4 firewall devices.

The applicabilities for reducing keepalive messages have been described in PCP protocol, in which such optimization is one of use cases from protocol design purpose. This memo would elaborate deployment from operational planning perspectives. The intention of publishing the memo would like to sharing the thoughts for potential usages in mobile networks. The audiences could get the information for their possible deployment of NAT64 in future. Some new works required to PCP protocol might be derived from particular considerations. These works are expected to be documented in separated memo.

2. Problem Statements

Understanding traffic features is essential for network protocol optimization and quality improvement. During the Instant services are deployed, keepalive transmitted from a client to servers is to prevent inactivity from disconnecting the channel. The scenario is depicted in Figure 1.

```
+-----------------+           +-----------------+
|                 |           |                 |
|                 |           |                 |
|                 |           |                 |
MN                |             | Internet        |
|     +-----+   | +-----+     | +-----+         |
| MN          |       | RAN       |       | APP Server |
|     +-----+   | +-----+     | +-----+         |
| MN: Mobile Terminals |
| RAN: Radio Access Network |
| PS: Packet Switch |
| MGW/FW: Mobile Gateway/Firewall |
```
Radio access network would provide wireless connectivity to the MN. Packages are transmitted through Packet Switch Domain heading to MGW. MGW would bear the responsibilities of Address allocation, Routing and Transfer. The connection between MN and MGW normally is point-to-point link, on which MGW is default router for MN. Firewall could either be integrated with MGW or posited behind MGW as standalone. The traffic is finally destined to application servers, which manage subscriber information.

The behaviors of keepalives messages would let several nodes to keep track of all connections that pass through them. Depending on frequent keepalive exchanges, following states remain alive.

- States on FW/NAT: NAT mapping records for UDP/TCP would be refreshed before the information becoming staled.
- States on MGW: IP allocation performed by PDP(Packet Data Protocol) process. GW reserve states of allocated IP address by monitoring passed packages. Keepalive messages would also make the PDP context is alive.
- States on APPs servers: applications server manage online/offline states for subscriber. The transmission of keepalive stands for active users. Otherwise, application servers advertise offline information to online subscribers inform the statues changes.

The maintenance on these states keeping is irreplaceable in some extents, because those heartbeat exchanges is a only way to hold these together. However, these frequent and short messages have side-effects on mobile networks.

- Radio resource consumption. A dedicated air channel needs to be assigned to each keepalive message. Radio resource utilization is very low in the case. Significant keepalive messages might even led air resource depletion.
- Terminal energy consumption. Mobile terminals are often "sleep" to extend battery life. Heartbeat message would prohibit such state changing to idle so as to oppose to energy saving.
- Operational profits consumption. According to a statistics, 16% instant signalling message would consume 50%~70% radio resource. The traffic mode would break balance of operational payments.

The PCP deployment cases proposed by this memo are trying to resolve above-mentioned issues. Meanwhile, the deployment architectures take
3. Case 1: PCP Client located on UE

3.1. Deployment Description

The Figure 2 depicted the deployment scenario, in which PCP client is installed on ME. PCP server is located with NAT/FW respectively. Available IP address/Port would be assigned to PCP client through PCP MAP/PEER signalling negotiation. Therefore, NAT binding states would be manipulated in an explicit way to reduce NAT keepalive frequency. PCP signalling could guarantee NAT states alive on Firewall. Regarding states on MGW and APP servers, additional works should be done to fit into a mobile architecture.

First off, APP server should track users states and synchronize online/offline information. Cancellation of keepalive message would led user states invisible. In PCP protocol spec[I-D.ietf-pcp-base] section 9.3, there is statement that PCP cannot do anything useful to reduce those keepalives. This memo recommended a way to optimize the application by integrating PCP capacities with always online application processes. To achieve that, PCP API design works should be done to provide application clients with explicit NAT binding information. After the awareness of session active duration, app clients could inform app servers the active behavior by extending app signalling. Such information could prolong time-out timers on the server side. The user states would align with NAT binding information.

Regarding the requirement of maintaining status on MGW, it could take the Policy and Charging Control (PCC) advantage to execute a policy for specific PDP session. The Figure 2 also shown the link on such reaction chain, in which Policy And Charging Rules Function (PCRF)
and Application Function (AF) are involved. Therein PCRF is responsible for authorizing and executing policy rules on MGW. AF is a functional element offering applications that require dynamic policy. The AF can be seen as bridging the gap between applications and how they affect node behaviour. The whole process is to let AF aware of PCP warranted time-slot and feedback such information to PCRF. Afterwards, PCRF could apply the policy to MGW for IP address maintenance. It should be noted that this part of work is out of the PCP work scope.

Some particular features have been proposed in the case 1 to optimize operational experiences, which was described in following sub-sections.

3.2. PCP API Design

In the case of PCP client located on a host (e.g. Mobile terminals) with various applications, it’s desirable to package PCP functionalities as a capacity open to upper layer applications. That is not only to facilitate the coordination between PCP caller(i.e. applications) and client, but also help to optimize functionalities both for PCP and applications. The basic roles of PCP API is to provide applications with functions of triggering PCP requests and feedback PCP responds to apps. So, the application developers could code PCP as one of components to coordinate with other part of applications. For example, when the application is aware of reserved binding information on NAT, the client could report to servers to optimize heartbeat at application layer. Another benefit introducing PCP API is to hand failure cases, when a application or PCP client is broken accidentally. PCP API could eject an exception inform PCP server to delete related port binding in time depending on system calls.

3.3. Authentication Consideration

It’s hard to determine whether PCP requests is coming from registered users or malicious attacker. Since operator don’t know the situations in wild. The authentication is necessary to defend PCP server. Some proper authentication mechanism fits into the mobile network.

The problem of PCP authentication comes from the fact that the PCP client (device) and PCP server (FW) do not have trust relationship with each other. [I-D.wasserman-pcp-authentication] has some considerations on the PCP authentication, in which an EAP option is included in the PCP requests from the devices. In mobile network, provisioning of new credentials to mobile devices is a difficult tasks. Taking this into consideration, the integration with SIM
authentication is one of these choices on the table. The other possible ways of PCP authentication include the use of open authentication capability such as 3GPP GAA (Generic Authentication Architecture) defined in 3GPP 33.220. So that, the PCP client can invoke the authentication ability provided by the operator.

4. Case 2: PCP Proxy located on MGW

4.1. Deployment Description

Figure 3 depicted another deployment scenario, in which PCP client is installed on MGW and PCP server is located on NAT/FW. MGW takes role of third party on behalf of MEs to initiate PCP signalling. THIRD_PARTY Option should be carried in MAP and PEER opcodes to control a mapping for an ME. The applicabilities of THIRD_PARTY Option is justified by following reasons.

- MGW and NAT/FW belong to "wallet garden" mode from operational perspectives. MGW could be treated as an operator-authorized node, which would facilitate the authorization process.

- MGW is capable of verifying the present/absence of delegated MNs by monitoring MN states(e.g. sending Routing Area Update(RAU)/Track Area Update(TAU) periodically). It could get rid of risks maintaining immortalized mappings even an ME has gone.

The case also leveraged PCC framework, in which PCRF servers as a policy enforcement point. It could keep statues information on behalf of ME and assign policies to MGW. Such deployment consideration also benefits to maintaining PDP session states on MGW. PDP session could be informed with binding period records through internal state machine. And it would take corresponding action to hold the PDP session for an identical period. On the other hand,
maintained time interval is sent to PCRF and forward to AF. APP server would be offered with a specific online character for user status management.

Some particular features have been proposed in the case 2 to optimize operational experiences, which was described in following subsections.

4.2. PCP Whitelist/Blacklist Design

PCP whitelist/blacklist feature is to prioritize incoming PCP requests message according to configured list. Using a "whitelist" in a generic sense means that PCP requests are not treated as high priority unless the source IP address of PCP request is contained in the whitelist. In contrast, using a "blacklist" means that all PCP requests are permitted to maintain IP/Port resource unless the source IP address is contained in the blacklist.

The whitelist is comprised of several addresses and configured on PCP servers manually. These whitelist configuration contribute to differentiated service purposes. In the case of scarcity of IP/Port resources, PCP server would offer high priority to PCP requests, the packet source address of which is matched with a specific record in the list. Whitelist usage would potentially promote PCP requests sending from an operators trusted node, e.g. MGW, etc.

A blacklist could protect PCP server from overloaded PCP requests process. A range of IP address subnet could be set in blacklist to rule out malicious attacks. When a PCP server receives a PCP request with source address falling into the blacklist, it MUST generate a error response.

4.3. Authentication Consideration

Basically, the PCP requests from MGW will change the TCP/UDP binding for the MN. But considering the MGW is a trusted entity in the 3GPP domain, so the security consideration in this case could be relaxed to some extent. The MGW should make sure that the change of binding relationship on the FW.

4.4. PCP Policy Implementation

PCP policy features provide a possibility for more fine-grained states management. Different PCP requests would be ranked to perform differentiated policies according to specific service features. In some cases, Service Level Agreement (SLA) would be signed to guarantee service delivery with QoS assurance. Such feature is useful to assign proper resource for each PCP request, which
corresponding different kinds of services.

The implementation of PCP policy should configure a policy /knowledge base on PCP server. Such policy configuration is made up with maximum resource guarantee represented with different indicator. Operators could treat PCP requests differently according to service categories, e.g. Gold service would admit to have unlimited ports resource and Silver service is only permitted to occupy pre-determinative ports (both on quantity and hold-up time).

Correspondingly, PCP requests should be marked with different service tags. Such tags would be labeled depending on DPI functionalities or policy delivery from PCRF. In the case, THIRD_PARTY Option should extend a new field to indicate the service tag. The extension could be done in separated document is this requirement is accepted.

5. Conclusion

PCP mechanism could be potentially adopted in different usage contexts. The deployment document could give audiences a explicit use case. Operators may benefit from such experiences sharing. It even becomes a guidance for future PCP deployment. There is recommendation to PCP WG to validate the inputs and indentify issues from operational aspects. It is expected that proper document should be drafted to determine solutions or workarounds to those issues.

6. Security Considerations

TBD

7. IANA Considerations

This document makes no request of IANA.

8. Acknowledgements

The authors would like to thank Ping Lin for her discussion and comments.

9. References
9.1. Normative References

[I-D.ietf-pcp-base]
Cheshire, S., Boucadair, M., Selkirk, P., Wing, D., and R. Penno, "Port Control Protocol (PCP)",
draft-ietf-pcp-base-23 (work in progress), February 2012.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate

9.2. Informative References

[I-D.cao-pcp-mobapp-online]
Cao, Z., Chen, G., Deng, H., and T. Sun, "Requirements for
Always Online Applications",
draft-cao-pcp-mobapp-online-00 (work in progress),
March 2011.

[I-D.wasserman-pcp-authentication]
Zhang, D., Wasserman, M., and S. Hartman, "Port Control
Protocol (PCP) Authentication Mechanism",
draft-wasserman-pcp-authentication-01 (work in progress),
October 2011.

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
The Port Control Protocol in Dual-Stack Lite environments
draft-dupont-pcp-dslite-01

Abstract

This document specifies the so-called "plain mode" for the use of the Port Control Protocol (PCP) in Dual-Stack Lite (DS-Lite) environments.

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 2, 2012.

Copyright Notice

Copyright (c) 2011 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
1. Introduction

The Dual-Stack Lite (DS-Lite, [RFC6333]) is a technology which enables a broadband service provider to share IPv4 addresses among customers by combining two well-known technologies: IP in IP (IPv4-in-IPv6) and Network Address Translation (NAT).

Typically, the home gateway embeds a Basic Bridging BroadBand (B4) capability that encapsulates IPv4 traffic into a IPv6 tunnel to the carrier-grade NAT, named the Address Family Transition Router (AFTR). AFTRs are run by service providers.

The Port Control Protocol (PCP, [I-D.ietf-pcp-base]) allows customer applications to create mappings in a NAT for new inbound communications destined to machines located behind a NAT. In a DS-Lite environment, PCP servers control AFTR devices.

Two different modes of operations were proposed: the plain and the encapsulation modes. This document selects the plain mode as the one to use.

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. Plain Mode

In the plain mode the B4, the customer end-point of the DS-Lite IPv6 tunnel, implements a PCP proxy ([I-D.bpw-pcp-proxy]) function and uses UDP over IPv6 with the AFTR to send PCP requests and receive PCP responses.

The B4 MUST source PCP requests with the IPv6 address of its DS-Lite tunnel end-point and MUST use a THIRD PARTY option either empty or carrying the IPv4 internal address of the mappings.

In the plain mode the PCP discovery ([I-D.ietf-pcp-base] section 7.1 "General PCP Client: Generating a Request") is changed into:
1. if a PCP server is configured (e.g., in a configuration file or via DHCPv6), that single configuration source is used as the list of PCP Server(s), else;
2. use the IPv6 address of the AFTR.

To summary: the first rule remains the same with the precision that DHCP is DHCPv6, in the second rule the default router list is...
replaced by the AFTR.

3. IANA Considerations

This document makes no request of IANA.

Note to RFC Editor: this section may be removed on publication as an RFC.

4. Security Considerations

The plain mode provides a control point inside the home network where any policy on PCP requests can be applied, e.g.:
- restrict the use of THIRD PARTY options to the B4
- apply an access-list on internal addresses and/or ports

At the opposite the encapsulation mode Appendix A by default is fully transparent for the B4: PCP requests are blindly encapsulated as any other IPv4 packets to the Internet. So to apply a policy on them requires heavier and far less flexible tools.

5. Acknowledgments

Reinaldo Penno who checks the validity of the argument about the relative complexity of the encapsulation mode at the AFTR side.

Christian Jacquenet and Mohammed Boucadair who proposed improvements to the document, including the PCP server discovery by Mohammed.

6. References

6.1. Normative References

[I-D.ietf-pcp-base]


6.2. Informative References

[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.bpw-pcp-upnp-igd-interworking]
Boucadair, M., Penno, R., Wing, D., and F. Dupont, "Universal Plug and Play (UPnP) Internet Gateway Device (IGD)-Port Control Protocol (PCP) Interworking Function",
draft-bpw-pcp-upnp-igd-interworking-02 (work in progress), February 2011.

Appendix A. Encapsulation Mode

The encapsulation mode deals at the B4 side with PCP traffic as any IPv4 traffic: it is encapsulated to and decapsulated from the AFTR over the DS-Lite IPv4 over IPv6 tunnel.

At the AFTR side things are a bit more complex because the PCP server needs the context, here the source IPv6 address, for both to manage mappings and to send back response. So the AFTR MUST tag PCP requests with the source IPv6 address after decapsulation and before forwarding them to the PCP server, and use the same tag to encapsulate PCP responses to correct B4s. (the term "tag" is used to describe the private convention between the AFTR and the PCP server).

Appendix B. Justification

We believe most customers will run a PCP proxy on the B4 because:
- they want a control point where to apply security (Section 4)
- they run an InterWorking Function (IWF) for other protocols ([I-D.bpw-pcp-upnp-igd-interworking]) on the B4 so the proxy is just part of a bigger system.

BTW when the home network has only one node (dual-stack capable with embedded B4 element) attached, it is the PCP client.

For a PCP proxy to use IPv4 (encapsulation mode) or IPv6 (plain mode) does not make a sensible difference, so from an implementation point of view the real difference is on the PCP server / AFTR side: the encapsulation mode require an Application Level Gateway (ALG) to tag PCP request with the corresponding customer after decapsulation, when the plain mode is fully transparent.
Authors’ Addresses

Francis Dupont
Internet Systems Consortium

Email: fdupont@isc.org

Tina Tsou
Huawei Technologies
2330 Central Expressway
Santa Clara
USA

Phone: +1-408-330-4424
Email: tina.tsou.zouting@huawei.com

Jacni Qin
ZTE Corporation
Shanghai
P.R.China

Email: jacni@jacni.com
Port Control Protocol (PCP)
draft-ietf-pcp-base-24

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 September 13, 2012.

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.
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 .... 10
5. Note on Fixed-Size Addresses ........................... 11
6. Protocol Design Note .................................... 12
7. Common Request and Response Header Format ................. 13
   7.1. Request Header .................................... 14
   7.2. Response Header ................................... 15
   7.3. Options ........................................... 16
   7.4. Result Codes ...................................... 19
8. General PCP Operation .................................. 20
   8.1. General PCP Client: Generating a Request .......... 20
   8.1.1. PCP Client Retransmission ...................... 21
   8.2. General PCP Server: Processing a Request .......... 23
   8.3. General PCP Client: Processing a Response ......... 25
   8.4. Multi-Interface Issues ............................ 26
   8.5. Epoch ............................................. 26
9. Version Negotiation ..................................... 28
10. Introduction to MAP and PEER Opcodes .................... 29
   10.1. For Operating a Server ............................ 31
   10.2. For Operating a Symmetric Client/Server .......... 33
   10.3. For Reducing NAT or Firewall Keepalive Messages .. 35
   10.4. For Restoring Lost Implicit TCP Dynamic Mapping State .. 36
11. MAP Opcode ........................................... 37
   11.1. MAP Operation Packet Formats ..................... 38
   11.2. Generating a MAP Request ......................... 40
   11.2.1. Renewing a Mapping ............................ 41
   11.3. Processing a MAP Request ......................... 41
   11.4. Processing a MAP Response ....................... 44
   11.5. Mapping Lifetime and Deletion .................... 44
   11.6. Address Change Events ............................ 47
   11.7. Learning the External IP Address Alone .......... 48
12. PEER Opcode ........................................... 48
   12.1. PEER Operation ................................... 49
12.2. Generating a PEER Request .................................... 52
12.3. Processing a PEER Request .................................... 52
12.4. Processing a PEER Response .................................... 53
13. Options for MAP and PEER Opcodes ................................. 54
13.1. THIRD_PARTY Option for MAP and PEER Opcodes ............... 54
13.2. PREFER_FAILURE Option for MAP Opcode ....................... 57
13.3. FILTER Option for MAP Opcode ................................ 58
14. Rapid Recovery .................................................... 60
14.1. ANNOUNCE Opcode ............................................ 61
14.1.1. ANNOUNCE Operation .................................... 61
14.1.2. Generating and Processing a Solicited ANNOUNCE Message ........................................ 62
14.1.3. Generating and Processing an Unsolicited ANNOUNCE Message ........................................ 62
14.2. PCP Mapping Update ............................................ 64
15. Implementation Considerations .................................... 65
15.1. Implementing MAP with EDM port-mapping NAT ................ 65
15.2. Lifetime of Explicit and Implicit Dynamic Mappings ........ 66
15.3. PCP Failure Scenarios ........................................ 66
15.3.1. Recreating Mappings ..................................... 66
15.3.2. Maintaining Mappings .................................... 67
15.3.3. SCTP .................................................... 68
15.4. Source Address Replicated in PCP Header ...................... 68
15.5. State Diagram ................................................ 69
16. Deployment Considerations ....................................... 70
16.1. Ingress Filtering ............................................ 70
16.2. Mapping Quota ............................................... 71
17. Security Considerations ......................................... 71
17.1. Simple Threat Model ......................................... 71
17.1.1. Attacks Considered ...................................... 72
17.1.2. Deployment Examples Supporting the Simple Threat Model ........................................ 73
17.2. Advanced Threat Model ....................................... 73
17.3. Residual Threats ............................................. 74
17.3.1. Denial of Service ....................................... 74
17.3.2. Ingress Filtering ....................................... 75
17.3.3. Mapping Theft .......................................... 75
17.3.4. Attacks Against Server Discovery ......................... 75
18. IANA Considerations ............................................. 75
18.1. Port Number ................................................ 75
18.2. Opcodes .................................................... 76
18.3. Result Codes ............................................... 76
18.4. Options .................................................... 76
19. Acknowledgments ................................................ 77
20. References ..................................................... 77
20.1. Normative References ...................................... 77
20.2. Informative References ..................................... 78
Appendix A. NAT-PMP Transition ..................................... 80
Appendix B.  Change History  . . . . . . . . . . . . . . . . . . . . 81
B.1.  Changes from draft-ietf-pcp-base-23 to -24  . . . . . . 81
B.2.  Changes from draft-ietf-pcp-base-22 to -23  . . . . . 83
B.3.  Changes from draft-ietf-pcp-base-21 to -22  . . . . 85
B.5.  Changes from draft-ietf-pcp-base-19 to -20  . . . . 85
B.7.  Changes from draft-ietf-pcp-base-17 to -18  . . . . 86
B.8.  Changes from draft-ietf-pcp-base-16 to -17  . . . . 86
B.9.  Changes from draft-ietf-pcp-base-15 to -16  . . . . 86
B.10. Changes from draft-ietf-pcp-base-14 to -15  . . . . 87
B.11. Changes from draft-ietf-pcp-base-13 to -14  . . . . 87
B.12. Changes from draft-ietf-pcp-base-12 to -13  . . . . 88
B.13. Changes from draft-ietf-pcp-base-11 to -12  . . . . 89
B.14. Changes from draft-ietf-pcp-base-10 to -11  . . . . 89
B.15. Changes from draft-ietf-pcp-base-09 to -10  . . . . 89
B.16. Changes from draft-ietf-pcp-base-08 to -09  . . . . 89
B.17. Changes from draft-ietf-pcp-base-07 to -08  . . . . 90
B.18. Changes from draft-ietf-pcp-base-06 to -07  . . . . 91
B.19. Changes from draft-ietf-pcp-base-05 to -06  . . . . 92
B.20. Changes from draft-ietf-pcp-base-04 to -05  . . . . 94
B.21. Changes from draft-ietf-pcp-base-03 to -04  . . . . 94
B.22. Changes from draft-ietf-pcp-base-02 to -03  . . . . 95
B.23. Changes from draft-ietf-pcp-base-01 to -02  . . . . 95
B.24. Changes from draft-ietf-pcp-base-00 to -01  . . . . 96
Authors’ Addresses  . . . . . . . . . . . . . . . . . . . . . . . . . . . . 96
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 Premise 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 webcam) 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 and port to an internal IP address 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 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:

- Basic NAT [RFC3022]
- Network Address and Port Translation [RFC3022], such as commonly deployed in residential NAT devices
- Carrier-Grade NAT [I-D.ietf-behave-lsn-requirements]
- Dual-Stack Lite (DS-Lite) [RFC6333]
- Layer-2 Aware NAT [I-D.miles-behave-l2nat]
- Dual-Stack Extra Lite [I-D.arkko-dual-stack-extra-lite]
- NAT64, both Stateless [RFC6145] and Stateful [RFC6146]
- IPv4 and IPv6 simple firewall control [RFC6092]
- 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 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) arrives at the host, the outbound response (e.g., TCP SYNACK) has to go through the same path so it is seen by the firewall or rewritten by the NAT. 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 have the necessary 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 (see
Section 6 of [RFC4787] for definition of hairpin).

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 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].
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 and port, and vice versa. In the case of a pure firewall, the "Mapping" is the identity function, translating an internal IP address and port number to the same external IP address and port number. Firewall filtering, applied in addition to that identity mapping function, is separate from the mapping itself.

Mapping Types:
There are two dimensions to classifying mapping types: how they are created (implicitly/explicitly) and their primary purpose (outbound/inbound).

* 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.

* 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, as with a DHCP address lease, if a 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 a new PCP request).

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) it can be helpful when talking about mappings 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. Unlike some other NAT and firewall control
applications which have to be embedded in the underlying operating
system or framework, 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 large-scale NAT 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.
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 device supports PCP and wants to receive traffic, and does not possess knowledge of such filtering, it SHOULD use PCP to create 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 IPv4-mapped IPv6 address [RFC4291] would not be used for mappings.

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
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:

- 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
- 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 (i) if the client has not yet received a response from the server for a prior request (retransmission), and (ii) if 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, at their discretion, 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 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 have its security 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 length of 1024 bytes. The PCP messages contain a request or response header containing an Opcode, any relevant Opcode-specific information, and zero or more Options. 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Version = 1 |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 1. PCP clients and servers compliant with this document use the value 1. 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 0 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 on-path NAT, see Section 8.3.

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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Version = 1  |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 MUST use version 1. 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:

```
+-----------------+-----------------+-----------------+-----------------+
<table>
<thead>
<tr>
<th>Option Code</th>
<th>Reserved</th>
<th>Option Length</th>
</tr>
</thead>
</table>
+----------------+-----------------+-----------------------+
: (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 1024-byte size limit; if its reply would exceed that size, the server generates an error.
If, while processing an Option, 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.,
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 an Option cannot successfully be parsed, the PCP server MUST
generate an error response with code MALFORMED_OPTION. The most
significant bit in the Option Code indicates if its processing is
optional or mandatory. 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 UNSUPP_OPTION if the Option is unrecognized,
unimplemented, or disabled, or if the client is not authorized to use
the Option. All processed options are included in successful
responses, but unprocessed options are not included in successful
responses. All options are returned in error responses.

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:

- The THIRD_PARTY Option is valid for both MAP and PEER Opcodes.
- The FILTER Option is valid only for the MAP Opcode (for the PEER
  Opcode it would have no meaning).
- 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: Unsupported protocol version. This is a long lifetime error.

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 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 Protocol. 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 PEER requests, for MAP requests that included the PREFER_FAILURE Option (because otherwise a new external port could have been assigned), or MAP requests for the SCTP protocol. 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. 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 will need to properly handle such updated responses for any request it sends, most notably to support Mapping Update (Section 14.2). Also see 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 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].

To assist with detecting an on-path NAT, 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 15.4 for how this can be coded.

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 client begins the message exchange by transmitting a message to the server. The message exchange terminates when either the client successfully receives the appropriate response or responses from the server, or 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:

\[ RT = IRT + RAND \times IRT \]

RT for each subsequent message transmission is based on the previous value of RT:

\[ RT = 2 \times RT_{prev} + RAND \times RT_{prev} \]

MRT specifies an upper bound on the value of RT (disregarding the randomization added by the use of RAND). If MRT has a value of 0, there is no upper limit on the value of RT. Otherwise:

\[
\text{if } (RT > MRT) \\
RT = MRT + RAND \times MRT
\]

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.

Once a PCP client has successfully received a response from a PCP server on that interface, it resets RT to its initial value and sends subsequent PCP requests to that same server.

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 interfaces from properly authorized clients.

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 (including the ANNOUNCE Opcode), 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:
- Copying the entire request packet, or 1024 octets, whichever is less, and zero-padding the response to a multiple of 4 octets if necessary
- Setting the R bit
- Setting the Result Code
- Setting the Lifetime, Epoch Time and Reserved fields
- 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:
- Copying the first four octets of request packet header
- Setting the R bit
- Setting the Result Code to zero
- Setting the Lifetime, Epoch Time and Reserved fields
- Possibly setting opcode-specific response data if appropriate
- 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 1024 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 1024 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 1024 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) 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 11.5.

If the result code is 0 (SUCCESS), the request succeeded.
If the result code is not 0, the request failed. If the result code is NO_RESOURCES, the PCP client SHOULD NOT send further requests for new mappings to that PCP server for the value of the Lifetime (limited by the sanity checking detailed in Section 11.5). If a request for renewal or deletion of an existing mapping results in NO_RESOURCES, the PCP client SHOULD NOT send further requests of any kind to that PCP server for the (limited) value of the Lifetime. For other error result codes, the PCP client SHOULD NOT resend the same request for the (limited) value of the Lifetime.

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 which 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., WiFi 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 public 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:

1. 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 value \( (current\_server\_time) \) is less than the previously received PCP server Epoch time value \( (previous\_server\_time) \) then the client treats the Epoch time value as obviously invalid (time should not go backwards), else
     + The client computes the difference between its current local time value \( (current\_client\_time) \) and the time the previous PCP response was received from this PCP server \( (previous\_client\_time) \): \( client\_delta = current\_client\_time - previous\_client\_time; \)
     + The client computes the difference between the current PCP server Epoch time value \( (current\_server\_time) \) and the previously received Epoch time value \( (previous\_server\_time) \): \( server\_delta = current\_server\_time - previous\_server\_time; \)
     + 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

2. The client records the current time values for use in its next comparison:
   \[previous\_client\_time = current\_client\_time\]
   \[previous\_server\_time = current\_server\_time\]
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 15.3.1.

Notes:

- The client clock MUST never go backwards. If current_client_time is found to be less than previous_client_time then this is a client bug, and how the client deals with this client bug is implementation specific.

- The calculations above are constructed to allow client_delta and server_delta to be computed as unsigned integer values.

- 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).

- 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 1. Should later updates to this document specify different message formats with a version number greater than 1 it is expected that PCP servers will still support version 1 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 1, and backwards compatibility is desired, this first octet can be used for forward and backward compatibility.

If future PCP versions greater than 1 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 1. If using version 1 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 15.5.

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 an 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 its 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.

<table>
<thead>
<tr>
<th></th>
<th>Internal</th>
<th>External</th>
<th>Remote Peer</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPv4 firewall</td>
<td>IPv4</td>
<td>IPv4</td>
<td>IPv4</td>
</tr>
<tr>
<td>IPv6 firewall</td>
<td>IPv6</td>
<td>IPv6</td>
<td>IPv6</td>
</tr>
<tr>
<td>NAT44</td>
<td>IPv4</td>
<td>IPv4</td>
<td>IPv4</td>
</tr>
<tr>
<td>NAT64</td>
<td>IPv6</td>
<td>IPv4</td>
<td>IPv6</td>
</tr>
<tr>
<td>NPTv6</td>
<td>IPv6</td>
<td>IPv6</td>
<td>IPv6</td>
</tr>
</tbody>
</table>

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 and port to itself as described in Section 11 and (b) publish that public IP address and port via some sort of rendezvous server (e.g., DNS, a SIP message, a proprietary protocol). 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-zeros 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:

```c
/* 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 cases, apart from the
     * Suggested External Address and Port which may differ between
     * (1), (2), and (3).
     */
    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 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 15.1.
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
     * The PCP packet sent is identical in all cases, apart from the
     * Suggested External Address and Port which may differ between
     * (1), (2), and (3).
     */
    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();
}
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:

```c
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
     * The PCP packet sent is identical in all cases, apart from the
     * Suggested External Address and Port which may differ between
     * (1), (2), and (3).
     */
    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 source port, the NAT should translate that packet’s Internal Address, Protocol, and Port to the mapping’s External Address and Port, so that replies addressed to the External Address and Port are correctly translated.
to the mapping’s Internal Address and Port.

On Operating Systems that allow multiple listening servers to bind to the same Internal Port, servers MUST ensure that they have exclusive use of that Internal 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 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 Assigned External Port in the PCP response always match the Internal IP Address and Port in 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   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".

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 Protocol is 0 (meaning ‘all protocols’)

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 following diagram shows the format of Opcode-specific information in a response packet for the MAP Opcode:

```
|   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.

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 and Section 11.5 describe 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 and Port, and in that case it will assign a different External IP Address and Port.

If the Protocol does not use 16-bit port numbers (e.g., RSVP), the port number MUST be 0. 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).
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 11.5). 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 and Section 11.5 describe 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 and Internal Port 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, 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, it MUST generate an UNSUPP_PROTOCOL error. If the requested Lifetime is non-zero, the Internal Port is zero, and the Protocol is non-zero, it indicates a request to map all incoming traffic for that entire Protocol. If this request cannot be fulfilled in its entirety, the error UNSUPP_PROTOCOL MUST be returned. If the requested Lifetime is non-zero, the Internal Port is zero, and the Protocol is zero, it indicates a request to map all incoming traffic for all protocols. If this request cannot be fulfilled in its entirety, the error UNSUPP_PROTOCOL MUST be returned. If the Protocol is zero but the Internal Port is non-zero, the error MALFORMED_REQUEST MUST be returned.
If the requested lifetime is zero, it indicates a request to delete an existing mapping or set of mappings. Processing of the lifetime is described in Section 11.5.

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).
The PCP server MUST NOT create mappings for the PCP ports themselves (5350 and 5351), and SHOULD have a policy control to deny mappings for other ports. In these cases, the error NOT_AUTHORIZED SHOULD be returned.

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). If the PCP server cannot assign the Suggested External Address and Port but can assign some other External Address and Port (and the request did not contain the PREFER_FAILURE Option) the PCP server MUST do so and return the newly assigned External Address and Port in the response. Cases where a PCP server or PCP-controlled device cannot assign the Suggested External Address and Port include:

- The Suggested External Address and Port is already assigned to another existing explicit, implicit, or static mapping (i.e., is already forwarding traffic to some other internal address, protocol, and port).
- The Suggested External Address and Port is already used by the NAT gateway for one of its own services (e.g., port 80 for the NAT gateway’s own configuration pages).
- The Suggested External Address and Port is otherwise prohibited by the PCP server’s policy.
- The Suggested External Address or port is invalid (e.g., 127.0.0.1, ::1, multicast address, or the port is not valid for the indicated protocol).
- The Suggested External Address does not belong to the NAT gateway.
- The Suggested External Address is not configured to be used as an external address of the firewall or NAT gateway.
- The PREFER_FAILURE option is included in the request and the Suggested External Address and Port are not assignable to the PCP client, which returns the CANNOT_PROVIDE_EXTERNAL error.

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 and the Internal Port. Other fields are not compared, because the PCP server sets those fields. Note that if the PCP server supports Mapping Update (Section 14.2) the PCP server will send additional MAP responses if the mapping changes (e.g., due to IP renumbering).

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 15.3.1.

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. 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...
server SHOULD be configurable for permitted minimum and maximum lifetime, and the RECOMMENDED values are 120 seconds for the minimum value and 24 hours for the maximum. It is RECOMMENDED that the server be configurable to restrict lifetimes to less than 24 hours, because mappings will consume 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 MAP or PEER 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.

If the requested lifetime is zero then:

- If both the internal port and protocol are non-zero, it indicates a request to delete the indicated mapping immediately.

- If both the internal port and protocol are zero, it indicates a request to delete all mappings for this Internal Address for all transport protocols. This is useful when a host reboots or joins a new network, to clear out prior stale state from the NAT gateway before beginning to install new mappings.

- If the internal port is zero and the protocol is non-zero, it indicates a request to delete a previous 'wildcard' (all-ports) mapping for that protocol.

- If the internal port is non-zero and the protocol is 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 PEER requests cannot delete or shorten lifetimes of any mappings. If a PEER request matches an existing mapping, and the requested lifetime is less than that of the existing mapping, the PCP server returns SUCCESS with the lifetime of the existing mapping.

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 PCP MAP request was for port=0 (indicating 'all ports'), the PCP server deletes all of the explicit dynamic mappings it can (but not any implicit or static mappings), and returns a SUCCESS response. If the deletion request was properly formatted and successfully processed, a SUCCESS response is generated with lifetime of 0 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).

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. PCP provides some protections against such port consumption: When a PCP client first acquires a new IP address (e.g., reboots or joins a new network), it SHOULD remove mappings that may already be instantiated for that new Internal Address. To do this, the PCP client sends a MAP request with protocol, internal port, and lifetime set to 0. Some port mapping APIs (e.g., the "DNSServiceNATPortMappingCreate" API provided by Apple’s Bonjour on Mac OS X, iOS, Windows, Linux [Bonjour]) automatically monitor for process exit (including application crashes) and automatically send port mapping deletion requests if the process that requested them goes away without explicitly relinquishing them.
To reduce unwanted traffic and data corruption, External UDP and TCP ports SHOULD NOT be re-used for an interval (TIME_WAIT interval [RFC0793]). However, the PCP server SHOULD allow the previous user of an External Port to re-acquire the same port during that interval.

11.6. 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 both implicit dynamic mappings and explicit dynamic mappings. However, 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 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 public 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 WiFi 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.
11.7. 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. In the case of PCP, this operation no longer makes sense. PCP supports Large Scale 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 Opcode 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

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:

![PEER Opcode Request Diagram](image)

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.

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.

Reserved: 24 reserved bits, MUST be set to 0 on transmission and MUST be ignored on reception.
Internal Port: Internal port for the mapping.

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.

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 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 or Port, then the PCP server MUST return a CANNOT_PROVIDEEXTERNAL 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 MALFORMEDREQUEST 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
|   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.

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; IPv4 or IPv6 address is indicated by the Opcode. 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 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, and Remote Peer Port.

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 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. If the requested mapping does not yet exist, and the Suggested External Address and Port can be honored, the 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 15.3.1). 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 the requested mapping does not yet exist, and Suggested External Address and Port cannot be honored, the error CANNOT_PROVIDE_EXTERNAL is returned. If the requested mapping already exists, it is a request to modify the lifetime of that existing mapping.

The PEER Opcode can extend the lifetime of an existing dynamic outbound mapping. The PCP server may grant the client’s requested lifetime, or may grant a value higher or lower, depending on local policy. The PEER Opcode MUST NOT ever reduce the lifetime of an existing outbound mapping. If the Requested Lifetime is less than the lifetime of the existing mapping, it is treated as a request for the lifetime of the mapping (this 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 and the Remote Peer Port. Other fields are not compared, because the PCP server sets those fields to provide information about the mapping created by the Opcode. Note that if the PCP server supports Mapping Update (Section 14.2) the PCP server will send additional PEER responses if the mapping changes (e.g., due to IP renumbering).

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 issue a new PCP request prior to the expiration, or it MAY rely on continued inside-to-outside traffic to ensure the mapping will continue to exist. See Section 11.2.1 for recommended renewal timing.

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.

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 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. 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. This is because many PCP servers may not implement the THIRD_PARTY Option at all, and a client using the THIRD_PARTY Option to specify the same address as the source address of the packet will cause mapping requests to fail where they would otherwise have succeeded.

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.

Determining which PCP clients are authorized to use the THIRD_PARTY Option for which other hosts is deployment-dependent. For example, an ISP using Dual-Stack Lite could choose to allow a client connecting over a given IPv6 tunnel to manage mappings for any other host connecting over the same IPv6 tunnel, or the ISP could choose to allow only the DS-Lite B4 element to manage mappings for other hosts connecting over the same IPv6 tunnel. A cryptographic authentication and authorization model is outside the scope of this specification. 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 to be a single host from the point of view of the ISP.

Where possible, it may beneficial if a client using the THIRD_PARTY Option to create and maintain mappings on behalf of some other device can take steps to verify that the other device is still present and active on the network. Otherwise the client using the THIRD_PARTY Option to maintain mappings on behalf of some other device 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.

A PCP client can delete all PCP-created explicit dynamic inbound mappings (i.e., those created by PCP MAP requests) that it is authorized to delete by sending a PCP MAP request with zero in the Internal Port field, zero in the Protocol field, and a THIRD_PARTY Option with the all-zeros address in the Internal IP Address field. Upon receipt of such a request from an authorized PCP client, the PCP server MUST delete all described mappings the PCP client is authorized to delete, and return SUCCESS. SUCCESS is returned if zero or more mappings were deleted.
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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| 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 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, rounded up to the next larger integer number of seconds.

This Option exists for interworking with non-PCP mapping protocols that have different semantics than PCP (e.g., UPnP IGDv1 interworking [I-D.bpw-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.

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). Because of this, the PCP client MUST validate that the External IP Address and Port 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. If the PCP server has no use for this new mapping, it SHOULD delete the mapping.

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 MAP Opcode, and the Remote Peer Port and Remote Peer IP Address of the FILTER Option indicate the permitted remote peer’s source IP address and 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 FILTER Option is formatted as follows:

```
  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 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.

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 a unwanted traffic.

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.

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 of PCP client(s), sends unicast ANNOUNCE responses to those address(es).

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 responses have no Opcode-specific payload (that is, the length of the Opcode-specific data is zero), so a packet layout is not shown. 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 (i.e., an error is generated if the request packet is malformed, such as invalid length). 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: A separate port is used when the server sends an unsolicited ANNOUNCE Opcode (5350) than normal client-initiated requests (5351) because a single device may have multiple roles. For example, a multi-function home gateway may also provide printer sharing, or a home computer may also provide "Internet Sharing" (NAT) functionality. 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

Generating a PCP restart announcement, described in this section, is an optional feature of PCP. This message is most typically multicast, but can also be unicast. Unicast requires the PCP server send the message to the PCP client’s IP address(es), which it may not know after it loses state. When sending unsolicited unicast ANNOUNCE responses, they are sent to port 5351. When sending unsolicited multicast ANNOUNCE responses, they are sent to port 5350. When sending unsolicited responses, the ANNOUNCE Opcode MUST have Result Code equal to zero (SUCCESS).

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). If sending a multicast ANNOUNCE message, 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 unicast ANNOUNCE messages, it sends ANNOUNCE response message to the IP address(es) of the PCP clients with destination port 5350. 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 listen to receive those on their unicast address. 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 unicast 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.

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 public IP address of a PCP-controlled NAT).

Implementation of the feature described in this section is optional for PCP servers. 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 MAY chose to remedy this situation by automatically repairing its mappings and notifying its clients by following the procedure described below.

For MAP-created and PEER-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. 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 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 MAY transmit such packets 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.

Upon receipt of such 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 indicating the external port or address changed, this would typically mean re-establishing 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. Implementation Considerations

15.1. Implementing MAP with EDM port-mapping NAT

This section provides non-normative guidance that may be useful to implementers.

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 such 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 public 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.

15.2. Lifetime of Explicit and Implicit Dynamic Mappings

This section provides non-normative guidance that may be useful to implementers.

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.

15.3. PCP Failure Scenarios

This section provides non-normative guidance that may be useful to implementers.

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].

15.3.1. Recreating Mappings

This section provides non-normative guidance that may be useful to implementers.
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.

15.3.2. Maintaining Mappings

This section provides non-normative guidance that may be useful to implementers.

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 public IP address and/or public port, or the PCP server itself, has changed (due to a new route to a different PCP server). Such events are 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, the PCP client may find it beneficial to use shorter lifetimes (so that it communicates with the PCP server more often).

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 to the previous values (see Section 10.1 and
Section 11.6).

15.3.3.  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 rewrite SCTP port numbers, a PCP MAP or PEER request for an SCTP mapping SHOULD provide the same Internal Port and Suggested External Port. If the PCP server supports SCTP, and the suggested external port cannot be provided in an explicit dynamic SCTP mapping, then the error CANNOT_PROVIDE_EXTERNAL is returned. This places an extra burden on the SCTP client because it then has to tear down its listening socket and try again with a different Internal Port, repeatedly until it is successful in finding a 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).

15.4.  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 to include 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.

15.5. State Diagram

Each mapping entry of the PCP-controlled device would go through the state machine shown below. This state diagram is non-normative.

![PCP State Diagram](image)

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.

16. Deployment Considerations

16.1. Ingress Filtering

As with implicit dynamic mappings created by outgoing TCP 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
16.2. 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.

17. 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.

17.1. 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.

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 Opcode can be secure under this threat model if 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), explicit mappings are created...
with the same lifetime as implicit mappings, the PCP server does not support deleting or reducing the lifetime of existing mappings, and the PCP server does not support the third party option. 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 17.2 MUST enforce the constraints described in the paragraph above.

17.1.1. 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.
17.1.2. 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.

17.1.2.1. Residential Gateway Deployment

Parity with many currently-deployed residential gateways can be achieved using a PCP Server that is constrained as described in Section 17.1.1 above.

17.1.2.2. DS-Lite Deployment

A DS-Lite deployment could be secure under the Simple Threat Model, even if the B4 device makes PCP mapping requests on behalf of internal clients using the THIRD_PARTY option. In this case the DS-Lite PCP server MUST be configured to only allow the B4 device to make THIRD_PARTY requests, and only on behalf of other Internal Hosts sharing the same DS-Lite IPv6 tunnel. The B4 device MUST guard against spoofed packets being injected into the IPv6 tunnel using the B4 device’s IPv4 source address, so the DS-Lite PCP Server can trust that packets received over the DS-Lite IPv6 tunnel with the B4 device’s source IPv4 address do in fact originate from the B4 device. The B4 device is in a position to enforce this requirement, because it is the DS-Lite IPv6 tunnel endpoint.

Allowing the B4 device to use the THIRD_PARTY Option to create mappings for hosts reached via the IPv6 tunnel terminated by the B4 device is acceptable, because the B4 device is capable of creating these mappings implicitly and can prevent others from spoofing these mappings.

17.2. Advanced Threat Model

In the Advanced Threat Model the PCP protocol must be ensure 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:

- Security infrastructure equipment, such as corporate firewalls, that does not create implicit mappings.

- 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.

- Any implementation that wants to be more permissive in authorizing explicit mappings than it is in authorizing implicit mappings.

- Implementations that support the THIRD_PARTY Option (unless they can meet the constraints outlined in Section 17.1.2.2).

- Implementations that wish to support any deployment scenario that does not meet the constraints described in Section 17.1.

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 17.1 above, in their default configuration, when processing unauthenticated requests.

17.3. Residual Threats

This section describes some threats that are not addressed in either of the above threat models, and recommends appropriate mitigation strategies.

17.3.1. 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 16.2 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.
17.3.2.  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.

17.3.3.  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 and port cannot be used by another host (e.g., by using a different IP address pool).

17.3.4.  Attacks Against Server Discovery

This document does not specify server discovery, beyond contacting the default gateway.

18.  IANA Considerations

IANA is requested to perform the following actions:

18.1.  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.]
18.2. Opcodes

IANA shall create a new protocol registry for PCP Opcodes, numbered 0-127, initially populated with the values:

<table>
<thead>
<tr>
<th>value</th>
<th>Opcode</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>ANNOUNCE</td>
</tr>
<tr>
<td>1</td>
<td>MAP</td>
</tr>
<tr>
<td>2</td>
<td>PEER</td>
</tr>
<tr>
<td>3-31</td>
<td>Standards Action [RFC5226]</td>
</tr>
<tr>
<td>32-63</td>
<td>Specification Required [RFC5226]</td>
</tr>
<tr>
<td>96-126</td>
<td>Private Use [RFC5226]</td>
</tr>
<tr>
<td>127</td>
<td>Reserved, Standards Action [RFC5226]</td>
</tr>
</tbody>
</table>

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].

18.3. 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].

18.4. Options

IANA shall create a new registry for PCP Options, numbered 0-255 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>

19. 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 Farrel, Ralph Droms, 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 Margaret Wasserman for writing the Security Considerations section.

Thanks to authors of DHCPv6 for retransmission text.

20. References

20.1. Normative References


20.2. Informative References

[Bonjour] "Bonjour",
<http://en.wikipedia.org/wiki/Bonjour_(software)>

[I-D.arkko-dual-stack-extra-lite]
Arkko, J., Eggert, L., and M. Townsley, "Scalable Operation of Address Translators with Per-Interface Bindings",
draft-arkko-dual-stack-extra-lite-05 (work in progress), February 2011.

[I-D.boucadair-pcp-failure]
draft-boucadair-pcp-failure-02 (work in progress), September 2011.

[I-D.bpw-pcp-upnp-igd-interworking]
Boucadair, M., Penno, R., Wing, D., and F. Dupont, "Universal Plug and Play (UPnP) Internet Gateway Device (IGD)-Port Control Protocol (PCP) Interworking Function",
draft-bpw-pcp-upnp-igd-interworking-02 (work in progress), February 2011.

[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., "NAT Port Mapping Protocol (NAT-PMP)",
draft-cheshire-nat-pmp-03 (work in progress), April 2008.

[I-D.ietf-behave-lsn-requirements]
Perreault, S., Yamagata, I., Miyakawa, S., Nakagawa, A.,
and H. Ashida, "Common requirements for Carrier Grade NATs
[I-D.ietf-behave-sctpnat]  

[I-D.miles-behave-l2nat]  

[IGDv1]  

[RFC0793]  

[RFC1918]  

[RFC2136]  

[RFC3007]  

[RFC3022]  

[RFC3581]  

[RFC3587]  

[RFC4291]  

[RFC4303]  
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-23 to -24

- Explained common questions regarding PCP’s design, such as lack of transaction identifiers and its request/response semantics and operation (Protocol Design Note (Section 6)).
- added MUST for all-zeros IPv6 and IPv4 address formats.
- included field definitions for Opcode-specific information and PCP options under both Figure 2 and Figure 3.
- adopted retransmission mechanism from DHCPv6.
- 1024 message size limit described in PCP message restriction.
- 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).
- 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).

- eliminated overuse of 'packet format' when it was 'opcode format'.

- for IANA registries, added code points assignable via Standards Action (previously was just Specification Required).

- Version negotiation, added explanation that retrying after 30 minutes makes the protocol self-healing if the PCP server is upgraded.

- Version negotiation now accommodates non-contiguous version numbers.

- Tweaked definition of VERSION field (that "1" is for this version, but other values could of course appear in the future).

- when receiving unsolicited ANNOUNCE, PCP client now waits random 0-5 seconds.

- Removed 'interworking function' from list of terminology because we no longer use the term in this document.

- tightened definitions of 'PCP client' and 'PCP server'.

- For 'Requested Lifetime' definitions, removed text requiring its value be 0 for not-yet-defined opcodes.

- Removed some unnecessary text suggesting logging (is an implementation detail).

- Added active-mode FTP as example protocol that can break with mappings to different IP addresses.

- 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.

- Changed "internal address and port" to "internal address, protocol, and port" in several places.

- Clarified which 96 bits are copied into error response. Clarified that only error responses are copied verbatim from request.

- a single PCP server can control multiple NATs or multiple firewalls (Section 4).
Clarified that sending unsolicited multicast ANNOUNCE is not always available on all networks.

Clarified option length error example is when option length exceeds UDP length.

Explained that an on-path attacker that can spoof packets can redirect traffic to a host of their choosing.

Instead of saying IPv4-mapped addresses won’t appear on the wire, say they aren’t used for mappings.

THIRD_PARTY is useful for management device (e.g., in a network operations center).

Clarified PCP responses have fields updated as indicated with 'set by the server' from field definitions.

Disallow using MAP to the PCP ports themselves and encourage implementations have policy control for other ports.

Instead of 'idempotent', now says 'identical requests generate identical response'.

Described which Options are included when sending Mapping Update (unsolicited responses), Section 14.2.

Dropped [RFC2136] and [RFC3007] to informative references.

Updated from 'should' to 'SHOULD' in Section 16.1.

Described 'hairpin' in terminology section.

B.2. Changes from draft-ietf-pcp-base-22 to -23

Instead of returning error NO_RESOURCES when requesting a MAP for all protocols or for all ports, return UNSUPP_PROTOCOL.

Clarify that PEER-created mappings are treated as if it was implicit dynamic outbound mapping (Section 12.3).

Point out that PEER-created mappings may be very short until bi-directional traffic is seen by the PCP-managed device.

Clarification that an existing implicit mapping (created e.g., by TCP SYN) can become managed by a MAP request (Section 11.3.)
- Clarified the ANNOUNCE Opcode is being defined in Section 14.1, and that the length of requests (as well as responses) is zero.
- Clarify that ANNOUNCE has Lifetime=0 for requests and responses.
- Clarify ANNOUNCE can be sent unicast by the client (to solicit a response), or can be multicasted (unsolicited) by the server.
- Allow ANNOUNCE to be sent unicast by the server, to accommodate case where PCP server fails but knows the IP address of a PCP client (e.g., web portal).
- Clarified ports used for unicast and multicast unsolicited ANNOUNCE.
- Tweaked NO_RESOURCES handling, to just disallow *new* mappings.
- 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).
- In section Section 11.5, clarified that PEER cannot delete or shorten any lifetime, and that MAP can only shorten or delete lifetimes of MAP-created mappings.
- Clarified handling of MAP when mapping already exists (4 steps).
- 2^32-1
- Randomize retry interval (1.5-2.5), and maximum retry interval is now 1024 seconds (was 15 minutes).
- Remove MUST be 0 for Reserved field when sending error responses for un-parseable message.
- 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.
- 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.
- 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.3. Changes from draft-ietf-pcp-base-21 to -22

- Removed paragraph discussing multiple addresses on the same (physical) interface; those will work with PCP.
- 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).
- Point out NO_RESOURCES attack vector in security considerations.
- 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.4. Changes from draft-ietf-pcp-base-20 to -21

- To delete all mappings using THIRD_PARTY, use the all-zeros IP address (rather than previous text which used length=0).
- added normative text for what PCP server does when it receives all-zeros IP address in THIRD_PARTY option.
- PREFER_FAILURE allowed for use by web portal.
- clarifications to mandatory option processing.
- cleanup and wordsmithing of the THIRD_PARTY text.

B.5. Changes from draft-ietf-pcp-base-19 to -20

- clarify if Options are included in responses.
- clarify when External Address can be ignored by the PCP server / PCP-controlled device
- added ‘Transition from state M to state I is implementation dependent’ to state diagram

B.6. Changes from draft-ietf-pcp-base-18 to -19

- Described race condition with MAP containing PREFER_FAILURE and Mapping Update.
- Added state machine (Section 15.5).
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.7. 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.8. 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 11.7).

o Added Restart Announcement.

o Tweaked terminology.

o Detailed how error responses are generated.

B.9. 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.10. 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.11. 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 "0" 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.
Added table to show which fields of MAP or PEER need IPv6/IPv4 addresses for IPv4 firewall, DS-Lite, NAT64, NAT44, etc.

Accommodate the server’s Epoch going up or down, to better detect switching to a different PCP server.

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.12. Changes from draft-ietf-pcp-base-12 to -13

All addresses are 128 bits. IPv4 addresses are represented by IPv4-mapped IPv6 addresses (::FFFF/96).

PCP request header now includes PCP client’s port (in addition to the client’s IP address, which was in -12).

new ADDRESS_MISMATCH error.

removed PROCESSING_ERROR error, which was too similar to MALFORMED_REQUEST.

Tweaked text describing how PCP client deals with multiple PCP server addresses (Section 8.1).

clarified that when overloaded, the server can send SERVER_OVERLOADED (and drop requests) or simply drop requests.

Clarified how PCP client chooses MAP4 or MAP6, depending on the presence of its own IPv6 or IPv4 interfaces (Section 10).

compliant PCP server MUST support MAPx and PEERx, SHOULD support ability to disable support.

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.

Integrated WGLC feedback (small changes to abstract, definitions, and small edits throughout the document)

allow new Options to be defined with a specification (rather than standards action)
B.13. Changes from draft-ietf-pcp-base-11 to -12

- added implementation note that MAP and implicit dynamic mappings have independent mapping lifetimes.

B.14. Changes from draft-ietf-pcp-base-10 to -11

- clarified what can cause CANNOT_PROVIDE_EXTERNAL error to be generated.

B.15. Changes from draft-ietf-pcp-base-09 to -10

- 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.16. Changes from draft-ietf-pcp-base-08 to -09

- Clarified in PEER Opcode introduction (Section 12) that they can also create mappings.
- 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).
- 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.
- Removed text encouraging PCP server to statefully remember its mappings from Section 15.3.1, as it didn’t belong there. Text in Security Considerations already encourages persistent storage.
- 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).
- 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.
- Mapping theft can also be mitigated by ensuring hosts can’t re-use same IP address or port after state loss.
- 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).
created new Implementation Considerations section (Section 15) which discusses non-normative things that might be useful to implementers. Some new text is in here, and the Failure Scenarios text (Section 15.3) has been moved to here.

Tweaked wording of EDM NATs in Section 15.1 to clarify the problem occurs both inside->outside and outside->inside.

removed "Interference by Other Applications on Same Host" section from security considerations.

fixed zero/non-zero text in Section 11.5.

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.

Renamed error from UNAUTH_TARGET_ADDRESS to UNAUTH_THIRD_PARTY_INTERNAL_ADDRESS.

for FILTER option, removed unnecessary detail on how FILTER would be bad for PEER, as it is only allowed for MAP anyway.

In Security Considerations, explain that PEER can create a mapping which makes its security considerations the same as MAP.

B.17. Changes from draft-ietf-pcp-base-07 to -08

moved all MAP4-, MAP6-, and PEER-specific options into a single section.

discussed NAT port-overloading and its impact on MAP (new section Section 15.1), which allowed removing the IMPLICIT_MAPPING_EXISTS error.

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.

Removed Deployment Scenarios section (which detailed NAT64, NAT44, Dual-Stack Lite, etc.).

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.18. 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
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.

- Added NAT-PMP transition section
- Separated Internal Address, External Address, Remote Peer Address definition
- Unified Mapping, Port Mapping, Port Forwarding definition
- Adjusted so DHCP configuration is non-normative.
- Mentioned PCP refreshes need to be sent over the same interface.
- Renamed the REMOTE_PEER_FILTER option to FILTER.
- Clarified FILTER option to allow sending an ICMP error if policy allows.
- 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.
- Clarified that PEER requests have to be sent from same interface as the connection itself.
- For MAP opcode, text now requires mapping be deleted when lifetime expires (per consensus on 8-Mar interim meeting)
- 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’.
- 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).
- Clarified that PCP server only listens on its internal interface.
- Abandoned ‘target’ term and reverted to simpler ‘internal’ term.

B.19. Changes from draft-ietf-pcp-base-05 to -06

- Dual-Stack Lite: consensus was encapsulation mode. Included a suggestion that the B4 will need to proxy PCP-to-PCP and UPnP-to-PCP.
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.

- 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.

- 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.

- Swapped Option values for the mandatory- and optional-to-process Options, so we can have a simple lowest..highest ordering.

- added MISORDERED_OPTIONS error.

- 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.

- clarified that, after successfully using a PCP server, that PCP server is declared to be non-responsive after 5 failed retransmissions.

- 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.)

- Added better discussion of multiple interfaces, including highlighting WiFi+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.

- added some text about why server might send SERVER_OVERLOADED, or might simply discard packets.

- Dis-allow internal-port=0, which means we dis-allow using PCP as a DMZ-like function. Instead, ports have to be mapped individually.

- Text describing server’s processing of PEER is tightened up.

- 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.20. Changes from draft-ietf-pcp-base-04 to -05

- Tweaked PCP common header packet layout.
- Re-added port=0 (all ports).
- Minimum size is 12 octets (missed that change in -04).
- Removed Lifetime from PCP common header.
- For MAP error responses, the lifetime indicates how long the server wants the client to avoid retrying the request.
- More clearly indicated which fields are filled by the server on success responses and error responses.
- Removed UPnP interworking section from this document. It will appear in [I-D.bpw-pcp-upnp-igd-interworking].

B.21. Changes from draft-ietf-pcp-base-03 to -04

- "Pinhole" and "PIN" changed to "mapping" and "MAP".
- Reduced from four MAP Opcodes to two. This was done by implicitly using the address family of the PCP message itself.
- New option THIRD_PARTY, to more carefully split out the case where a mapping is created to a different host within the home.
- Integrated a lot of editorial changes from Stuart and Francis.
- Removed nested NAT text into another document, including the IANA-registered IP addresses for the PCP server.
- Removed suggestion (MAY) that PCP server reserve UDP when it maps TCP. Nobody seems to need that.
- Clearly added NAT and NAPT, such as in residential NATs, as within scope for PCP.
- HONOR_EXTERNAL_PORT renamed to PREFER_FAILURE
- Added 'Lifetime' field to the common PCP header, which replaces the functions of the 'temporary' and 'permanent' error types of the previous version.
Allow arbitrary Options to be included in PCP response, so that PCP server can indicate un-supported PCP Options. Satisfies PCP Issue #19.

- Reduced scope to only deal with mapping protocols that have port numbers.
- Reduced scope to not support DMZ-style forwarding.
- Clarified version negotiation.

B.22. Changes from draft-ietf-pcp-base-02 to -03

- Adjusted abstract and introduction to make it clear PCP is intended to forward ports and intended to reduce application keepalives.
- 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.
- Moved subscriber identity from common PCP section to MAP* section.
- Made clearer that PCP client can reduce mapping lifetime if it wishes.
- Added discussion of host running a server, client, or symmetric client+server.
- Introduced PEER4 and PEER6 Opcodes.
- Removed REMOTE_PEER Option, as its function has been replaced by the new PEER Opcodes.
- IANA assigned port 44323 to PCP.
- Removed AMBIGUOUS error code, which is no longer needed.

B.23. Changes from draft-ietf-pcp-base-01 to -02

- More error codes
- PCP client source port number should be random
- PCP message minimum 8 octets, maximum 1024 octets.
- 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.24. 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
Abstract

This document specifies DHCP (IPv4 and IPv6) options to configure hosts with Port Control Protocol (PCP) Server addresses. The use of DHCPv4 or DHCPv6 depends on the PCP deployment scenario.

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 July 16, 2012.

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
1. Introduction

This document defines DHCPv4 [RFC2131] and DHCPv6 [RFC3315] options which can be used to provision PCP Server [I-D.ietf-pcp-base] reachability information; more precisely it defines DHCP options to convey a name (as per Section 3.1 of [RFC1035]) of PCP Server(s).

In order to make use of these options, this document assumes appropriate name resolution means (e.g., Section 6.1.1 of [RFC1123]) are available on the host client.

The use of DHCPv4 or DHCPv6 depends on the PCP deployment scenarios.

2. Terminology

This document makes use of the following terms:

- 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].
- PCP Client denotes a PCP software instance responsible for issuing PCP requests to a PCP Server. Refer to [I-D.ietf-pcp-base].
- DHCP refers to both DHCPv4 [RFC2131] and DHCPv6 [RFC3315].
- DHCP client (or client) denotes a node that initiates requests to obtain configuration parameters from one or more DHCP servers [RFC3315].
- DHCP server (or server) refers to a node that responds to requests from DHCP clients [RFC3315].
- Name is a domain name (as per Section 3.1 of [RFC1035]) 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. Rationale

Both IP Address and Name DHCP options have been considered in early stages of this specification. This flexibility aims to let service providers to make their own engineering choices and use the convenient option according to their deployment context. Nevertheless, DHC WG’s position is this flexibility have some drawbacks such as inducing errors. Therefore, only the Name option is maintained within this document.
This document defines an option to carry a name rather than an IP address. This choice is motivated by operational considerations: In particular, some Service Providers are considering two levels of redirection:

1. The first level is national-wise is undertaken by DHCP: a regional-specific FQDN will be returned;
2. The second level is done during the resolution of the regional-specific FQDN to redirect the customer to a regional PCP server among a pool deployed regionally.

Distinct operational teams are responsible for each of the above mentioned levels. A clear separation between the functional perimeter of each team is a sensitive task for the maintenance of the offered services. Regional teams will require to introduce new resources (e.g., new PCP-controlled devices such as Carrier Grade NATs (CGNs, [I-D.ietf-behave-lsn-requirements])) to meet an increase of customer base. Operations related to the introduction of these new devices (e.g., addressing, redirection, etc.) are implemented locally. Having this regional separation provides flexibility to manage portions of network operated by dedicated teams. This two-level redirection can not be met by the IP Address option.

In addition to the operational considerations:
- The use of the Name for NAT64 [RFC6146] might be suitable for load-balancing purposes;
- For the DS-Lite case [RFC6333], if the encapsulation mode is used to send PCP messages, an IP address may be used since the AFTR selection is already done via the AFTR_NAME DHCPv6 option [RFC6334]. Of course, this assumes that the PCP Server is co-located with the AFTR function. If these functions are not co-located, conveying the Name would be more convenient.

4. Consistent NAT and PCP Configuration

The PCP Server discovered through DHCP must be able to install mappings on the appropriate upstream PCP-controlled device that will be crossed by packets transmitted by the host or any terminal belonging to the same realm (e.g., DHCP client is embedded in a CP router). In case this prerequisite is not met, customers would experience service troubles and their service(s) won’t be delivered appropriately.

Note that this constraint is implicitly met in scenarios where only one single PCP-controlled device is deployed in the network.
5. IP Address Selection

Resolving the Name conveyed in DHCP PCP Name options may return a list of IP addresses. This section specifies the behavior to be followed by the PCP Client to contact its PCP Server.

1. If only one PCP Name option is returned in DHCP: the PCP Client follows the procedure specified in Section 5.1 if a list of IP addresses are returned as a result of resolving the name conveyed in the PCP Name DHCP option.

2. If several PCP Name options are returned in DHCP: the PCP Client contacts in parallel all PCP Servers as defined in Section 5.2. For each PCP Name option occurrence, the PCP Client resolves the conveyed name; if more than one IP address are returned, the PCP Client follows the procedure specified in Section 5.1.

5.1. 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 Servers. 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 on that interface, it sends subsequent PCP requests to that same server 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 on its list.

5.2. Parallel Queries

The PCP Client contacts in parallel all the PCP Servers in the IP addresses list. For each IP address in the list, the PCP Client follows the procedure specified in Section 7.1 of [I-D.ietf-pcp-base].

6. DHCPv6 PCP Server Option

This DHCPv6 option conveys a domain name to be used to retrieve the IP addresses of PCP Server(s). Appropriate name resolution queries
should be issued to resolve the conveyed name. For instance, in the context of a DS-Lite architecture [RFC6333], the retrieved address may be an IPv4 address or an IPv4-mapped IPv6 address [RFC4291], and in the case of NAT64 [RFC6146] an IPv6 address can be retrieved.

6.1. Format

The format of the DHCPv6 PCP Server option is shown in Figure 1.

```
+---------------------------------+--------------------------------------------------+
| Option-code: OPTION_PCP_SERVER  | Option-length                                    |
| +---------------------------------+--------------------------------------------------+
| : PCP Server Domain Name         |                                                 |
+---------------------------------+--------------------------------------------------+
```

Figure 1: PCP Server Name DHCPv6 Option

The fields of the option shown in Figure 1 are as follows:

- Option-code: OPTION_PCP_SERVER (TBA, see Section 10.1)
- Option-length: Length of the ‘PCP Server Domain Name’ field in octets.
- PCP Server Domain Name: The domain name of the PCP Server to be used by the PCP Client. The domain name is encoded as specified in Section 8 of [RFC3315].

6.2. Client Behaviour

To discover a PCP Server [I-D.ietf-pcp-base], the DHCPv6 client MUST include an Option Request Option (ORO) requesting the DHCPv6 PCP Server Name option as described in Section 22.7 of [RFC3315] (i.e., include OPTION_PCP_SERVER on its OPTION_ORO). A client MAY also include the OPTION_DNS_SERVERS option on its OPTION_ORO to retrieve a DNS servers list.

If the DHCPv6 client receives more than one OPTION_PCP_SERVER option from the DHCPv6 server, it extracts the Name conveyed in each OPTION_PCP_SERVER option and proceeds to validating it. If more than one Name is included in a OPTION_PCP_SERVER option occurrence, only the first instance MUST be used. Then, the DHCPv6 client MUST verify that the option length does not exceed 255 octets (RFC1035). The DHCPv6 client MUST verify the name is properly encoded as detailed in Section 8 of [RFC3315].

Once the name conveyed in each OPTION_PCP_SERVER option is validated,
the included 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).

The PCP Client MUST follow the procedure specified in Section 5 to contact its PCP Server(s).

It is RECOMMENDED to associate a TTL with any address resulting from resolving the Name conveyed in a OPTION_PCP_SERVER DHCPv6 option when stored in a local cache. Considerations on how to flush out a local cache are out of the scope of this document.

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. When an application issues a PCP request to a PCP Server, the source address of the request MUST be among those assigned on the interface to which the destination PCP Server is bound.

7. DHCPv4 PCP Option

7.1. Format

The PCP Server Name DHCPv4 option can be used to configure a name to be used by the PCP Client to contact a PCP Server. The format of this option is illustrated in Figure 2.

```
| TBA | n  | s1 | s2 | s3 | s4 | s5 | ... |
```

The values s1, s2, s3, etc. represent the domain name labels in the domain name encoding.

Figure 2: PCP Server Name DHCPv4 Option

The description of the fields is as follows:
- Code: OPTION_PCP_SERVER (TBA, see Section 10.2);
- Length: Includes the length of the "PCP Server Domain Name" field in octets; The maximum length is 255 octets.
- PCP Server Domain Name: The domain name of the PCP Server to be used by the PCP Client when issuing PCP messages. The encoding of the domain name is described in Section 3.1 of [RFC1035].
7.2. Client Behaviour

DHCPv4 client expresses the intent to get OPTION_PCP_SERVER by specifying it in Parameter Request List Option [RFC2132].

If the DHCPv4 client receives more than one OPTION_PCP_SERVER option from the DHCPv4 server, it extracts the Name conveyed in each OPTION_PCP_SERVER option and proceeds to validating it. If more than one Name is included in a OPTION_PCP_SERVER option occurrence, only the first instance MUST be used. Then, the DHCPv4 client MUST verify that the option length does not exceed 255 octets [RFC1035]).

Once the name conveyed in each OPTION_PCP_SERVER option is validated, the included Name is passed to the name resolution library (e.g., Section 6.1.1 of [RFC1123] or [RFC6055]) to retrieve the corresponding IPv4 address(es).

The PCP Client MUST follow the procedure specified in Section 5 to contact its PCP Server(s).

It is RECOMMENDED to associate a TTL with any address resulting from resolving the Name conveyed in a OPTION_PCP_SERVER DHCPv4 option when stored in a local cache. Considerations on how to flush out a local cache are out of the scope of this document.

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. When an application issues a PCP request to a PCP Server, the source address of the request MUST be among those assigned on the interface to which the destination PCP Server is bound.

8. Dual-Stack Hosts

A PCP Server configured using OPTION_PCP_SERVER over DHCPv4 is likely to be resolved to IPv4 address(es).

A PCP Server configured using OPTION_PCP_SERVER over DHCPv6 may be resolved to IPv4 address(es) (e.g., DS-Lite [RFC6333]) or IPv6 address(es) (e.g., NAT64 [RFC6146], IPv6 firewall [RFC6092], NPTv6 [RFC6296]).

In some deployment contexts, the PCP Server may be reachable with an IPv4 address but DHCPv6 is used to provision the PCP Client. In such scenarios, a plain IPv4 address or an IPv4-mapped IPv6 address can be configured to reach the PCP Server.
A Dual-Stack host may receive OPTION_PCP_SERVER via both DHCPv4 and DHCPv6. The content of these OPTION_PCP_SERVER options may refer to the same or distinct PCP Servers. This is deployment-specific and as such it is out of scope of this document.

9. Security Considerations

The security considerations in [RFC2131], [RFC3315] and [I-D.ietf-pcp-base] are to be considered.

10. IANA Considerations

10.1. DHCPv6 Option

Authors of this document request the following DHCPv6 option code:

<table>
<thead>
<tr>
<th>Option Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>OPTION_PCP_SERVER</td>
<td>TBA</td>
</tr>
</tbody>
</table>

10.2. DHCPv4 Option

Authors of this document request the following DHCPv4 option code:

<table>
<thead>
<tr>
<th>Option Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>OPTION_PCP_SERVER</td>
<td>TBA</td>
</tr>
</tbody>
</table>

11. Acknowledgements

Many thanks to B. Volz, C. Jacquenet, R. Maglione, D. Thaler, T. Mrugalski and T. Lemon for their review and comments.

12. References

12.1. Normative References

[I-D.ietf-pcp-base]

12.2.  Informative References

[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-05 (work in
progress), November 2011.

[RFC1123]  Braden, R., "Requirements for Internet Hosts - Application

Stevens, "Basic Socket Interface Extensions for IPv6",

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

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


[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
Juniper Networks
1194 N Mathilda Avenue
Sunnyvale, California  94089
USA

Email: rpenno@juniper.net

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

Email: dwing@cisco.com
Universal Plug and Play (UPnP) Internet Gateway Device (IGD)-Port Control Protocol (PCP) Interworking Function
draft-ietf-pcp-upnp-igd-interworking-01

Abstract

This document specifies the behavior of the UPnP IGD (Internet Gateway Device)/PCP Interworking Function. An UPnP IGD-PCP Interworking Function (IGD-PCP IWF) is required to be embedded in CP routers to allow for transparent NAT control in environments where UPnP is used in the LAN side and PCP in the external side of the CP router.

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 September 13, 2012.

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 .............................................. 4
2. Acronyms .................................................. 4
3. Architecture Model ....................................... 5
4. UPnP IGD-PCP Interworking Function: Overview .............. 7
   4.1. UPnP IGD-PCP: State Variables ......................... 7
   4.2. IGD-PCP: Methods ..................................... 9
   4.3. UPnP IGD-PCP: Errors ................................ 10
5. Specification of the IGD-PCP Interworking Function .......... 12
   5.1. PCP Server Discovery ................................. 12
   5.2. Control of the Firewall .............................. 12
   5.3. NAT Control in LAN Side ............................ 12
   5.4. Port Mapping Tables .................................. 12
   5.5. Interworking Function Without NAT in the CP Router ... 13
   5.6. NAT Embedded in the CP Router ........................ 13
   5.7. Creating a Mapping ................................... 14
      5.7.1. AddAnyPortMapping() ............................ 14
      5.7.2. AddPortMapping() ................................ 15
   5.8. Listing One or a Set of Mappings ..................... 19
   5.9. Delete One or a Set of Mappings: DeletePortMapping()
        or DeletePortMappingRange() ...................... 19
6. IANA Considerations ...................................... 22
7. Security Considerations .................................. 23
8. Acknowledgments .......................................... 23
9. References ................................................ 23
9.1. Normative References ........................................ 23
9.2. Informative References ...................................... 23

Authors' Addresses .................................................. 24
1. Introduction

PCP [I-D.ietf-pcp-base] discusses the implementation of NAT control features that rely upon Carrier Grade NAT devices such as DS-Lite AFTR [RFC6333] or NAT64 [RFC6146]. Nevertheless, in environments where UPnP is used in the local network, an interworking function between UPnP IGD and PCP is required to be embedded in the CP router (see the example illustrated in Figure 1).

Two configurations are considered:

- No NAT function is embedded in the CP router. This is required for instance in DS-Lite or NAT64 deployments;

- The CP router embeds a NAT function.

```
<table>
<thead>
<tr>
<th>UPnP Control Point</th>
<th>UPnP-PCP Interworking Function</th>
<th>PCP Server</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1) AddPortMapping</td>
<td>(2) PCP MAP Request</td>
<td></td>
</tr>
</tbody>
</table>
```

Figure 1: Flow Example

The UPnP IGD-PCP Interworking Function (IGD-PCP IWF) maintains a local mapping table which stores all active mappings instructed by internal UPnP Control Points. This design choice restricts the amount of PCP messages to be exchanged with the PCP Server.

Triggers for deactivating the UPnP IGD-PCP Interworking Function from the CP router and relying on a PCP-only mode are out of scope of this document.

2. Acronyms

This document make use of the following abbreviations:
3. Architecture Model

As a reminder, Figure 2 illustrates the architecture model adopted by UPnP IGD [IGD2]. In Figure 2, the following UPnP terminology is used:

- **Client** refers to a host located in the local network.
- **IGD Control Point** is a UPnP control point using UPnP to control an IGD (Internet Gateway Device).
- **Host** represents a remote peer reachable in the Internet.

![Figure 2: UPnP IGD Model](image)

This model is not valid when PCP is used to control for instance a Carrier Grade NAT (a.k.a., Provider NAT) while internal hosts continue to use UPnP. In such scenarios, Figure 3 shows the updated model.
In the updated model depicted in Figure 3, one or two levels of NAT can be encountered in the data path. Indeed, in addition to the Carrier Grade NAT, the CP router may embed a NAT function (Figure 4).

To ensure a successful interworking between UPnP IGD and PCP, an interworking function is embedded in the CP router. In the model defined in Figure 3, all UPnP IGD server-oriented functions, a PCP Client [I-D.ietf-pcp-base] and a UPnP IGD-PCP Interworking Function are embedded in the CP router (i.e., IGD). In the rest of the document, IGD-PCP Interworking Function refers to PCP Client and UPnP IGD-PCP Interworking Function.

UPnP IGD-PCP Interworking Function is responsible for generating a well-formed PCP (resp., UPnP IGD) message from a received UPnP IGD (resp., PCP) message.
4. UPnP IGD-PCP Interworking Function: Overview

Three tables are provided to specify the mapping between UPnP IGD and PCP:

(1) Section 4.1 provides the mapping between WANIPConnection State Variables and PCP parameters;

(2) Section 4.2 focuses on the correspondence between supported methods;

(3) Section 4.3 lists the PCP error messages and their corresponding IGD ones.

Note that some enhancements have been integrated in WANIPConnection as documented in [IGD2].

4.1. UPnP IGD-PCP: State Variables

ConnectionType: Not applicable
Out of scope of PCP but as the controlled device is a NAT the default value IP_Routed is very likely used.

PossibleConnectionTypes: Not applicable
Out of scope of PCP (same comment than for ConnectionType).

ConnectionStatus: Not applicable
Out of scope of PCP but when it is possible to successfully communicate with a PCP Server the Connected value could be expected, otherwise Disconnected.

Uptime: Not applicable
Out of scope of PCP (possible values are the number of seconds since a successful communication was established with a PCP Server, or with a state maintained in a stable storage the number of seconds since the initialization of the current state).

LastConnectionError: Not applicable
Out of scope of PCP but expected to be ERROR_NONE in absence of errors.

RSIPAvailable: Not applicable
Out of scope of PCP (expected to be 0, i.e., RSIP not available).

ExternaIPAddress: External IP Address
Read-only variable with the value from the last PCP response or the empty string if none was received yet.
PortMappingNumberOfEntries: Not applicable
   Managed locally by the UPnP IGD-PCP Interworking Function.

PortMappingEnabled: Not applicable
   PCP does not support deactivating the dynamic NAT mapping since
   the initial goal of PCP is to ease the traversal of Carrier Grade
   NAT. Supporting such per-subscriber function may overload the
   Carrier Grade NAT.
   On reading the value should be 1, writing a value different from 1
   is not supported.

PortMappingLeaseDuration: Requested Mapping Lifetime
   In IGD:1 the value 0 means infinite, in IGD:2 its is remapped to
   the IGD maximum of 604800 seconds [IGD2]. PCP allows for a
   maximum value of 65535 seconds.
   The UPnP IGD-PCP Interworking Function simulates long and even
   infinite lifetimes using renewals. The behavior in the case of a
   failing renewal is currently undefined.
   IGD:1 doesn’t define the behavior in the case of state lost, IGD:2
   doesn’t require to keep state in stable storage, i.e., to make the
   state to survive resets/reboots. Of course the IGD:2 behavior
   should be implemented.

RemoteHost: Unsupported
   Not yet supported by PCP (part of the firewall features). Note a
   domain name is allowed by IGD:2 and has to be resolved into an IP
   address.

ExternalPort: External Port Number
   Not wildcard (0) value mapped to PCP external port field in MAP
   messages. The explicit wildcard (0) value is not supported.

InternalPort: Internal Port Number
   Mapped to PCP internal port field in MAP messages.

PortMappingProtocol: Transport Protocol
   Mapped to PCP protocol field in MAP messages. Note IGD only
   supports TCP and UDP.

InternalClient: Internal IP Address
   InternalClient can be an IP address or a domain name. Only an IP
   address scheme is supported in PCP. If a domain name is used
   Point, it must be resolved to an IP address by the Interworking
   Function when relying the message to the PCP Server.
PortMappingDescription: Not applicable
Not supported in base PCP. When present in UPnP IGD messages, this parameter SHOULD NOT be propagated in the corresponding PCP messages. If the local PCP Client support a PCP Option to convey the description, this option MAY be used.

SystemUpdateID (only for IGD:2): Not applicable
Managed locally by the UPnP IGD-PCP Interworking Function

A_ARG_TYPE_Manage (only for IGD:2): Not applicable
Out of scope of PCP (but has a clear impact on security).

A_ARG_TYPE_PortListing (only for IGD:2): Not applicable
Managed locally by the UPnP IGD-PCP Interworking Function

4.2. IGD-PCP: Methods

Both IGD:1 and IGD:2 methods are listed here.

SetConnectionType: Not applicable
Calling this method doesn’t make sense in this context. An error (IGD:1 501 "ActionFailed" or IGD:2 731 "ReadOnly") may be directly returned.

GetConnectionTypeInfo: Not applicable
May directly return values of corresponding State Variables.

RequestConnection: Not applicable
Calling this method doesn’t make sense in this context. An error (IGD:1 501 "ActionFailed" or IGD:2 606 "Action not authorized") may be directly returned.

ForceTermination: Not applicable
Same than RequestConnection.

GetStatusInfo: Not applicable
May directly return values of corresponding State Variables.

GetNATRSIPStatus: Not applicable
May directly return values of corresponding State Variables.

GetGenericPortMappingEntry: Not applicable
This request is not relayed to the PCP Server. IGD-PCP Interworking Function maintains an updated list of active mappings instantiated in the PCP Server by internal hosts. See Section 5.8 for more information.
GetSpecificPortMappingEntry: MAP with PREFER_FAILURE Option
This request is relayed to the PCP Server by issuing MAP with
PREFER_FAILURE Option. It is RECOMMENDED to use a short lifetime
(e.g., 60s).

AddPortMapping: MAP
We recommend the use of AddAnyPortMapping() instead of
AddPortMapping(). Refer to Section 5.7.2.

AddAnyPortMapping (for IGD:2 only): MAP
No issue is encountered to proxy this request to the PCP Server.
Refer to Section 5.7.1 for more details

DeletePortMapping: MAP with a requested lifetime set to 0
Refer to Section 5.9.

DeletePortMappingRange (for IGD:2 only): MAP with a lifetime
positioned to 0
Individual requests are issued by the IGD-PCP Interworking
Function. Refer to Section 5.9 for more details

GetExternalIPAddress: Not applicable
PCP does not support a method for retrieving the external IP
address. Issuing MAP may be used as a means to retrieve the
external IP address.
May directly return the value of the corresponding State Variable.

GetListOfPortMappings: Not applicable
The IGD-PCP Interworking Function maintains an updated list of
active mapping as instantiated in the PCP Server. The IGD-PCP
Interworking Function handles locally this request. See
Section 5.8 for more information

4.3. UPnP IGD-PCP: Errors
This section lists PCP errors codes and the corresponding UPnP IGD
ones. Error codes specific to IGD:2 are tagged accordingly.
1 UNSUPP_VERSION: 501 "ActionFailed"
   Should not happen.

2 NOT_AUTHORIZED: IGD:1 718 "ConflictInMappingEntry" / IGD:2 606
   "Action not authorized"
   729 "ConflictWithOtherMechanisms" is possible too.

3 MALFORMED_REQUEST: 501 "ActionFailed"

4 UNSUPP_OPCODE: 501 "ActionFailed"
   Should not happen.

5 UNSUPP_OPTION: 501 "ActionFailed"
   Should not happen at the exception of PREFER_FAILURE (this
   option is not mandatory to support but AddPortMapping() cannot be
   implemented without it).

6 MALFORMED_OPTION: 501 "ActionFailed"
   Should not happen.

7 NETWORK_FAILURE: Not applicable
   Should not happen after communication was successfully established
   with a PCP Server. Before the ConnectionStatus State Variable
   must not be set to Connected.

8 NO_RESOURCES: IGD:1 501 "ActionFailed" / IGD:2 728
   "NoPortMapsAvailable"
   Cannot be distinguished from USER_EX_QUOTA.

9 UNSUPP_PROTOCOL: 501 "ActionFailed"
   Should not happen.

10 USER_EX_QUOTA: IGD:1 501 "ActionFailed" / IGD:2 728
    "NoPortMapsAvailable"
    Cannot be distinguished from NO_RESOURCES.

11 CANNOT_PROVIDE_EXTERNAL: 718 "ConflictInMappingEntry"

12 ADDRESS_MISMATCH: 501 "ActionFailed"
   Should not happen.

13 EXCESSIVE_REMOTE_PEERS: 501 "ActionFailed"
5. Specification of the IGD-PCP Interworking Function

This section covers the scenarios with or without NAT in the CP router.

5.1. PCP Server Discovery

The IGD-PCP Interworking Function implements one of the discovery methods identified in [I-D.ietf-pcp-base] (e.g., DHCP [I-D.ietf-pcp-dhcp]). The IGD-PCP Interworking Function behaves as a PCP Client when communicating with the provisioned PCP Server.

In order to not impact the delivery of local services requiring the control of the local IGD during any failure event to reach the PCP Server (e.g., no IP address/prefix is assigned to the CP router), IGD-PCP Interworking Function MUST NOT be invoked. Indeed, UPnP machinery is used to control that device and therefore lead to successful operations of internal services.

5.2. Control of the Firewall

In order to configure security policies to be applied to inbound and outbound traffic, UPnP IGD can be used to control a local firewall engine.

No IGD-PCP Interworking Function is therefore required for that purpose.

5.3. NAT Control in LAN Side

Internal UPnP Control Points are not aware of the presence of the IGD-PCP Interworking Function in the CP router (IGD). Especially, UPnP Control Points MUST NOT be aware of the deactivation of the NAT in the CP router.

No modification is required in the UPnP Control Point.

5.4. Port Mapping Tables

IGD-PCP Interworking Function MUST store locally all the mappings instantiated by internal UPnP Control Points in the PCP Server. Port Forwarding mappings SHOULD be stored in a permanent storage.

Upon receipt of a PCP MAP Response from the PCP Server, the IGD-PCP Interworking Function MUST retrieve the enclosed mapping and MUST store it in the local mapping table. The local mapping table is an image of the mapping table as maintained by the PCP Server for a given subscriber.
5.5. Interworking Function Without NAT in the CP Router

When no NAT is embedded in the CP router, the content of received WANIPConnection and PCP messages is not altered by the IGD-PCP Interworking Function (i.e., the content of WANIPConnection messages are mapped to the PCP messages (and mapped back) according to Section 4.1).

5.6. NAT Embedded in the CP Router

Unlike the scenario with one level of NAT (Section 5.5), the IGD-PCP Interworking Function MUST update the content of received mapping messages with the IP address and/or port number belonging to the external interface of the CP router (i.e., after the NAT1 operation in Figure 4) and not as initially positioned by the UPnP Control Point.

All WANIPConnection messages issued by the UPnP Control Point (resp., PCP Server) are intercepted by the IGD-PCP Interworking Function. Then, the corresponding messages (see Section 4.1, Section 4.2 and Section 4.3) are generated by the IGD-PCP Interworking Function and sent to the provisioned PCP Server (resp., corresponding UPnP Control Point). The content of PCP messages received by the PCP Server reflects the mapping information as enforced in the first NAT. In particular, the internal IP address and/or port number of the requests are replaced with the IP address and port number as assigned by the NAT of the CP router. For the reverse path, PCP response messages are intercepted by the IGD-PCP Interworking Function. The content of the corresponding WANIPConnection messages are updated:

- The internal IP address and/or port number as initially positioned by the UPnP Control Point and stored in the CP router NAT are used to update the corresponding fields in received PCP responses.

- The external IP and port number are not altered by the IGD-PCP Interworking Function.

- The NAT mapping entry in the first NAT is updated with the result of PCP request.

The lifetime of the mappings instantiated in all involved NATs SHOULD be the one assigned by the terminating PCP Server. In any case, the lifetime MUST be lower or equal to the one assigned by the terminating PCP Server.
5.7. Creating a Mapping

Two methods can be used to create a mapping: AddPortMapping() or AddAnyPortMapping().

AddAnyPortMapping() is the RECOMMENDED method.

5.7.1. AddAnyPortMapping()

When an UPnP Control Point issues a AddAnyPortMapping(), this request is received by the UPnP Server. The request is then relayed to the IGD-PCP Interworking Function which generates a PCP MAP Request (see Section 4.1 for mapping between WANIPConnection and PCP parameters). Upon receipt of PCP MAP Response from the PCP Server, an XML mapping is returned to the requesting UPnP Control Point (the content of the messages follows the recommendations listed in Section 5.6 or Section 5.5 according to the deployed scenario). A flow example is depicted in Figure 5.

If a PCP Error is received from the PCP Server, a corresponding WANIPConnection error code Section 4.3 is generated by the IGD-PCP Interworking Function and sent to the requesting UPnP Control Point. If a short lifetime error is returned (e.g., NETWORK_FAILURE, NO_RESOURCES), the PCP IWF MAY re-send the same request to the PCP Server after 30s. If a negative answer is received, the error is then relayed to the requesting UPnP Control Point.

Justification: Some applications (e.g., uTorrent, Vuzz, Emule) wait approximately 150s, 90s, 90s, respectively for a response after sending an UPnP request. If a short lifetime error occurs, re-sending the requesting may lead to a positive response from the PCP Server. UPnP Control Points are therefore not aware of short lifetime errors that were recovered quickly.
5.7.2. AddPortMapping()

A dedicated option called PREFER_FAILURE is defined in
[I-D.ietf-pcp-base] to toggle the behavior in a PCP Request message.
This options is inserted by the IGD-PCP IWF when issuing its requests
to the PCP Server only if a specific external port is requested by
the UPnP Control Point. The mapping of wildcard (i.e., 0)
ExternalPort is not yet defined.

Upon receipt of AddPortMapping() from an UPnP Control Point, the IGD-
PCP Interworking Function first checks if the requested external port
number is not used by another Internal UPnP Control Point. In case a
mapping bound to the requested external port number is found in the
local mapping table, the IGD-PCP IWF MUST send back a
ConflictInMappingEntry error to the requesting UPnP Control Point
(see the example shown in Figure 6).
Some applications use `GetSpecificPortMapping()` to check whether a mapping exists.

Figure 6: IWF Local Behaviour

This exchange (Figure 6) is re-iterated until an external port number that is not in use is requested by the UPnP Control Point. Then, the IGD-PCP IWF MUST generate a PCP MAP Request with all requested mapping information as indicated by the UPnP Control Point if no NAT is embedded in the CP router or updated as specified in Section 5.6. In addition, the IGD-PCP IWF MUST insert a PREFER_FAILURE Option to the generated PCP request.

If the requested external port is in use, a PCP error message MUST be sent by the PCP Server to the IGD-PCP IWF indicating CANNOT_PROVIDE_EXTERNAL as the error cause. If a short lifetime error is returned, the PCP IWF MAY re-send the same request to the PCP Server after 30s. If a negative answer is received, the IGD-PCP IWF relays a negative message to the UPnP Control Point indicating ConflictInMappingEntry as error code. The UPnP Control Point may re-issue a new request with a new requested external port number. This process is repeated until a positive answer is received or maximum retry is reached.

If the PCP Server is able to honor the requested external port, a positive response is sent to the requesting IGD-PCP IWF. Upon

<table>
<thead>
<tr>
<th>UPnP Control Point</th>
<th>UPnP-PCP Interworking Function</th>
<th>PCP Server</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1) AddPortMapping ExternalPort=2356</td>
<td>&lt;---------------------</td>
<td></td>
</tr>
<tr>
<td>(2) Error: ConflictInMappingEntry</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(3) AddPortMapping ExternalPort=4586</td>
<td>&lt;---------------------</td>
<td></td>
</tr>
<tr>
<td>(4) Error: ConflictInMappingEntry</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
receipt of the response from the PCP Server, the returned mapping MUST be stored by the IGD-PCP Interworking Function in its local mapping table and a positive answer MUST be sent to the requesting UPnP Control Point. This answer terminates this exchange.

Figure 7 shows an example of the flow exchange that occurs when the PCP Server satisfies the request from the IGD-PCP IWF. Figure 8 shows the messages exchange when the requested external port is in use.

<table>
<thead>
<tr>
<th>UPnP Control Point</th>
<th>UPnP-PCP Interworking Function</th>
<th>PCP Server</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1) AddPortMapping</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ExternalPort=8080</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Protocol=TCP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------------------</td>
<td>-------------------------------</td>
<td>-------------</td>
</tr>
<tr>
<td></td>
<td>(2) PCP MAP Request</td>
<td></td>
</tr>
<tr>
<td></td>
<td>requested external port=8080</td>
<td></td>
</tr>
<tr>
<td></td>
<td>protocol=TCP</td>
<td></td>
</tr>
<tr>
<td></td>
<td>PREFER_FAILURE</td>
<td></td>
</tr>
<tr>
<td></td>
<td>------------------------------</td>
<td>-------------</td>
</tr>
<tr>
<td></td>
<td>(3) PCP MAP Response</td>
<td></td>
</tr>
<tr>
<td></td>
<td>assigned external port=8080</td>
<td></td>
</tr>
<tr>
<td></td>
<td>protocol=TCP</td>
<td></td>
</tr>
<tr>
<td></td>
<td>---------------------------------</td>
<td>-------------</td>
</tr>
<tr>
<td></td>
<td>(4) AddPortMapping</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ExternalPort=8080</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Protocol=TCP</td>
<td></td>
</tr>
<tr>
<td></td>
<td>---------------------------------</td>
<td>-------------</td>
</tr>
</tbody>
</table>

Figure 7: Flow Example (Positive Answer)
<table>
<thead>
<tr>
<th>UPnP Control Point</th>
<th>UPnP-PCP Interworking Function</th>
<th>PCP Server</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1) AddPortMapping ExternalPort=8080</td>
<td>(2) PCP MAP Request requested external port=8080 PREFER_FAILURE</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>(3) PCP MAP Response CANNOT_PROVIDE_EXTERNAL</td>
</tr>
<tr>
<td></td>
<td>(4) Error: ConflictInMappingEntry</td>
<td></td>
</tr>
<tr>
<td>(5) AddPortMapping ExternalPort=5485</td>
<td>(6) PCP MAP Request requested external port=5485 PREFER_FAILURE</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>(7) PCP MAP Response CANNOT_PROVIDE_EXTERNAL</td>
</tr>
<tr>
<td>(8) Error: ConflictInMappingEntry</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(a) AddPortMapping ExternalPort=6591</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(b) PCP MAP Request requested external port=6591 PREFER_FAILURE</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>(c) PCP MAP Response CANNOT_PROVIDE_EXTERNAL</td>
</tr>
<tr>
<td>(d) Error: ConflictInMappingEntry</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 8: Flow Example (Negative Answer)
Note: According to some experiments, some UPnP 1.0 implementations, e.g.- uTorrent, simply try the same external port X times (usually 4 times) and then fail.

5.8. Listing One or a Set of Mappings

In order to list active mappings, an UPnP Control Point may issue GetGenericPortMappingEntry(), GetSpecificPortMappingEntry() or GetListOfPortMappings().

GetGenericPortMappingEntry() and GetListOfPortMappings() methods MUST NOT be proxied to the PCP Server since a local mapping is maintained by the IGD-PCP Interworking Function.

Upon receipt of GetSpecificPortMappingEntry() from an UPnP Control Point, the IGD-PCP IWFMUST check first if the external port number is used by the requesting UPnP Control Point or another Internal UPnP Control Point. If the external port is already in use by the requesting UPnP Control Point, the IGD-PCP IWFMUST send back a positive answer. If the external port is already in use by another UPnP Control Point, the IGD-PCP IWFMUST send back a ConflictInMappingEntry error to the requesting UPnP Control Point. If no mapping is found in the local mapping table, the IWFMUST reply to the PCP Server a MAP request, with short lifetime (e.g. 60s), including a PREFER_FAILURE Option.

5.9. Delete One or a Set of Mappings: DeletePortMapping() or DeletePortMappingRange()

A UPnP Control Point proceeds to the deletion of one or a list of mappings by issuing DeletePortMapping() or DeletePortMappingRange(). In IGD:2, we assume the IGD applies the appropriate security policies to grant whether a Control Point has the rights to delete one or a set of mappings. When authorization fails, "606 Action Not Authorized" error code MUST be returned the requesting Control Point.

When DeletePortMapping() or DeletePortMappingRange() is received by the IGD-PCP Interworking Function, it first checks if the requested mappings to be removed are present in the local mapping table. If no mapping matching the request is found in the local table an error code is sent back to the UPnP Control Point: "714 NoSuchEntryInArray" for DeletePortMapping() or "730 PortMappingNotFound" for DeletePortMappingRange().

Figure 9 shows an example of UPnP Control Point asking to delete a mapping which is not instantiated in the local table of the IWFM.
Figure 9: Local Delete (IGD-PCP IWF)

If a mapping matches in the local table, a PCP MAP delete request is generated taking into account the input arguments as included in DeletePortMapping() if no NAT is enabled in the CP router or the corresponding local IP address and port number as assigned by the local NAT if a NAT is enabled in the CP router. When a positive answer is received from the PCP Server, the IGD-PCP Interworking Function updates its local mapping table (i.e., remove the corresponding entry) and notifies the UPnP Control Point about the result of the removal operation. Once PCP MAP delete request is received by the PCP Server, it proceeds to removing the corresponding entry. A PCP MAP delete response is sent back if the removal of the corresponding entry was successful; if not, a PCP Error is sent back to the IGD-PCP Interworking Function including the corresponding error cause (See Section 4.3).

In case DeletePortMappingRange() is used, the IGD-PCP IWF undertakes a lookup on its local mapping table to retrieve individual mappings instantiated by the requested Control Point (i.e., authorization checks) and matching the signalled port range (i.e., the external port is within "StartPort" and "EndPort" arguments of DeletePortMappingRange()). If no mapping is found, "730 PortMappingNotFound" error code is sent to the UPnP Control Point (Figure 10). If a set of mappings are found, the IGD-PCP IWF generates individual PCP MAP delete requests corresponding to these mappings (See the example shown in Figure 11).

The IWF MAY send a positive answer to the requesting UPnP Control Point without waiting to receive all the answers from the PCP Server. It is unlikely to encounter a problem in the PCP leg because the IWF has verified authorization rights and also the presence of the mapping in the local table.
Figure 10: Flow example when an error encountered when processing DeletePortMappingRange()
This example illustrates the exchanges that occur when the IWF receives DeletePortMappingRange(). In this example, only two mappings having the external port number in the 6000-6050 range are maintained in the local table. The IWF issues two MAP requests to delete these mappings.

```
 UPnP Control         Interworking Function
                       PCP Server
 (1)DeletePortMappingRange()
     StartPort=6000
     EndPort  =6050
     Protocol =UDP
-------------------------->

 (2a)PCP MAP Request
     protocol=UDP
     internal-ip-address
     internal-port
     external-ip-address
     external-port= 6030
     Requested-lifetime= 0
-------------------------->

 (2c)PCP MAP Request
     protocol=UDP
     internal-ip-address
     internal-port
     external-ip-address
     external-port= 6045
     Requested-lifetime= 0
-------------------------->

 (2b)Positive answer
 <------------------------
```

Figure 11: Example of DeletePortMappingRange()

6. IANA Considerations

This document makes no request of IANA.

Note to RFC Editor: this section may be removed on publication as an RFC.
7. Security Considerations

IGD:2 authorization framework SHOULD be used. When only IGD:1 is available, one MAY consider to enforce the default security, i.e., operation on the behalf of a third party is not allowed.

This document defines a procedure to instruct PCP mappings for third party devices belonging to the same subscriber. Identification means to avoid a malicious user to instruct mappings on behalf of a third party must be enabled. Such means are already discussed in Section 7.4.4 of [I-D.ietf-pcp-base].

Security considerations elaborated in [I-D.ietf-pcp-base] and [Sec_DCP] should be taken into account.

8. Acknowledgments

Authors would like to thank F. Fontaine, C. Jacquenet and X. Deng for their review and comments.

9. References

9.1. Normative References

[I-D.ietf-pcp-base]


9.2. Informative 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-02 (work in progress), January 2012.


Authors’ Addresses

Mohamed Boucadair
France Telecom
Rennes, 35000
France

Email: mohamed.boucadair@orange.com

Francis Dupont
Internet Systems Consortium

Email: fdupont@isc.org

Reinaldo Penno
Juniper Networks
1194 N Mathilda Avenue
Sunnyvale, California 94089
USA

Email: rpenno@juniper.net

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

Email: dwing@cisco.com
PCP Server Discovery with IPv4 traffic offload for Proxy Mobile IPv6
draft-rpcw-pcp-pmipv6-serv-discovery-00

Abstract

This document proposes a solution to PCP Server Discovery problems in
Proxy Mobile IPv6 (PMIPv6) networks when both home network traffic
and traffic off-loaded to local access network require traversing a
gateway implementing NAT and/or Firewall. This draft proposes
enhancements to DHCPv4 Relay Agent by introducing a new sub-option
under DHCPv4 Relay Option and to PMIPv6 signaling through additional
options to Proxy Binding Update/Acknowledgement 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 September 1, 2012.

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
Table of Contents

1. Introduction .............................................. 3
2. Terminology .............................................. 3
3. Solution overview ......................................... 4
4. Mobility Options ........................................... 6
5. DHCPv4 Relay Agent co-located with MAG .................. 8
   5.1. Format .................................................. 8
   5.2. Relay Agent behavior ................................... 9
   5.3. DHCPv4 Server behavior ............................... 9
6. DHCPv4 Server co-located with MAG ....................... 9
7. Security Considerations .................................... 10
8. IANA Considerations ....................................... 10
9. Acknowledgements .......................................... 11
10. References ................................................ 11
    10.1. Normative References ................................. 11
    10.2. Informative References .............................. 12
11. Authors’ Addresses ........................................ 12
1. Introduction

Given the exponential growth in the mobile data traffic, Mobile Operators are looking for ways to offload some of the IP traffic flows at the nearest access edge that has an Internet peering point. This approach results in efficient usage of the mobile packet core and helps lower the transport cost.

[I-D.ietf-netext-pmipv6-sipto-option] defines a way to signal the Traffic Offload capability of a Mobile Access Gateway (MAG) to the Local Mobility Anchor (LMA) in Proxy Mobile IP Networks. There are scenarios in PMIPv6 Mobile Networks where the traffic going through the Mobile Packet Core as well as the traffic that is off-loaded to the Local Access Networks end up going through a NAT or Firewall gateway. If the mobile node applications desire to find or control the external addresses assigned to the internal address used by the Mobile Node (MN), it could be achieved by having a Port Control Protocol (PCP) Client on the mobile node.

[I-D.ietf-pcp-dhcp] specifies DHCP (IPv4 and IPv6) options to configure hosts with Port Control Protocol (PCP) Server addresses. However, PCP Client on the mobile node will not know whether a flow will traverse the Mobile Packet Core or will get offloaded at the local access network and hence will not know which PCP server to send its queries to. Even if the mobile node tries to find its PCP server using DHCP, it may only find out about the PCP server in the Home Network since the source of information is the DHCP server in the Home Network. The mobile node may never learn the presence of the PCP server in the Local Access Network. This requires mobile access gateway to act as a smart PCP Proxy for the PCP servers in both the mobile node’s home network as well as in the Local Access Network. However, this alone does not solve this problem since the mobile node needs to be informed of the PCP proxy on the MAG. This draft proposes an extension to DHCPv4 Relay Information Option and PMIPv6 Options to achieve these objectives.

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].

All the mobility related terms used in this document are to be interpreted as defined in the Proxy Mobile IPv6 specifications [RFC5213], [RFC5844]. This note also uses terminology defined in [I-D.ietf-pcp-base].

Additionally, this document uses the following abbreviations:
3. Solution overview

The following illustrates a scenario where the Mobile Node is a PCP client, Mobile Access Gateway in the access network is a PCP server with smart PCP proxy functionality [I-D.bpw-pcp-proxy], Local Mobility Anchor in the home network has a PCP server and PCP proxy functionality. The assumption made for this specification is Mobile Access Gateway is always co-located with NAT.

Mobile access gateway has the ability to offload some of the IPv4 traffic flows based on the traffic selectors it receives from the local mobility anchor. Using IP Traffic Offload Selector option [I-D.ietf-netext-pmipv6-sipto-option] mobile access gateway will negotiate IP Flows that can be offloaded to the local access network. For example, consider a mobile node acting as both client and server for FTP, VoIP and P2P. In this case FTP flows for that mobility...
session may be offloaded at the mobile access gateway and P2P, Voice
over IP (VoIP) flows tunneled back to the local mobility anchor.
Mobile node uses PCP to create mappings between external IP address/
port and internal IP address/port. These mappings will be used for
successful inbound communication destined to the mobile node behind
NAT and/or firewall.

The mobile node learns the PCP server domain name from DHCPv4 server
using DHCPv4 option OPTION_PCP_SERVER [I-D.ietf-pcp-dhcp]. If IP
Flows are offloaded at the mobile access gateway then the mobile node
needs to learn the domain name of the mobile access gateway acting as
smart PCP proxy. Mobile access gateway will compare the Remote Peer
IP Address and Port fields set in PCP PEER request from the mobile
node with the Traffic Selector fields and IP Traffic Offload Mode
Flag in IP Traffic Offload Selector Option to determine if the
dynamic outbound mapping is to be created in the local access network
or home network. In case of PCP MAP request mobile access gateway
will compare the Remote Peer IP Address and Port fields in FILTER
Option with the Traffic Selector fields and IP Traffic Offload Mode
Flag in IP Traffic Offload Selector Option to determine if dynamic
outbound mapping is to be created in the local access network or home
network. For PCP MAP request without FILTER option since the Remote
Peer IP Address is not available the mobile access gateway will
function as smart PCP proxy and forward the PCP MAP request to the
PCP server in the home network.

If the dynamic outbound mapping is for the local access network then
there are two cases to consider - In the first case where there is a
nested NAT[I-D.penno-ppc-nested-nat], the mobile access gateway will
function as both PCP server and PCP proxy forwarding the accepted PCP
request to CGN PCP server. In the second case, where there is no
CGN, mobile access gateway will function as a PCP server in the local
access network.

If dynamic outbound mapping is for the home network then mobile
access gateway will function as smart PCP proxy and forward the
accepted PCP request to the PCP server in the home network.
4. Mobility Options

A new mobility option, Capability Exchange Option is defined for use with Proxy Binding Update sent by the mobile access gateway to the local mobility anchor. The option is used for conveying device capabilities such as PCP Server, smart PCP Proxy.
Figure 2: Capability Exchange Option

Type: <IANA-1>

Length: An 8-bit unsigned integer indicating the length of the option in octets, excluding the Type and Length fields. This field MUST be set to 2.

Reserved (R): This 14-bit field is unused for now. The value MUST be initialized to (0) by the sender and MUST be ignored by the receiver.

PCP Server Support Mode (S): A 1-bit field that specifies the PCP server support mode. The flag value of (1) indicates that mobile access gateway is capable of functioning as PCP Server to the Mobile node.

PCP Proxy Support Mode (P): A 1-bit field that specifies the smart PCP proxy support mode. The flag value of (1) indicates that mobile access gateway is capable of functioning as smart PCP Proxy to the Mobile node.

A new mobility option, PCP Server Option is defined for use with Proxy Binding Acknowledgement sent by the local mobility anchor to the mobile access gateway. The option is used to provide PCP server domain name of the home network to the mobile access gateway.

Figure 3: PCP Server Option
Type:  <IANA-2>

Length: An 8-bit unsigned integer indicating the length of the option in octets, excluding the Type and Length fields.

Reserved (R): This 16-bit field is unused for now.

PCP Server Domain Name: The domain name of the PCP Server to be used by the mobile access gateway. The domain name is encoded as specified in Section 8 of [RFC3315].

5. DHCPv4 Relay Agent co-located with MAG

When DHCPv4 Relay Agent is co-located with the mobile access gateway, the proposal is for the relay agent to influence the DHCPv4 Server to opt for the PCP server domain name proposed by the Relay Agent over the one configured on the DHCPv4 Server. The DHCPv4 Relay Agent will insert a new suboption under relay agent information option indicating the domain name of the appropriate PCP server/proxy only after successful Tunnel/Route setup. For this to happen, the MN MUST ensure that it includes OPTION_PCP_SERVER in the Parameter Request List Option in the DHCPv4 Discover/Request message. The mobile access gateway will also have to act as a smart PCP-Proxy in this case so that it can handle PCP Servers of both the local access network and the home network. This will ensure that the right PCP Server is picked by the proxy based on IP Flow.

5.1. Format

To realize the mechanism described above, the document proposes a new PCP Server suboption for the DHCPv4 relay agent information option that carries the domain name of PCP Server/Proxy.
Code: TBA

Length: Includes the length of the "PCP Server Domain Name" field in octets; The maximum length is 255 octets.

PCP Server Domain Name: The domain name of the PCP Server to be used by the PCP Client when issuing PCP messages. The domain name is encoded as specified in Section 8 of [RFC3315].

5.2. Relay Agent behavior

DHCPv4 relay agents MAY be configured to include a PCP Server suboption if they include a relay agent information option in relayed DHCPv4 messages. The PCP Server Domain name is assigned and configured through mechanisms that are outside the scope of this memo.

5.3. DHCPv4 Server behavior

This suboption provides additional information to the DHCP server. Upon receiving a DHCPv4 Discover/Request containing the suboption, the DHCPv4 server, if configured to support this suboption, MUST populate the DHCPv4 Offer/Ack with the suggested PCP server domain name overriding any other PCP server domain name configuration that it may already have. There is no special additional processing for this suboption.

6. DHCPv4 Server co-located with MAG

When the DHCPv4 Server is co-located with the mobile access gateway, the DHCPv4 Server will have to provide the appropriate PCP server domain name in the DHCP Offer/Ack based on traffic offload negotiation between the mobile access gateway and local mobility anchor.

If traffic offload is successfully negotiated between the mobile access gateway and the local mobility anchor, the proposal is for the DHCPv4 Server to include the domain name of the PCP Proxy in the DHCP Offer/Ack. The mobile access gateway will act as a smart PCP-Proxy in this case to ensure that it can handle PCP Servers of both the local access network and the home network. This will ensure that the right PCP Server is picked by the proxy based on IP Flow.
If traffic offload is not negotiated between the mobile access gateway and the local mobility anchor, the proposal is for the DHCPv4 Server to include the domain name of the home network PCP server in the DHCPv4 Offer/Ack. The domain name of the PCP server in the home network is obtained from Proxy Binding message exchange explained in Section 4. Option OPTION_PCP_SERVER will be used as described in [I-D.ietf-pcp-dhcp].

7. Security Considerations

The Capability Exchange option defined in this specification is for use in Proxy Binding Update messages. The PCP server option defined in this specification is for the Proxy Binding Acknowledgement messages. These options are carried like any other mobility header option as specified in [RFC5213] and does not require any special security considerations. When IPv4 traffic offload support is enabled for a mobile node, the mobile access gateway selectively offloads some of the mobile node's traffic flows to the local access network. Typically, these offloaded flows go through a NAT gateway and that essentially introduces certain vulnerabilities which are common to any NAT deployment. These vulnerabilities and the related considerations have been well documented in the NAT specification [RFC2663]. There are no additional considerations above and beyond what is already documented by the NAT specifications and which are unique to the approach specified in this document.

The security considerations in [I-D.ietf-pcp-base], [I-D.bpw-pcp-proxy] and section 5 of [RFC3046] also apply to this use.

8. IANA Considerations

This specification defines two new Mobility Header options -
Capability Exchange option, PCP server option. These options are described in Section 4. The Type value for this option needs to be assigned from the same numbering space as allocated for the other mobility options [RFC6275].

IANA is requested to assign a suboption number for the PCP Server Suboption from the DHCP Relay Agent Information Option [RFC3046] suboption number space.

9. Acknowledgements

The authors would like to thank Sri Gundavelli for his valuable comments.

10. References

10.1. Normative References

[I-D.bpw-pcp-proxy]

[I-D.ietf-netext-pmipv6-sipto-option]

[I-D.ietf-pcp-base]

[I-D.ietf-pcp-dhcp]
Boucadair, M., Penno, R., and D. Wing, "DHCP Options for the Port Control Protocol (PCP)", draft-ietf-pcp-dhcp-02 (work in progress), January 2012.

[I-D.penno-pcp-nested-nat]

[RFC2119]  Bradner, S., "Key words for use in RFCs to Indicate
10.2. Informative References


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

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

Email: praspati@cisco.com
Ravikumar Chandrasekaran  
Cisco Systems, Inc.  
Cessna Business Park, Varthur Hobli  
Sarjapur Marthalli Outer Ring Road  
Bangalore, Karnataka 560103  
India  
Email: sravikum@cisco.com

Dan Wing  
Cisco Systems, Inc.  
170 West Tasman Drive  
San Jose, California 95134  
USA  
Email: dwing@cisco.com
Abstract

This document defines an extension to the base PCP. New OpCode and Options are 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 September 14, 2012.

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. Port-Set Options Formats ............................ 5
   2.2.1. Port_Range_Option ............................... 6
   2.2.2. Cryptographically_Random_Port_Range_Option .... 6
2.3. Generating a MAP_PORT_SET Request ................. 8
2.4. Renewing a MAP_PORT_SET Mapping .................. 8
2.5. Processing a MAP_PORT_SET Request ................. 8
2.6. Processing a MAP_PORT_SET Response ................. 9
2.7. Mapping Lifetime and Deletion ....................... 9
2.8. PREFER_FAILURE Option for MAP_PORT_SET Opcode ... 9
3. Security Considerations ............................... 9
4. IANA Considerations ................................ 9
5. Contributor List ..................................... 9
6. References ........................................... 10
   6.1. Normative References .............................. 10
   6.2. informative References ............................ 10
   Authors’ Addresses ................................. 10
1. Application Scenario

PCP can be used to control an upstream device to achieve the following goals:

1. A plain (i.e., a non-shared) IP address can be assigned to a given subscriber because the subscriber 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 set instead of individual mapping. This is motivated by the need to offload to a port-restricted device, reduce the logging and enhance the performance of the CGN.

A new PCP OpCode and two new PCP Options are defined in this document.

2. MAP_PORT_SET Opcode

This section defines a new OpCode which requests port set from a PCP-controlled device to a PCP client. 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 port set. It 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 instance to lightweight 4over6 [14over6] in port-set allocation process.

MAP_PORT_SET: Create an explicit dynamic mapping between an Internet Address and an External Address + Port set
The format of a port-set can either be contiguous or non-contiguous including a cryptographically assigned port set. The contiguous port-set is simple but since the port space for a subscriber shrinks significantly, the randomness for the port numbers is decreased significantly. This may allow an attacker to guess the port number used. Non-contiguous port-set, e.g., cryptographical algorithm [RFC6431], can be provided to improve the randomness of port number. It may be used as a mitigation tool against blind attacks. Therefore, in MAP_PORT_SET Opcode, it is mandatory to support two port-set options: PORT_MASK Option and Cryptographically_Random_Port_set Option. Besides, PREFERE_FAILURE Option would also apply for MAP_PORT_SET Opcode.

PCP-controlled device SHOULD provide a configuration option to allow administrators to configure the size of the port set to be assigned and whether cryptographical option is supported or not.

2.1. MAP_PORT_SET Operation Packet Formats

The MAP_PORT_SET Opcode has a similar packet layout for both requests and response. The following figure shows the format of the Opcode in a request 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Protocol    |          Reserved (24 bits)                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|           Suggested External IP Address (128 bits)            |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
0                   1                   2                   3
```

Figure 1: MAP_PORT_SET Opcode format of Request

These fields are described below:

- **Protocol**: the default value is zero (to indicate all transport protocols).

- **Reserved bits**: 24 bits MUST be set to 0.

- **Suggested External IP Address**: Suggested external IPv4 or IPv6 address. Same as Section 10.1 of [PCP-base].
The following figure shows the format of Opcode-specific information in a response packet for the MAP_PORT_SET Opcode:

```
+--------+--------+--------+--------+
<table>
<thead>
<tr>
<th>Protocol</th>
<th>Reserved</th>
<th>Assigned</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Protocol</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Assigned</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

Figure 2: MAP_PORT_SET Opcode format of Response

These fields are described below:

- **Protocol**: MUST be copied from the request.
- **Reserved bits**: 24 bits MUST be set to 0.
- **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.
- **Requested lifetime (in common header)**: Requested lifetime for the whole port-set mapping, in seconds. The value 0 also indicates "delete" here.

Discussion note: Assess further whether THIRD_PARTY Option is needed for PORTRANGE OpCode.

### 2.2. Port-Set Options Formats

The Port-Set options are used to specify one set of ports pertaining to a given IP address. As defined in [RFC6431], there are three kinds of port range: contiguous, non-contiguous and random. A cryptographically random Port Range Option may be used as a mitigation tool against blind attacks. We will describe the two port set PCP options in this section.
2.2.1. Port_Range_Option

<table>
<thead>
<tr>
<th>0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>TBA</td>
</tr>
<tr>
<td>-------------------------------</td>
</tr>
<tr>
<td>Port Range Value</td>
</tr>
<tr>
<td>-------------------------------</td>
</tr>
</tbody>
</table>

Figure 3: Port_Range_Option

- **Port Range Value (PRV):** The PRV indicates the value of the significant bits of the Port Mask. By default, no PRV is assigned.

- **Port Range Mask (PRM):** The Port Range Mask indicates the position of the bits that are used to build the Port Range Value. By default, no PRM value is assigned. The 1 values in the Port Range Mask indicate by their position the significant bits of the Port Range Value.

This option:

- **name:** Port range option

- **number:** TBA

- **purpose:** A PCP Client inserts this option in a PCP request to specify one set of ports (contiguous or not contiguous) pertaining to a given IP address.

- **is valid for OpCodes:** MAP_PORT_SET.

- **length:** 4 octets

- **may appear in:** request and response

- **maximum occurrences:** 1

2.2.2. Cryptographically_Random_Port_Range_Option

The cryptographically random Port Range PCP Option is formatted as below.
Figure 4: Cryptographically Random Port Range Option

- function/starting point/number of delegated ports/k: In request packet, it is the suggested function/starting point/number of delegated ports/k which might be helpful for refreshing a mapping after the PCP server loses state. For a success response packet, it is the assigned function/starting point/number of delegated ports/k, while for an error response packet, it is copied from the request.

This option:

- name: Cryptographically Random Port Range Option
- number: TBA
- purpose: A PCP Client inserts this option in a PCP request to specify one set of random ports pertaining to a given IP address. The random ports can be achieved by defining a function that takes as input a key ‘K’ and an integer ‘x’ within the 1024-65535 port range and produces an output ‘y’ also within the 1024-65535 port range.
- is valid for OpCodes: MAP_PORT_SET.
- length: 24 octets.
- may appear in: request and response
- maximum occurrences: 1
2.3. Generating a MAP_PORT_SET Request

The request MAP_PORT_SET MUST contains one of the port-set options, either PORT_RANGE option or Cryptographically_Random_Port_Set option. The request MAY contain values in the Suggested IP Address field and corresponding parameters in PORT_RANGE option. 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 free port set.

2.4. Renewing a MAP_PORT_SET Mapping

The similar actions defined in PCP-BASE specification [section 10.2.1 of [ID.ietf-pcp-base]] can be applied to MAP_PORT_SET Opcode to extend the lifetime of a port-set mapping. The MAP_PORT_SET request SHOULD include the currently assigned IP address and port-set in the suggested IP address and port-set options. The PCP-client should renew the port-set mapping before its expiry time.

[Discussion note: Do we need to cover the case in which a client MAY send a request to the LSN for another delegated set of ports? I think we should recommend that the same external IP address. Otherwise, the client has to determine which address should be used in actual data communication process among multiple external addresses.]

2.5. Processing a MAP_PORT_SET Request

The procedures regarding to lifetime is similar to the single port processes in MAP Opcode [section 10.3 of [ID.ietf-pcp-base]], except that the whole port-set should be treated consistently in MAP_PORT_SET Opcode.

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 the PCP server or PCP-controlled device does not support the port-set option indicated in MAP_PORT_SET request, the error UNSUPP_OPTION MUST be returned.

- If an option does not make sense, (e.g., the PREFER_FAILURE Option is included in a request with lifetime=0, or MAP_PORT_SET Opcode does not include port-set options, etc.), the request is invalid and generates a MALFORMED_OPTION error. This procedure is the same with section 10.3 of [ID.ietf-pcp-base].
If the MAP request contains the PREFER_FAILURE Option, but the Suggested External Address and Port-set do not match the External Address and Port of the existing mapping, the PCP server MUST return CANNOT_PROVIDE_EXTERNAL.

If all of the preceding operations were successful (did not generate an error response), then the requested port-set mapping is created or refreshed as described in the request and a SUCCESS response is built. The assigned external IPv4 (encoded using IPv4-mapped IPv6 address) or IPv6 address for the mapping should be returned.

2.6. 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 [ID.ietf-pcp-base]] should be followed by the PCP Client to validate the response (except the considerations related to the internal port).

2.7. Mapping Lifetime and Deletion

The procedure for port-set mapping lifetime and deletion is also the same with individual mapping [section 10.5 of [ID.ietf-pcp-base]].

2.8. PREFER_FAILURE Option for MAP_PORT_SET Opcode

This option [section 10.2 of [ID.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.

3. Security Considerations

None.

4. IANA Considerations

The authors request the following new OpCode: MAP_PORT_SET and the following two Options: PORT RANGE Cryptographically_Random_Port_Set

5. Contributor List

Gabor Bajko
Nokia
6. References

6.1. Normative References

[ID.ietf-pcp-base]
Wing, D., "Port Control Protocol (PCP)", February 2012.

[RFC6431]  IETF, "Huawei Port Range Configuration Options for PPP IP
Control Protocol (IPCP)", November 2011,

6.2. Informative References

[ID.behave-natx4-log-reduction]
Tsou, T., Li, W., and T. Taylor, "Port Management To
Reduce Logging In Large-Scale NATs", September 2010.

[ID.cui-softwire-b4-translated-ds-lite]
Cui, Y., Sun, Q., Boucadair, M., Tsou, T., and Y. Lee,
"Lightweight 4over6: An Extension to DS-Lite

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-ftgroup.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
Port Control Protocol (PCP) Authentication Mechanism
draft-wasserman-pcp-authentication-02

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 communications 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 proposes 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.

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 September 12, 2012.

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
Table of Contents

1. Introduction .................................................. 3
2. Terminology .................................................. 4
3. Protocol Details .............................................. 4
   3.1. Session Initiation ........................................ 5
   3.2. Session Termination ....................................... 7
   3.3. Result Codes .............................................. 7
4. PA Security Association ....................................... 7
5. Packet Format ................................................ 8
   5.1. Authentication OpCode Format ............................. 8
   5.2. Nonce Option ............................................. 9
   5.3. Authentication Tag Option ............................... 10
   5.4. EAP Payload Option ....................................... 11
   5.5. PRF Option ............................................... 11
   5.6. Hash Algorithm Option ................................... 11
   5.7. Session Lifetime Option .................................. 12
6. Processing Rules .............................................. 12
   6.1. Authentication Data Generation ......................... 12
   6.2. Authentication Data Validation ......................... 12
   6.3. Sequence Number ......................................... 13
   6.4. Retransmission Policies ................................. 13
   6.5. MTU Considerations ...................................... 14
7. IANA Considerations ......................................... 15
8. Security Considerations ..................................... 15
9. Acknowledgements ............................................ 15
10. Change Log ................................................... 15
    10.1. Changes from -00 to -01 ............................... 16
    10.2. Changes from -01 to -02 ............................... 16
11. References .................................................. 16
    11.1. Normative References .................................. 16
    11.2. Informative References ............................... 16
Authors’ Addresses .............................................. 17
1. Introduction

Using the Port Control Protocol (PCP) [I-D.ietf-pcp-base], an IPv4 or IPv6 host 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 unauthorized users from accessing such devices. This document proposes a PCP security extension which enables PCP servers to authenticate the clients that they are communicating with using Extensible Authentication Protocol (EAP). The following issues are considered in the design of this extension:

- Loss of EAP messages during transportation
- Disordered delivery of EAP messages
- Generation of transport keys
- Integrity protection and data origin authentication for PCP messages
- Algorithm agility

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

- On security infrastructure equipment, such as corporate firewalls, that does not create implicit mappings.
- 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.
- For any implementation that wants to be more permissive in authorizing explicit mappings than it is in authorizing implicit mappings.
- For implementations that support the THIRD_PARTY Option (unless they can meet the constraints outlined in Section 14.1.2.2).
- For implementations that wish to support any deployment scenario that does not meet the constraints described in Section 14.1.
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 [I-D.ietf-pcp-base].

PCP Client (PCC): A PCP device (e.g., a host) which is responsible for issuing PCP requests to a PCP server. In this document, a PCC is also a EAP peer [RFC3748], and it is the responsibility of a PCC to provide the credentials when authentication is required.

PCP Server (PCS): A PCP device (e.g., a NAT or a firewall) that implements the server-side of the PCP protocol, via which PCCs request and manage explicit mappings. In this document, a PCS is integrated with an EAP authenticator [RFC3748]. Therefore, when necessary, a PCS can verify the credentials provided by a PCC and make an access control decision based on the authentication result.

PCP Authentication (PA) Session: A series of PCP message exchanges transferred between a PCC and a PCS in order to perform authentication, authorization, key distribution and secured PCP communication. Each PA session is assigned a distinctive Session ID. The PCP devices involved within a PA session are called session partners. A typical PA session has two session partners.

Session Lifetime: The life period associated with a PA session, which decided the lifetime of the current authorization given to the PCC.

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

Master Session Key (MSK): A key derived by the partners of a PA session, using a EAP key generating method specified in [RFC3748].


3. Protocol Details
3.1. Session Initiation

To carry out an EAP authentication process between two PCP devices, a set of PA messages need to be exchanged. Each PA message contains an Authentication OpCode (and additional Options if needed). The Authentication OpCode consists of four fields: Session ID, Flag, EAP Type, and Sequence Number. The Session ID field is used to identify the session which the message belongs to. The Flag field indicates the type of the PCP message, while EAP Type is used to identify the type of the attached EAP message. The sequence number field is used to detect the disorder or the duplication occurred during packet delivery.

The message exchanges conveyed within an PA session is introduced in the remainder section.

When a PCC intends to initiate a PA session with a PCS, it sends a PCC-Initiation message to the PCS. The Session ID and Sequence Number fields of the Authentication OpCode in the PCC-Initiation message are set as 0, and the I bit is set. the PCC also needs to select a random nonce and append it with the PCC-Initiation message in order to deal with off-line attacks. Specifically, the nonce is transported within a nonce option. After receiving the PCC-Initiation, if the PCS would like to initiate a PA session, it will reply with a PA-Request which contains an EAP Identity Request. The Sequence Number field in the PA-Request is set as 0, and the Session ID field MUST be filled with the session identifier assigned by the PCS for this session. the PA-Request also needs to be attached with the nonce. Form now on, every PA message within this session must be attached with the session identifier. Otherwise, the session partner receiving the message will discard the message silently. If the PCC intends to simplify the authentication process, it can append an EAP Identity Response message within the PCC-Initiation request so as to skip over the step of waiting for the EAP Identity Request and inform the PCS that it would like to perform EAP authentication.

In the scenario where a PCS receives a non-PA PCP message from a PCC which needs to be authenticated, the PCS can reply with a PA-Request to initiate a PA session; the result code field of the PA-Request is set as AUTHENTICATION-REQUIRED. In addition, the PCS MUST assign a session ID for the session and transfer it within the PA-Request. In the PA messages exchanged afterwards in this session, the session ID MUST be appended. Therefore, in the subsequent communication, the PCC can distinguish the messages in this session from those in other sessions through the PCS IP address and the session ID. When the PCC receives the initial PA-Request message from the PCS, it can reply with a PA-Answer message to continue the session or silently discards the request message according to its local policies.
In a PA session, PA-Request messages are sent from PCSs to PCCs while PA-Answer messages are only sent from PCCs to PCSs. Correspondently, an EAP request messages MUST be transported within a PA-Request message, and an EAP answer messages MUST be transported within a PA-Answer message. Particularly, when a PCP device receives a PA-Request or a PA-Answer message from its partner, the PCP device needs to reply with a PA-Acknowledge message to indicate that the message has been received. This solution is used to deal with the conditions where the device cannot generate a response within a pre-specified period due to certain reasons (e.g., waiting for human input to construct a EAP message). Therefore, the partner does not have to unnecessarily retransfer the PCP message.

In this work, it is mandated for a PCC and a PCS to perform a key-generating EAP method in authentication, and so a successful EAP authentication process will result in a Master Session Key (MSK). If the PCC and the PCS want to generate a traffic 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 packets. On this occasion, the PCS needs to append the initial PA-Request message with a set of PRF Options and MAC Algorithm Options. Each PRF Option (MAC Algorithm Option) contains a PRF (MAC (Message Authentication Code) algorithm) that the PCS supports. After receiving the request, the PCC selects a PRF and a MAC algorithm which it intends to support, and sends back a PA-Answer with a PRF Option and a MAC Algorithm Option for the selected algorithms.

The last PA-Request message transported within a PA session carries the EAP authentication and PCP authorization results. The last PA-Request and PA-Answer messages MUST have their the 'C' (Complete) bit set.

If the EAP authentication successes, the result code of the last PA-Request is AUTHENTICATION-SUCCESS. In this case, before sending out the PA-Request, the PCS must derive a transport key and use it to generate digests to protect the integrity and authenticity of the PA-Request and any subsequent PCP message. Such digests are transported within Authentication Tag Options. In addition, the PA-Request needs to be appended with a Session Lifetime Option which indicates the life time of the PA session (i.e., the life time of the MSK).

If the EAP authentication fails, the result code of the last PA-Request is AUTHENTICATION-FAILED. If the EAP authentication succeeds but Authorization fails, the result code of the last PA-Request is AUTHORIZATION-FAILED. In the latter two cases, the PA session MUST be terminated immediately after the last PCP authentication message exchange.
3.2. Session Termination

A PA session can be explicitly terminated by sending a termination-indicating PA acknowledge message from either session partner. After receiving a termination-indicating message from the session partner, a PCP device MUST response with a termination-indicating PA Acknowledge message and remove the PA SA immediately. When the session partner initiating the termination process receives the acknowledge message, it will remove the associated PA SA immediately.

3.3. Result Codes

Following result codes are defined in the solution:

- XXX AUTHENTICATION-REQUIRED
- XXX AUTHENTICATION-FAILED
- XXX AUTHENTICATION-SUCCESS
- XXX AUTHORIZATION-FAILED

4. PA Security Association

At the beginning a PA session, a session SHOULD generate a PA SA to maintain its state information during the session. The parameters of a PA SA is listed as follows:

- IP address and UDP port number of the PCC
- IP address and UDP port number of the PCS
- Session Identifier
- Sequence number for the next outgoing PCP message
- Sequence number for the next incoming PCP message
- Last outgoing message payload
- Retransmission interval
- MSK
- MAC algorithm: The algorithm that the transport key should use to generate digests for PCP messages.
o Pseudo-random function: The pseudo random function negotiated in the initial PA-Request and PA-Answer exchange for the transport key derivation.

o Transport key: the key derived from the MSK to provide integrity protection and data origin authentication for the messages in the PA session. The life time of the transport key SHOULD be identical to the life time of the session.

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

o The 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 Session_ID: The ID of the session which the MSK is derived from

o Key ID: The ID assigned for the traffic key

5. Packet Format

5.1. Authentication OpCode Format

The following figure illustrates the format of an 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|             Flags             |     EAP Message Type       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Session ID                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     Sequence Number                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Flags: The Flags field is two octets. The following bits are assigned:
* I (Initiation): This bit is set in a PCC-Initiation message.

* C (Complete): If the message is the last PA-Request or PA-
Answer message in the session, this bit MUST be set. For other
messages, this bit MUST be cleared.

* R (Request): This bit is set in a PA-Request message.

* A (Answer): This bit is set in a PA-Answer message.

* K (acknowledgement): This bit is set and only set in a PA-
Acknowledgement message.

* T (Termination): If this bit is set in a PA-Acknowledgement
message, the message is used for session-termination
indication.

* S (Fragmentation start): This bit is set in a PA message which
contain the first fragment of a EAP message.

* E (Fragmentation start): This bit is set in a PA message which
contain the last fragment of a EAP message.

Message Type: The Message Type field is two octets. This field is
used to indicate the type of the EAP message attached within the
message. Message Type allocation is managed by IANA [IANAWEB].

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

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

5.2. Nonce Option
Nonce: A random 32 bits number which is transported within a PCC-Initiate message and the correspondent reply message from the PCS.

5.3. Authentication Tag Option

Option-Length: The length of the Authentication Tag Option (in octet), including the 8 octet fixed header and the variable length of the authentication data.

Session ID: A 32-bit field used to indicate the identifier of the session that the message belongs to and identifies the secret key used to create the message digest appended to the PCP message.

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

Authentication Data: A variable length field that carries the Message Authentication Code for the PCP packet. The generation of the digest can be various according to the algorithms specified in...
different PCP SAs.

5.4. EAP Payload Option

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

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

5.5. PRF Option

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

PRF: The pseudo-random Function which the sender supports to generate a MSK.

5.6. Hash Algorithm Option

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

MAC Algorithm: The MAC algorithm which the sender supports to generate authentication data.
5.7. Session Lifetime Option

<table>
<thead>
<tr>
<th>Option Code</th>
<th>Reserved</th>
<th>Option-Length</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Session Lifetime: The life period of the PA Session, which is decided by the authorization result.

6. Processing Rules

6.1. Authentication Data Generation

If a PCP SA is generated as the result of a successful EAP authentication process, every subsequent PCP message within the session needs carry an Authentication Tag Option which contains the digest of the PCP message for data origin authentication and integrity protection.

Before generating a digest for a PCP message, a device needs to first select a traffic key in the session and append the Authentication Tag Option at the end of the protected PCP message. The length of the Authentication Data field is decided by the MAC algorithm adopted in the session. The device then fills the Session ID field and the PCP SA ID field, and sets the Authentication Data field as 0. After this, the device generates a digest for the PCP message with the MAC algorithm and the selected traffic key, and input the generated digest into the Authentication Data field.

6.2. Authentication Data Validation

When a device receives a PCP packet with an Authentication Tag Option, it needs to use the session ID transported in the option to locate the proper SA, and then find out the associated transport key and the MAC algorithm. After storing the value of the Authentication field of the Authentication Tag Option, the device fills the the Authentication field with zeros. Then, the device generates a digest for the packet with the transport key and the MAC algorithm found in the first step. If the value of the newly generated digest is identical to the stored one, the device can ensure that the packet has not been tampered during the transportation. The validation succeeds. Otherwise, the packet MUST be discarded.
6.3. Sequence Number

PCP adopts UDP to transport signaling messages. As an un-reliable transporting protocol, UDP does not guarantee the 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 packet exchanged during EAP authentication must carry an monotonically increased sequence number. During a PA session, a PCP device needs to maintain two sequence numbers, one for incoming packets and one for outgoing packets. When generating an outgoing PCP packet, the device attaches the outgoing sequence number to the packet and increments the sequence number by 1. When receiving a PCP packet from its session partner, the device will not accept it if the sequence number carried in the packet does not match the incoming sequence number the device maintains.

After confirming that the received packet is valid, the device increments the incoming sequence number by 1. However, the above rules are not applied to PA-Acknowledgement messages. When receiving or sending out a PA-Acknowledgement message, the device does not increase the correspondent sequence number. Another exception is message retransmission. When a device does not receive any response message from its session partner in a certain period, it needs to retransmit the last sent message with a limited rate. The duplicate messages and the original message MUST use the identical sequence number. When the device receives such duplicate messages from its session partner, it MUST tries to answer them by sending the last outgoing message with a limited rate unless it has received another valid message with a larger sequence number from its session. Note that in these cases the incoming and outgoing sequence number will not be affected by the message retransmission.

6.4. Retransmission Policies

This work provides a retransmission mechanism for reliable PA message delivery. The timer, the variables, and the rules used in this mechanism is mostly brought from PANA[RFC5191].

The retransmission behavior is controlled and described by the following variables:

- RT: Retransmission timeout from the previous (re)transmission
- IRT: Base value for RT for the initial retransmission
- MRC: Maximum retransmission count
MRT: Maximum retransmitting time interval
RAND: Randomization factor

With each message transmission or retransmission, the sender sets RT according to the rules given below.

If RT expires before receiving any reply, the sender re-calculates RT and retransmits the message. Each of the computations of a new RT include a 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 the synchronization of messages. 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. RT for the first message retransmission is based on IRT:

\[ RT = IRT \]

RT for each subsequent message retransmission is based on the previous value of RT (RTprev):

\[ RT = (2+RAND) \times RTprev \]

MRT specifies an upper bound on the value of RT (disregarding the randomization added by the use of RAND). If MRT has a value of 0, there is no upper limit on the value of RT. Otherwise:

\[ \text{if} \ (RT > MRT) \]
\[ RT = (1+RAND) \times MRT \]

MRC specifies an upper bound on the number of times a sender may retransmit a message. Unless MRC is zero, the message exchange fails once the sender has transmitted the message MRC times. In this case, the sender needs to start a session termination process illustrated in Section 3.2.

6.5. MTU Considerations

The fragmentation and reassembly of EAP messages must be provided in order to ensure the length of a PA message is not larger than the MTU of the link that it will be transported through. Therefore, a PA message may only transport a fragment of an EAP message. Because any loss or tamper of a EAP fragment will be detected and sequencing information is provided, fragmentation support can be added in a simple manner. Particularly, the S bit is set in a PA message which contain the first fragment of a EAP message, and the The E bit is set
in a PA message which contain the last fragment of a EAP message.

7. IANA Considerations

TBD

8. Security Considerations

In this work, a successful EAP authentication process performed between two PCP devices will result in the generation of a MSK which can be used to derive the transport keys to generate MAC digests for subsequent PCP message exchanges. This work does not exclude the possibility of using the MSK to generate keys for different security protocols to enable per-packet cryptographic protection. The methods of deriving the transport key for the security protocols is out of scope of this document.

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-Request and PA-Answer exchange is vulnerable to spoofing attacks as these messages are not authenticated and integrity protected. In order to prevent very basic DOS attacks, a PCP device SHOULD generate state information as little as possible in the initial PA-Request and PA-Answer exchanges. The choice of EAP method is also very important. The selected EAP method must be resilient to the attacks possibly occurred in an insecure network environment, and the user-identity confidentiality, protection against dictionary attacks, and session-key establishment must be supported.

9. Acknowledgements

This document was written using the xml2rfc tool described in RFC 2629 [RFC2629].

Some of the ideas in this document were adopted from PANA[RFC5191].

10. Change Log
10.1. Changes from -00 to -01
   o Editorial changes, added use cases to introduction.

10.2. Changes from -01 to -02
   o Add a nonce into the first two exchanged PA message between the PCC and PCS. When a PCC 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 traffic keys.
   o Specify that when a PCP device receives a PA-Request or a PA-Answer message from its partner the PCP device needs to reply with a PA-Acknowledge message to indicate that the message has been received.
   o Add the support of fragmenting EAP messages.

11. References

11.1. Normative References


11.2. Informative References


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

Phone:
Fax:
Email: zhangdacheng@huawei.com
URI: