

PCP WG
Internet-Draft
Intended status: Informational
Expires: November 5, 2012

M. Boucadair, Ed.
France Telecom
T. Zheng
P. NG Tung
X. Deng
J. Queiroz
Orange Labs
May 4, 2012

Behavior of BitTorrent service in PCP-enabled networks with Address
Sharing
draft-boucadair-pcp-bittorrent-00.txt

Abstract

This document describes the behavior of BitTorrent service in the context of PCP-enabled address sharing functions. It provides an overview of the used testbed and main results of the tests that have been conducted in order to assess the limitations of an architecture based on shared IP addresses.

Status of this Memo

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

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

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

This Internet-Draft will expire on November 5, 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. BitTorrent Overview	3
2.1. BitTorrent at a Glance	3
2.2. Software Configuration	4
2.2.1. BitTorrent Client	4
2.2.2. BitTorrent Server	4
2.2.3. BitTorrent Tracker	4
3. Testbed Overview	4
3.1. Testbed Description	4
3.2. Files	5
3.3. Methodology	5
4. Description of Tests	6
4.1. Connection to Overlay Test Group	6
4.2. Upload Test Group	7
4.3. Mutual Download Test Group	7
4.4. Simultaneous Download Test Group	8
5. Results	11
5.1. Allow Same IP Address & PCP Disabled	11
5.2. Forbid Same IP Address & PCP Disabled	13
5.3. Allow Same IP Address & PCP Enabled	15
5.4. Forbid Same IP Address & PCP Enabled	17
6. Conclusions	19
7. IANA Considerations	20
8. Security Considerations	20
9. References	20
9.1. Normative References	20
9.2. Informative References	21
Authors' Addresses	21

1. Introduction

Recently, several proposals have been disseminated within IETF to allow for IPv4 service continuity. These solutions share the same IP address among several subscribers (e.g., CGN (Carrier Grade NAT) [I-D.ietf-behave-lsn-requirements] or A+P [RFC6346])

Several issues are encountered in address sharing context as elaborated in [RFC6269].

This memo focuses on BitTorrent as an example of application which applies a restriction based on IP address. This memo describes a testing campaign that has been carried out to assess the impact of IP shared address on BitTorrent.

A particular focus has been put on the impact of activating port forwarding (using PCP [I-D.ietf-pcp-base]) on the download speed.

2. BitTorrent Overview

2.1. BitTorrent at a Glance

BitTorrent is a distributed file sharing infrastructure. It is based on P2P (Peer to Peer) techniques for exchanging files between connected users. Three parties are involved in a BitTorrent architecture as detailed hereafter:

1. The Server: The server into which, has been uploaded the torrent file.
2. The Tracker: Maintains a list of clients which have the file or some portions of that file.
3. The Client: Entities which are downloading and/or uploading portions of the file. Two categories of clients may be distinguished:
 - A. Leechers: Clients which are currently downloading the file but do not yet detain all the portions of the file. As for the portions already obtained, the leechers upload them towards requesting clients;
 - B. Seeders: Clients which detain all the portions of the file and are uploading them to other requesting clients.

A torrent file is a file which includes the meta-data information of the file to be shared: the file name, its length, a hash and the URL

of the tracker. In order to download a given file, a BitTorrent client needs to obtain the corresponding torrent file. Afterwards, it connects to the tracker to retrieve a list of leechers and seeders. Then, the client connects to those machines and downloads the available portions of the requested file. It uploads also the portions already obtained towards requesting clients.

2.2. Software Configuration

This section provides an overview of installed tools.

2.2.1. BitTorrent Client

Various BitTorrent clients are available for public use. The following one has been installed for the purposes of our testing activities:

URL: www.bittorrent.com

2.2.2. BitTorrent Server

The BitTorrent server that has been used is the following:

URL: www.torrentbox.com

2.2.3. BitTorrent Tracker

The BitTorrent tracker that has been used is the following:

URL: tracker.torrentbox.com:2710/announce

3. Testbed Overview

3.1. Testbed Description

The testbed used to conduct the testing activities is illustrated in the figure below:

- o The CGN DS-Lite which is responsible to share the same IP address among several subscribers. The CGN embeds a PCP Server.
- o CPE-1 and CPE-2 are two CPEs sharing the same IP address (by the CGN). Each CPE embeds a IGD/PCP IWF [I-D.ietf-pcp-upnp-igd-interworking].
- o T1 (respectively T2) is a machine located in the LAN behind CPE-1 (respectively CPE-2). No NAT is enabled in CPE-1 and CPE-2.

- o RT1 and RT2 are remote machines reachable through Internet. RT1 and RT2 are assigned with public IP addresses.

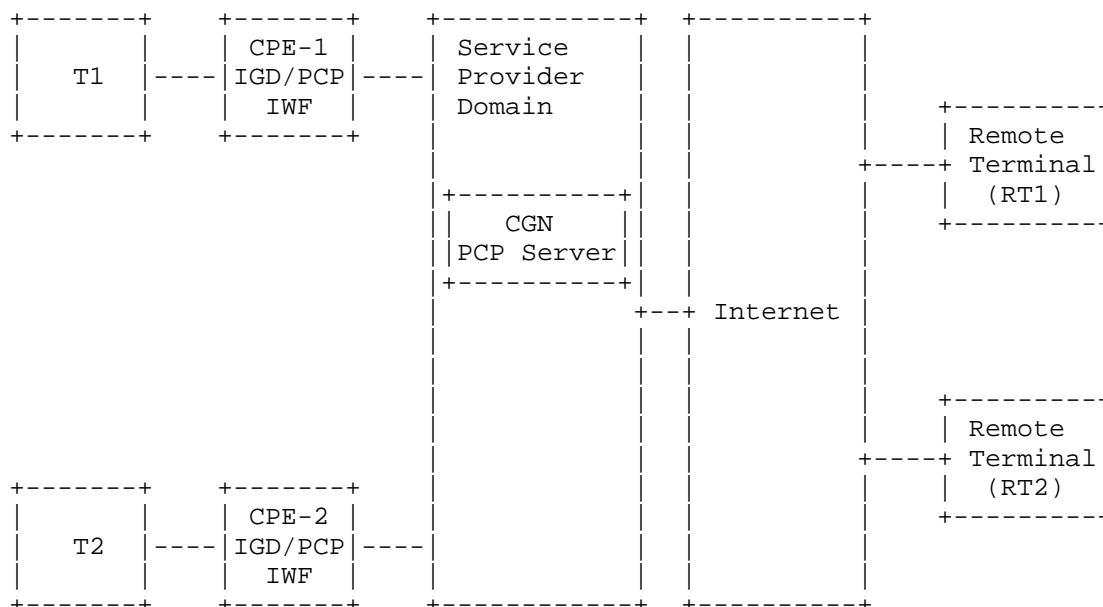


Figure 1: Testbed Overview

3.2. Files

The following table lists the files available in each machine:

Machine' s name	Available files
T1	TestCaenF1 and TestCaenFa
T2	TestCaenF1 and TestCaenFb
RT1	TestCaenFRT1 and TestCaenFRTa
RT2	TestCaenFRT1 and TestCaenFRTb

Table 1: Available files

3.3. Methodology

BitTorrent client can be configured to accept multiple connections using the same IP address. A dedicated parameter can therefore be positioned. This parameter is called: `bt.allow_same_ip`. Possible values that can be taken by this parameter are: `FALSE (0)` or `TRUE`

(1).

Tests are conducted using four configurations:

Configuration	bt.allow_same_ip	PCP
Section 5.1	TRUE in all machines (T1, T2, RT1, RT2)	Disabled
Section 5.2	FALSE in all machines (T1, T2, RT1, RT2)	Disabled
Section 5.3	TRUE in all machines (T1, T2, RT1, RT2)	Enabled
Section 5.4	TRUE in all machines (T1, T2, RT1, RT2)	Enabled

When PCP is disabled, all port forwarding entries are flushed out.

4. Description of Tests

This section lists the tests that have been conducted.

4.1. Connection to Overlay Test Group

This table lists the test to assess the ability of distinct machines having the same IP address to connect to BitTorrent overlay.

Test Index	Test Title	Purpose & Description
Test_1	Connection to BitTorrent Overlay	Check if two terminals, having the same public IP address, are able to connect to BitTorrent overlay network. Check if BitTorrent client installed on T1 and T2 machines are able to use the same tracker and that no problems are experienced to use the same tracker by T1 and T2.

Connecting to Overlay Test Group

4.2. Upload Test Group

This test group aims at checking if upload operations are not impacted/restricted due to the presence of several machines with the same IP address.

Test Index	Test Title	Purpose & Description
Test_2	Uploading distinct files using the same BitTorrent tracker and server	Check if two terminals, having the same public IP address, are able to upload torrent files (referring to distinct files) using the same tracker and same server. Check if torrent files may be uploaded from T1 and T2 using the same tracker. On T1 (resp. T2), generate a torrent file TestCaenFa.torrent (resp. TestCaenFb.torrent) referring to the file TestCaenFa (resp. TestCaenFb) and pointing to the tracker TRA. From T1 (resp. T2) try to put TestCaenFa.torrent (resp. TestCaenFb.torrent) onto server S. Check if the upload operation has succeeded
Test_3	Uploading torrent files referring to the same file	Check if two terminals, having the same public IP address, are able to upload torrent files, which refer to the same file, using the same tracker. On T1 (resp. T2), generate a torrent file TestCaenF1.torrent (resp. TestCaenF1.torrent) referring to the file TestCaenF1 and pointing to the tracker TRA. From T1 (resp. T2) try to put TestCaenF1.torrent (resp. TestCaenF1.torrent) onto server S. Check if the upload operation has succeeded

Upload Test Group

4.3. Mutual Download Test Group

The purpose of this test group is to check if mutual downloading operations can occur between machines having the same IP address.

Test Index	Test Title	Purpose & Description
Test_4	Mutual Downloading between machines sharing the same IP address	Check if two terminals having the same public IP address can download a file from each another. Check if T1 can download the file uploaded by T2 (ref. Test_2) and vice versa. Three scenarios are to be tested: (1) T1 downloads TestCaenFb but T2 does not download any file from T1, (2) T2 downloads TestCaenFa but T1 does not download any file from T2, (3) T1 downloads TestCaenFb and T2 downloads TestCaenFa at the same time
Test_5	Mutual Downloading between machines located behind an address sharing function	Check if two terminals located behind an address sharing function but assigned with distinct public IP addresses can download a file from each another. Check if T1 can download the file uploaded by T2 (ref. Test_2) and vice versa.

Mutual Download Test Group

4.4. Simultaneous Download Test Group

This test group aims at checking if simultaneous downloading operations from remote seed(s)/leecher(s) can be performed by several machines sharing the same IP address.

Test Index	Test Title	Purpose & Description
Test_6	Downloading distinct files	Check if two terminals, having the same public IP address, are able to download distinct files available on BitTorrent infrastructure. Check if distinct files available on BitTorrent infrastructure may be downloaded by T1 and T2 simultaneously
Test_7	Downloading the same file located on several seeders	Check if two terminals, having the same public IP address, are able to download the same file located on several seeders. Check if a file available on several seeders may be downloaded from T1 and T2 simultaneously. As an example, check if T1 and T2 can download the same file located in RT1 and RT2 (referred to as TestCaenFRT1)
Test_8	Download the same file available on a single machine	Check if two terminals having the same public IP address are able to download, at the same time, the same file available on a single seed. Check if T1 and T2 can download the same file uploaded by RT1 (referred to as TestCaenFRTa) concurrently. In case the test fails, one of the two host is called the "waiting client"

Test_9	Simultaneous downloading from the same seeder	Check if it is not precluded that a different file can be downloaded by the waiting client from the same seeder. In case Test_7 fails, check that it is not precluded that a different file can be downloaded by the waiting client (T1 or T2) from the same seeder (RT1) at the same time the other terminal (respectively T2 or T1) is downloading TestCaenFRTa. Execute Test_7 in launching on T1 the downloading of TestCaenFRT1 and just few seconds afterwards in launching on T2 the downloading of TestCaenFRT1 and TestCaenFRTa. Check that while T1 is downloading TestCaenFRT1 that does not preclude T2 to concurrently download TestCaenFRTa.
Test_10	Downloading distinct files from the same seeder	Check if the two terminals having the same public IP address are able to download at the same time two distinct files from the same seeder. Check if T1 (respectively T2) can download files uploaded by RT1 (referred to as TestCaenRF1 and TestCaenFRTa) concurrently. Particularly, check if T1 can download TestCaenFRT1 and T2 can download TestCaenFRTa simultaneously
Test_11	Download the same file located on machines having the same IP address	Check if the same file can be downloaded by a given machine from seeders having the same IP address. In RT1, launch the downloading of TestCaenF1. Check that RT1 is downloading portions of TestCaenF1 at the same time from T1 and T2
Test_12	Automatic query to download the same file available on a single machine	Check if the terminal which was waiting can finally download the file once the other terminal has finished. In case Test_7 fails, check that the terminal which was waiting can finally download the file once the other terminal has finished

Test_13	Download distinct files from two machines having the same IP address	Check if distinct files can be downloaded by the same machine from seeders having the same IP address. Check if RT1 can download simultaneously TestCaenFa (from T1) and TestCaenFb (from T2)
---------	--	---

Simultaneous Download Test Group

5. Results

The following tables summarize the results of the tests listed in Section 4 as performed using the testbed described in Section 3. Four configurations have been tested as documented in Section 3.3.

5.1. Allow Same IP Address & PCP Disabled

The following table summarizes the results of the tests when "bt.allow_same_ip == TRUE" in all involved BitTorrent clients and PCP is disabled.

Index	Results	Downloading Speed
Test_1	No problems have been experienced	
Test_2	Both T1 and T2 are able to upload distinct torrent files using the same tracker and the same server.	
Test_3	Only one machine can upload a torrent file referring to the same file. The server ensures that only one single torrent file corresponding to the same file is listed in its base.	

Test_4	Three scenarios have been tested: (1) T1 downloads TestCaenFb but T2 does not download any file from T1 (2) T2 downloads TestCaenFa but T1 does not download any file from T2 (3) T1 downloads TestCaenFb and T2 downloads TestCaenFa in the same time. For all these scenarios, mutual downloading between T1 and T2 is not observed.	
Test_5	No mutual downloading between T1 and T2 was observed.	
Test_6	Both T1 and T2 are able to download distinct files from the BitTorrent infrastructure.	T1: 50-110KBps; T2: 60-80KBps
Test_7	Both T1 and T2 are able to download the same file located in several seeders. Mutual downloading between T1 and T2 is not observed.	T1 and T2: 50-70KBps
Test_8	Both T1 and T2 are able to download TestCaenFRTa from RT1 simultaneously. Mutual downloading between T1 and T2 is not observed.	T1: 50-70KBps; T2: 40-80KBps
Test_9	Not applicable	
Test_10	No problem has been encountered. Distinct files located in RT1 have been successfully downloaded by T1 (respectively T2).	T1: 30-90KBps; T2: 50-80KBps
Test_11	No problem has been encountered. RT1 is able to download TestCaenF1 from T1 and T2 simultaneously.	RT1: 60-100KBps
Test_12	Not applicable	
Test_13	No problem has been encountered. RT1 has succeeded to download simultaneously TestCaenFa (from T1) and TestCaenFb (from T2).	RT1: 30-50KBps from T1 and 30-40KBps from T2

Table 2: Allow Same IP & PCP Disabled

5.2. Forbid Same IP Address & PCP Disabled

The following table summarizes the results of the tests when "bt.allow_same_ip == FALSE" in all involved BitTorrent clients and PCP is disabled.

Index	Results	Downloading Speed
Test_1	No problems have been experienced	
Test_2	Both T1 and T2 are able to upload distinct torrent files using the same tracker and the same server.	
Test_3	Only one machine can upload a torrent file referring to the same file. The server ensures that only one single torrent file corresponding to the same file is listed in its base.	
Test_4	Three scenarios have been tested: (1) T1 downloads TestCaenFb but T2 does not download any file from T1 (2) T2 downloads TestCaenFa but T1 does not download any file from T2 (3) T1 downloads TestCaenFb and T2 downloads TestCaenFa in the same time. For all these scenarios, mutual downloading between T1 and T2 is not observed.	
Test_5	No mutual downloading between T1 and T2 was observed.	
Test_6	Both T1 and T2 are able to download distinct files from the BitTorrent infrastructure.	T1: 50-110KBps T2: 60-80KBps

Test_7	Both T1 and T2 are able to download the same file located in several seeders. But for each file it is sending (here TestCaenFRT1) RT1 can allow no more than one unique connection to the same address IP. This is the same behavior for RT2. Mutual downloading between T1 and T2 is not observed.	T1: :100-120KBps, After T1 finished, T2 started 100-120KBps
Test_8	Both T1 and T2 are able to download the file but only one single connection is accepted by RT1 at the same time. This is because for each file it is sending (here TestCaenFRTa) RT1 can allow no more than one unique connection to the same address IP. The result is that, once T1 (or T2) has begun to download TestCaenFRTa, the other terminal (T2 or respectively T1) cannot get any portion of TestCaenFRTa directly from RT1 till the other (T1 or respectively T2) has completed the downloading of TestCaenFRTa. Mutual downloading between T1 and T2 is not observed.	T1: 70-100KBps
Test_9	The test has succeeded. While T1 has been downloading TestCaenFRT1 from RT1, T2 could download TestCaenFRTa from RT1 and in addition it can get portions of TestCaenFRTa already downloaded by T1.	T1: 50-70KBps T2: 40-50KBps
Test_10	No problem has been encountered. Distinct files located in RT1 have been successfully downloaded by T1 (respectively T2).	T1: 50-70KBps T2: 40-60KBps
Test_11	Both T1 and T2 are able to upload the file, but only one connection is accepted by RT1 at the same time. The test failed because, once RT1 has begun to download portions of TestCaenF1 from T1 (respectively T2) it cannot accept additional connection with T2 for the same file.	RT1: 20-40KBps from T1

Test_12	The test succeeded. Once T1 has completed its downloading from RT1, T2 has been able automatically to connect to RT1 for receiving the same file.	T2: 80-100KBps
Test_13	No problem has been encountered. RT1 has succeeded to download simultaneously TestCaenFa (from T1) and TestCaenFb (from T2).	RT1: 30-50KBps from T1 and 30-50KBps from T2

Table 3: Forbid Same IP & PCP Disabled

5.3. Allow Same IP Address & PCP Enabled

The following table summarizes the results of the tests when "bt.allow_same_ip == TRUE" in all involved BitTorrent clients and PCP is enabled.

Index	Results	Downloading Speed
Test_1	No problems have been experienced	
Test_2	Both T1 and T2 are able to upload distinct torrent files using the same tracker and the same server.	
Test_3	Only one machine can upload a torrent file referring to the same file. The server ensures that only one single torrent file corresponding to the same file is listed in its base	
Test_4	Three scenarios have been tested: (1) T1 downloads TestCaenFb but T2 does not download any file from T1 (2) T2 downloads TestCaenFa but T1 does not download any file from T2 (3) T1 downloads TestCaenFb and T2 downloads TestCaenFa in the same time. For all these scenarios, no problems have been encountered. The downloading operations have succeeded.	(1)T1: 1.4-1.5MBps (2)T2: 1.4-1.5MBps (3)T1 and T2: 600-800KBps

Test_5	The mutual downloading operations have succeeded	T1/T2: 1.4-1.5MBps
Test_6	Both T1 and T2 are able to download distinct files from the BitTorrent infrastructure.	T1: 100-110KBps T2: 60-80KBps
Test_7	Both T1 and T2 are able to download the same file located in several seeders. Mutual downloading by T1 of portions of TestCaenFRT1 already downloaded by T2 (and vice versa) has been observed.	T1 and T2: normal speed is 90-140KBps (the highest is 800KBps), between T1 and T2, the normal speed is 50-70KBps (the highest is 700KBps)
Test_8	Both T1 and T2 are able to download TestCaenFRTa from RT1 simultaneously. Mutual downloading by T1 of portions of TestCaenFRTa already downloaded by T2 (and vice versa) has been observed.	T1 and T2: normal speed is 80-110KBps (the highest is 700KBps), between T1 and T2, the normal speed is 40-50KBps (the highest is 600KBps)
Test_9	Not applicable	
Test_10	No problem has been encountered. Distinct files located in RT1 have been successfully downloaded by T1 (respectively T2).	T1: 50-70KBps T2: 40-70KBps
Test_11	No problem has been encountered. RT1 is able to download TestCaenF1 from T1 and T2 simultaneously.	RT1: 60-80KBps
Test_12	Not applicable	

Test_13	No problem has been encountered. RT1 has succeeded to download simultaneously TestCaenFa (from T1) and TestCaenFb (from T2).	RT1: 30-50KBps from T1 and 30-40KBps from T2
---------	--	--

Table 4: Allow Same IP & PCP Enabled

5.4. Forbid Same IP Address & PCP Enabled

The following table summarizes the results of the tests when "bt.allow_same_ip == FALSE" in all involved BitTorrent clients and PCP is enabled.

Index	Results	Downloading Speed
Test_1	No problems have been experienced	
Test_2	Both T1 and T2 are able to upload distinct torrent files using the same tracker and the same server.	
Test_3	Only one machine can upload a torrent file referring to the same file. The server ensures that only one single torrent file corresponding to the same file is listed in its base.	
Test_4	Three scenarios have been tested: (1) T1 downloads TestCaenFb but T2 does not download any file from T1 (2) T2 downloads TestCaenFa but T1 does not download any file from T2 (3) T1 downloads TestCaenFb and T2 downloads TestCaenFa in the same time. For (1) and (2), after several tries, downloading operations have succeeded to be observed. But for (3), mutual downloading between T1 and T2 is not observed.	(1)T1: 1.4-1.5MBps (2)T2: 1.4-1.5MBps
Test_5	The mutual downloading operations have succeeded.	T1/T2: 1.4-1.5MBps

Test_6	Both T1 and T2 are able to download distinct files from the BitTorrent infrastructure.	T1: 100-110KBps T2: 60-70KBps
Test_7	Both T1 and T2 are able to download the same file located in several seeders. But for each file it is sending (here TestCaenFRT1) RT1 can allow no more than one unique connection to the same address IP. This is the same behavior for RT2. Mutual downloading between T1 and T2 is not observed.	T1: 100-120KBps After T1 finished, T2 started 100-120KBps
Test_8	Both T1 and T2 are able to download the file but only one single connection is accepted by RT1 at the same time. This is because for each file it is sending (here TestCaenFRTa) RT1 can allow no more than one unique connection to the same address IP. The result is that, once T1 (or T2) has begun to download TestCaenFRTa, the other terminal (T2 or respectively T1) cannot get any portion of TestCaenFRTa directly from RT1 till the other (T1 or respectively T2) has completed the downloading of TestCaenFRTa. Mutual downloading between T1 and T2 is not observed.	T1: 60-90KBps
Test_9	The test has succeeded. While T1 has been downloading TestCaenFRT1 from RT1, T2 could download TestCaenFRTa from RT1 and in addition it can get portions of TestCaenFRTa already downloaded by T1.	T1: 50-70KBps T2: 40-50KBp
Test_10	No problem has been encountered. Distinct files located in RT1 have been successfully downloaded by T1 (respectively T2).	T1: 50-70KBps T2: 30-50KBps

Test_11	Both T1 and T2 are able to upload the file, but only one connection is accepted by RT1 at the same time. The test failed because, once RT1 has begun to download portions of TestCaenF1 from T1 (respectively T2) it cannot accept additional connection with T2 for the same file.	RT1: 20-40KBps from T1
Test_12	The test succeeded. Once T1 has completed its downloading from RT1, T2 has been able automatically to connect to RT1 for receiving the same file.	T2: 80-100KBps
Test_13	No problem has been encountered. RT1 has succeeded to download simultaneously TestCaenFa (from T1) and TestCaenFb (from T2).	RT1: 30-40KBps from T1 and 40-50KBps from T2

Table 5: Forbid Same IP & PCP Enabled

6. Conclusions

This document describes the main behavior of BitTorrent in an IP shared address environment. The impact of activating port forwarding (here PCP is used) has been also assessed.

Mutual file sharing between hosts sharing the same IP address has been checked. Machines having the same IP address can share files with no alteration compared to current IP architectures only if port forwarding (PCP in our case) is enabled.

Mutual file sharing between hosts behind an IP address sharing function has been also checked. Machines having distinct IP addresses but located behind an address sharing function can share files with no alteration compared to current IP architectures only if port forwarding (PCP in our case) is enabled.

Even if PCP is enabled, two limitations were experienced:

The first limitation occurs when two clients sharing the same IP address want to simultaneously retrieve the SAME file located in a SINGLE remote peer. This limitation is due to the default BitTorrent configuration on the remote peer which does not permit

sending the same file to multiple ports of the same IP address. This limitation is mitigated by the fact that clients sharing the same IP address can exchange portions with each other, provided the clients can find each other through a common tracker, DHT, or Peer Exchange. Even if they can not, we observed that the remote peer would begin serving portions of the file automatically as soon as the other client (sharing the same IP address) finished downloading. This limitation is eliminated if the remote peer is configured with `bt.allow_same_ip == TRUE`.

The second limitation occurs when a client tries to download a file located on several seeders, when those seeders share the same IP address. This is because the clients are enforcing `bt.allow_same_ip` parameter to `FALSE`. The client will only be able to connect to one seeder, among those having the same IP address, to download the file (note that the client can retrieve the file from other seeders having distinct IP addresses). This limitation is eliminated if the local client is configured with `bt.allow_same_ip == TRUE`, which is somewhat likely as those clients will directly experience better throughput by changing their own configuration.

7. IANA Considerations

This document raises no IANA considerations.

8. Security Considerations

This memo does not introduce any security issue.

9. References

9.1. Normative References

[I-D.ietf-pcp-base]

Wing, D., Cheshire, S., Boucadair, M., Penno, R., and P. Selkirk, "Port Control Protocol (PCP)",
draft-ietf-pcp-base-24 (work in progress), March 2012.

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

Boucadair, M., Dupont, F., Penno, R., and D. Wing,
"Universal Plug and Play (UPnP) Internet Gateway Device
(IGD)-Port Control Protocol (PCP) Interworking Function",
draft-ietf-pcp-upnp-igd-interworking-01 (work in
progress), March 2012.

9.2. 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-06 (work in
progress), May 2012.
- [RFC6269] Ford, M., Boucadair, M., Durand, A., Levis, P., and P.
Roberts, "Issues with IP Address Sharing", RFC 6269,
June 2011.
- [RFC6346] Bush, R., "The Address plus Port (A+P) Approach to the
IPv4 Address Shortage", RFC 6346, August 2011.

Authors' Addresses

Mohamed Boucadair (editor)
France Telecom
Rennes 35000
France

Email: mohamed.boucadair@orange.com

Tao Zheng
Orange Labs
Beijing
China

Email: tao.zheng@orange.com

NG Tung
Orange Labs
Issy Les Moulineaux
France

Email: paul.ngtung@orange.com

Xiaohing Deng
Orange Labs
Beijing
China

Email: xiaohong.deng@orange.com

Jaqueline Queiroz
Orange Labs
Issy Les Moulineaux
France

Email: jaqueline.queiroz@orange.com

Network Working Group
Internet-Draft
Intended status: Informational
Expires: January 17, 2013

G. Chen
Z. Cao
China Mobile
M. Boucadair
France Telecom
A. Vizdal
Deutsche Telekom AG
L. Thiebaut
Alcatel-Lucent
July 16, 2012

Analysis of Port Control Protocol in Mobile Network
draft-chen-pcp-mobile-deployment-01

Abstract

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

Status of this Memo

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

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

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

This Internet-Draft will expire on January 17, 2013.

Copyright Notice

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

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

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

Table of Contents

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

1. Introduction

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

It should be noticed that mobile network have their particular characteristics. There are several factors that should be investigated before implementing PCP in a mobile context. Without the particular considerations, PCP may not provide desirable outcomes. Some default behaviours may even cause negative impacts or system failures in a mobile environment. Considering very particular environments of mobile networks, it's needed to have a document describing specific concerns from mobile network side. That would also encourage PCP support in mobile network as well.

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

2. Benefits of Introducing PCP in Mobile Network

2.1. Restoring Internet Reachability

Many Mobile networks are making use of a Firewall to protect their customers from an unwanted Internet originated traffic. The firewall is usually configured to reject all unknown inbound connections and only permit inbound traffic that belongs to a connection initiated from the Firewall or NAT/PAT device. There are applications that can be running on the terminal that require to be reachable from the Internet or there could be services running behind the terminal that require reachability from the Internet. PCP enabled applications /

devices could request a port or a port range from the Firewall to ensure Internet reachability, and thus would not need to be using keep-alive to keep the Firewall session open. This would result in resource savings on the Firewall node whilst still keeping the customer protected from the unwanted traffic.

2.2. Keepalive Message Optimization

Many always-on applications, e.g. instant message and p2p applications, are usually keeping long-lived connections with their network peers. To make sure that they can receive incoming traffic from their network peers, they issue periodic keep-alive messages in order to keep the NAT/FW bindings active. As the NAT/FW binding timer may be short and unknown to the UE, the frequency of these keep-alive may be high. These keep-alive generally do not contain useful data and thus correspond to "useless" usage of the radio spectrum and of network resources, e.g.:

- o Allocation of radio resources to traffic that could be avoided or limited
- o For each of these keep-alive messages, the UE needs to be put in CONNECTED state, i.e. an operation that consumes a fair amount of signaling

PCP helps to reduce the frequency of periodic messages aimed at refreshing a NAT/FW binding by indicating to the mobile the Life time of a binding. PCP helps to avoid different periodic (keep-alive) messages from different applications by allowing the aggregation of binding refresh within one round-trip control message with the NAT/FW.

2.3. Energy Saving

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

2.4. Balance Resource Assignment

Network resources have been consumed due to heavy signaling process, like frequent beacon message, retransmission control. Such various usages are significantly increasing the resource consumption on a

control plan and decreasing the efficiency on data forwarding (user plane). For example, 16% of traffic caused by instant signalling message would consume 50%~70% radio resource in some area. Since radio access is a resource constrained environment, imbalance of resource assignment would decline Call Setup Success Rate(CSSR) and operational profits. Reduction on control plan load would shift more resources for data transmission, which could contribute the optimization of resource arrangements.

3. Overviews of PCP Deployment in Mobile Network

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

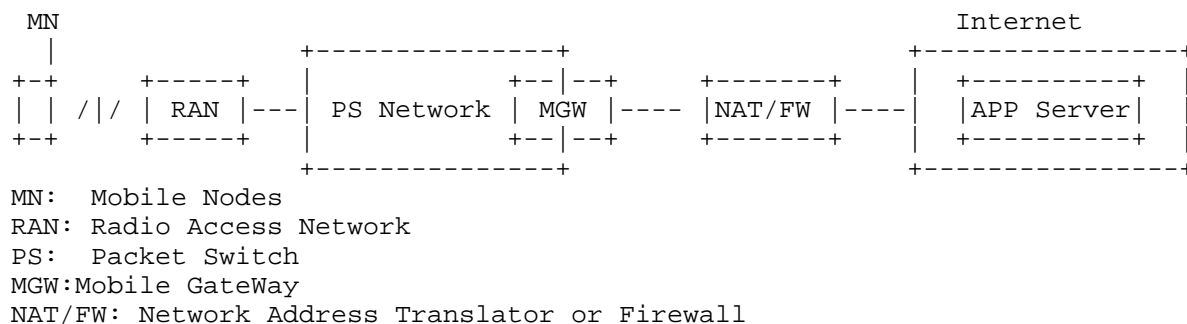


Figure 1: Mobile Networks Scenario

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

4. PCP Server Discovery

A straightforward solution seems that MN assume their default router

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

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

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

Regarding the IPv4 bearer, MN generally indicates that it prefers to obtain an IPv4 address as part of the PDP context activation procedure. In such a case, the MN relies on the network to provide IPv4 parameters as part of the PDP context activation/ PDN connection set-up procedure. The MN may nevertheless indicate that it prefers to obtain the IPv4 address and configuration parameter after the PDP Context activation by DHCPv4, but it is not available on a wide scale[RFC6459]. PCP server name options in DHCPv4 would not help the PCP servers discovery in that case. Alternative ways could be considered to support PCP server discovery by a MN:

- o Protocol Configuration Options(PCO) based[TS24.008]
- o DNS based

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

Another alternative solution is to directly perform an inverse name query in IN-ADDR.ARPA domain[RFC1035]. Normally, MN and NAT/FW would locate in same IPv4 subnet. The MN could easily determine the number of labels associating with IN-ADDR.ARPA to identify a particular zone. For example,

UE with IPv4 10.1.0.0/16 could resolve the 1.10.IN-ADDR.ARPA locating PCP servers, the domain database would contain:

1.10.IN-ADDR.ARPA. PTR PCP.server.3gppnetwork.org.

When it receives a RRs in response, like PCP.server.3gppnetwork.org. The UE could then originate QTYPE=A, QCLASS=IN queries for PCP.server.3gppnetwork.org. to discover the addresses.

5. MN and multi-homing

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

6. Retransmission Consideration

PCP designed retransmission mechanisms on the client for reliable delivery of PCP request. The client must retransmit request message until successfully receiving response or determining failure. Several timers were specified to control the retransmission behavior. Configurable timers of Maximum Retransmission Duration(MRD) gives an opportunity to optimize the behavior fitting into different environments.

A class of devices in mobile networks are usually powered with limited battery . Users would like to use such MN for several days without charging, even several weeks in sensor case. Many applications do not send or receive traffic constantly; instead, the network interface is idle most of the time. That could help to save energy unless there is data leading the link to be activated. Such state changes is based on network-specific timer values corresponding to a number of Radio Resource Control (RRC) states(see more at Section 8.2.2 3GPP[TS23.060]. The time transiting to idle is normally less than default Maximum Retransmission Time (MRT), i.e. 1024 seconds. With "no maximum" of MRD, would cause devices activating their uplink radio in order to retransmit the request messages. Furthermore, the state transition and the transmission take some time, which causes significant power consumption. The MRD should be configured with an optimal time which in line with activated state duration on the device. That could help to avoid

frequent wake-up the device and consume the battery.

The power consumption problem is made complicated if several PCP clients residing on a MN. Several clients are potentially sending requests at random times and by so doing causing MN uplink radio into a significantly power consuming state for unnecessarily often. It's necessary to perform a synchronization process for tidy up several PCP clients retransmission. A time-line observer is required to control different PCP clients resending requests in an optimal transmission window. If the uplink radio of MN is active at the time of sending retransmission from several clients, a proper MRD described as above should be set for the clients. If the uplink radio of MN is in idle mode, the time-line observer should hold Initial Retransmission Time(IRT) for while to synchronize different retransmitted PCP requests into same optimal transmission window. Such duration of optimal transmission window should equal with RRC state timer on the MN. The holding timer in idle mode may be set as 100 seconds as recommended in [I-D.savolainen-6man-optimal-transmission-window]. Several PCP clients should wait a random amount of time between 0 and 100 milliseconds to prevent synchronization of all PCP clients.

7. Unsolicited Messages Delivery

When the states on NAT/FW have been changed like reboot or changed configuration, PCP servers can send unsolicited messages (e.g. ANNOUNCE Operation)to clients informing them of the new state of their mappings. This aims at achieving rapid detection of PCP failure and rapid PCP recovery. However, it may induce difficulties in mobile environments.

Multicast delivery may not be available in 3GPP network, because it optionally supports IP Multicast routing of packets. When multicast delivery is not possible, PCP servers may use unicast delivery of ANNOUNCE noting that

- o This requires PCP servers to retain knowledge of the IP address(es) and port(s) of their clients even though they have rebooted
- o Care should be taken not to generate floods of unicast ANNOUNCE messages, e.g. to multiple thousands of MN that were served by a PCP server that has rebooted. Such flood may have a detrimental impact on Mobile Networks as it may imply the simultaneous generation of Paging process(see more at Section 8.2.4 3GPP[TS23.060]) for very big numbers of MN.

Thus a PCP server SHOULD take care to throttle unicast ANNOUNCE messages it sends towards a collection of MN.

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

8. SIPTO Architecture

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

The mainstream standard deployment relies on selecting a MGW that is / are geographically/ topologically close to a UE's point of attachment. This deployment may apply to both 3G and LTE. The MN may sometimes be requested to re-activate its PDP context / PDN connection, in which case it is allocated a new MGW and thus a new IP address and a new PCP server. In this case SIPTO has no detrimental impact on PCP as SIPTO resolves to a change of MGW and of PCP server.

As an implementation option dedicated to 3G networks, it is also possible to carry out Selected IP Traffic Offload in a TOF entity located at the interface of the Radio Access Network i.e. in the path between the Radio stations and the Mobile Gateway. The TOF decides on which traffic to offload and enforces NAT for that traffic. The point is that the deployment of a TOF is totally transparent for the UE that even cannot know which traffic is subject to TOF (NATed at the TOF) and which traffic is processed by the MGW (and the FW/NAT controlled by the PCP server whose address has been determined per mechanisms described in section 5 of this document). In case of TOF deployment, the PCP server advertised by the MGW does not take into account the NAT carried out by the TOF function.

Therefore, PCP client doesn't know which PCP servers should be selected to send the request.

[I-D.rpcw-pcp-pmipv6-serv-discovery] provides a solution in similar architecture, in which a smart PCP proxy [I-D.ietf-pcp-proxy] is required on the offloading point to dispatch requests to a right PCP server. However, TOF in 3GPP stores radio network layer information (e.g. RAB ID) to build the local offload context. That

can't directly be used to identify a IP flow with 5 tuples. Additional functionalities is required to map identifier of IP flow to RAB ID. PCP proxy may need to include such radio link information in its local context.

9. Authentication Consideration

The authentication issue in PCP is important to any operating networks, because operators do not want unauthenticated requests to control their NAT/FW ports and addresses. In mobile networks, this issue becomes especially important due to the fact that the mis-function of Carrier Grade NAT will severely destroy user experience and network operating.

The problem of PCP authentication comes from the fact that the PCP client (device) and PCP server (FW) usually do not have trust pre-established relationship with each other. To ensure client authentication, we can either use in-band or out-of-band solutions. In-band means that the authentication service is provided within the PCP exchange (e.g., by defining extended options), while out-of-band solutions handle the problem by establishing new trust relationships or reuse existing trust without extending the PCP base protocol.

As an in-band solution, [I-D.ietf-pcp-authentication] has provided solutions for PCP authentication, in which an EAP option is included in the PCP requests from the devices. In mobile network, provisioning of new credentials to mobile devices is a difficult task. Taking this into consideration, using EAP-SIM/EAP-AKA/ EAP-AKA' authentication is recommended as in-band solution for 3GPP network.

One possible out-band solution is the use of open authentication capability such as 3GPP GAA (Generic Authentication Architecture) defined in 3GPP[TS33.220]. So that, the PCP client can invoke the authentication ability provided by the operator. The other way is to reuse the trust relationship between UE and the MGW. Because the UE has been authenticated to the MGW during context setup, if the MGW delegates its trust to the NAT/FW device (PCP server), the NAT/FW device can trust the PCP requests from those users.

10. Conclusion

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

considerations. The memo would take the role to document such considerations for PCP deployment in mobile network.

11. Security Considerations

TBD

12. IANA Considerations

This document makes no request of IANA.

13. Acknowledgements

The authors would like to thank Ping Lin and Tao Sun for their discussion and comments.

14. References

14.1. Normative References

- [I-D.ietf-pcp-authentication]
Wasserman, M., Hartman, S., and D. Zhang, "Port Control Protocol (PCP) Authentication Mechanism", draft-ietf-pcp-authentication-00 (work in progress), June 2012.
- [I-D.ietf-pcp-base]
Wing, D., Cheshire, S., Boucadair, M., Penno, R., and P. Selkirk, "Port Control Protocol (PCP)", draft-ietf-pcp-base-26 (work in progress), June 2012.
- [I-D.ietf-pcp-dhcp]
Boucadair, M., Penno, R., and D. Wing, "DHCP Options for the Port Control Protocol (PCP)", draft-ietf-pcp-dhcp-03 (work in progress), May 2012.
- [I-D.ietf-pcp-proxy]
Boucadair, M., Dupont, F., Penno, R., and D. Wing, "Port Control Protocol (PCP) Proxy Function", draft-ietf-pcp-proxy-00 (work in progress), April 2012.
- [RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, November 1987.

- [RFC3736] Droms, R., "Stateless Dynamic Host Configuration Protocol (DHCP) Service for IPv6", RFC 3736, April 2004.
- [RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, September 2007.
- [TS23.060] "General Packet Radio Service (GPRS); Service description; Stage 2", June 2012.
- [TS23.401] "General Packet Radio Service (GPRS) enhancements for Evolved Universal Terrestrial Radio Access Network (E-UTRAN) access", June 2012.

14.2. Informative References

- [I-D.rpcw-pcp-pmipv6-serv-discovery] Reddy, T., Patil, P., Chandrasekaran, R., and D. Wing, "PCP Server Discovery with IPv4 traffic offload for Proxy Mobile IPv6", draft-rpcw-pcp-pmipv6-serv-discovery-00 (work in progress), February 2012.
- [I-D.savolainen-6man-optimal-transmission-window] Savolainen, T. and J. Nieminen, "Optimal Transmission Window Configuration Option for ICMPv6 Router Advertisement", draft-savolainen-6man-optimal-transmission-window-00 (work in progress), June 2012.
- [RFC3315] Droms, R., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, July 2003.
- [RFC6459] Korhonen, J., Soininen, J., Patil, B., Savolainen, T., Bajko, G., and K. Iisakkila, "IPv6 in 3rd Generation Partnership Project (3GPP) Evolved Packet System (EPS)", RFC 6459, January 2012.
- [TS24.008] "Mobile radio interface Layer 3 specification; Core network protocols; Stage 3", 9.11.0 3GPP TS 24.008, June 2012.
- [TS33.220] "Generic Authentication Architecture (GAA); Generic Bootstrapping Architecture (GBA)", 10.1.0 3GPP TS 33.220,

March 2012.

[VTC2007_Energy_Consumption]

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

Authors' Addresses

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

Email: phdgang@gmail.com

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

Email: caozhen@chinamobile.com

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

Email: mohamed.boucadair@orange.com

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

Phone:
Fax:
Email: ales.vizdal@t-mobile.cz
URI:

Laurent Thiebaut
Alcatel-Lucent

Phone:
Fax:
Email: laurent.thiebaut@alcatel-lucent.com
URI:

Internet Engineering Task Force
Internet Draft
Intended status: Informational
Expires: January 10, 2013

X.Deng
M.Boucadair
France Telecom
X.Wang
BUPT
July 9, 2012

Using PCP to update dynamic DNS
draft-deng-pcp-ddns-01.txt

Abstract

This document focuses on the problems encountered when using dynamic DNS in address sharing contexts (e.g., DS-Lite, NAT64, A+P) during IPv6 transition. Issues, possible solutions and preliminary implementation and validation of one of the solutions are documented in this memo.

Status of this Memo

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

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

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

This Internet-Draft will expire on January 10, 2013.

Copyright Notice

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

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect

to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Problem statement	2
2. Solution Space	3
2.1. Locate a service port.....	3
2.2. Detect the changes	4
3. Implementation & Validation	7
4. References	8
4.1. Normative References.....	8
4.2. Informative References.....	8
5. Authors' Addresses	9

1. Problem statement

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

When the network architecture evolves towards an IPv4 sharing architecture during IPv6 transition, the DDNS Client will have to not only inform the IP address updates if any, but also to notify the changes of external port on which the service is listening, because a well know port numbers, e.g. port 80 will no longer be available to every web server. It will also require the ability to configuring corresponding port forwarding on CGN devices, so that incoming communications initiated from outside can be routed to the appropriate server behind the CGN.

This document focuses on the problems encountered when using dynamic DNS in address sharing contexts (e.g., DS-Lite, NAT64, A+P). Below are listed the main challenges to us:

- (1) The DDNS service MUST be able to maintain an alternative port number instead of the default port number.
- (2) Appropriate means to instantiate port mapping in the address sharing device MUST be supported.
- (3) DDNS client MUST be triggered by the change of the external IP address and the port number. Concretely, upon change of the external IP address, the DDNS client MUST refresh the DNS records otherwise the server won't be reachable from outside. This issue is event exacerbated in the DS-Lite context because no IPv4 address is assigned to the CPE.

This document describes solutions to counter the issues listed above in the particular case of DS-Lite.

Note DDNS may be considered as an implementation of the Rendez-vous service mentioned in [I-D.ietf-pcp-base].

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

Dynamic Updates in the standard Domain Name System (DNS UPDATE) (RFC2136) is out of scope of this memo.

2. Solution Space

2.1. Locate a service port

At least two solutions can be used to associate a port number with a service identified:

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

- (2) Use SRV records. Unfortunately, the majority of browsers do not support this record type.

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

2.2. Detect the changes

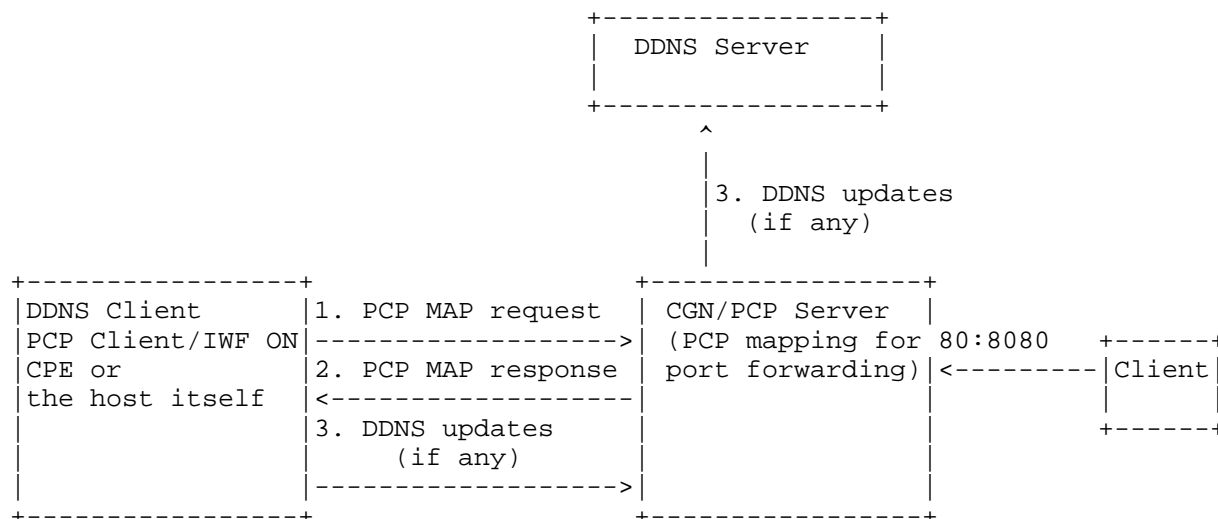


Figure 1 : Flow chat

First of all, PCP MUST be used to install the appropriate mapping in the CGN so that incoming packets can be delivered to the appropriate server.

In a network described in figure 1, DDNS Client/ PCP Client can either be running on a Customer Premise Equipment (CPE) or be running

on the host that is hosting some services, itself. There are possible ways to address the problems stated in section 1.

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

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

3. Implementation & Validation

So far the topology of network has been implemented as Figure 1. Based on the DS-Lite environment some new roles added into it such as DDNS. It could be implemented by Apache or other applications which has virtual host functions. The DDNS need to be configured as a virtual host and redirect corresponding request to the pointed IPv4 address and port number. It could be validated as Figure 2 shows.

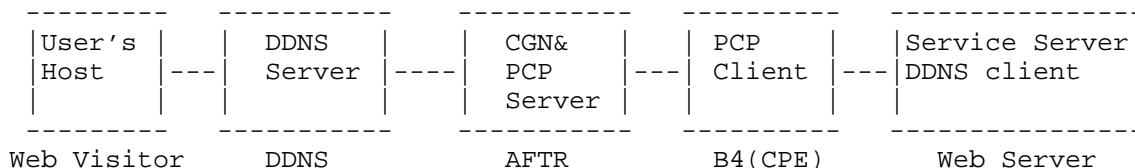


Figure 2 : Implementation Chart

Web Visitor: Some users who need to access service on the Web Server. They send service request needed to resolve domain name. And the Web response would returned to their hosts as the ways of request reached to the Web Server.

DDNS: Maintaining mappings between domain name and external IPv4 address: port. If a DNS request was sent to it, DDNS server could resolve it to the AFTR which contains that IPv4 address and port number.

AFTR: Responsible for mappings between internal IPv4 Address: port and external IPv4 address: port. It maintains a table to restore these data to keep state of every mapping.

B4 (CPE): An endpoint of IPv4-in-v6 tunnel, and PCP client also runs on it. A package from Web Server is encapsulated into a IPv4-in-v6 one and is sent to the AFTR. A package from AFTR will be decapsulated to a normal IPv4 package and to their destination.

Web Server: Web server was deployed in the DS-Lite network environment. It just has private IPv4 address and with a mapping in AFTR to the public network. Web server may offer Web, FTP, SIP service and so on. And these services may not be set as their specific port. (this also is the reason why introducing DDNS into DS-Lite environment)

If the DDNS client is enabled, the A/AAAA records of DNS (which could be normal one as using on the Internet now) were set to point the DDNS Server. DDNS is responsible for the translation between public IPv4 address (address of DDNS) with specific port (E.g. web with 80 port) and public IPv4 address (outside IPv4 address and port number of mappings). Show as Figure 3.

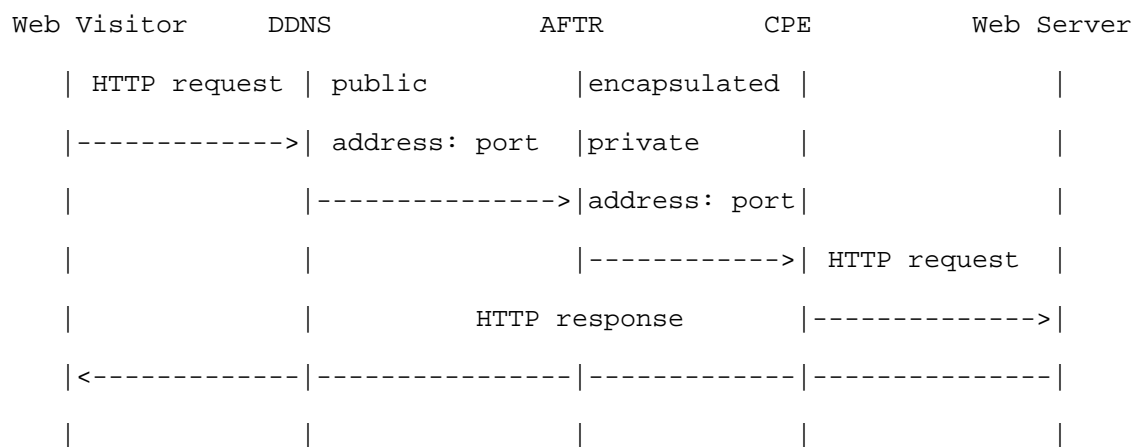


Figure 3 : Time Sequence Chart

If a user of another client outside DS-Lite network wants to access a Web Server behind AFTR, the role of DDNS started to become important. Before that the following mappings should had been configured well:

a. PCP mappings: private IPv4 address: port number <--> public IPv4 address: port number

b. DDNS mappings: public IPv4 address: port number <--> domain name

c. DNS Resolution: A/AAAA Records (point to the DDNS server) <--> domain name

A domain resolution request is sent from host of customer who asking service. The request is sent to the DNS server. And DNS server would return a DNS response with A/AAAA records pointing to the DDNS server. If the request is sent to the DDNS directly, it would redirect the request to the pointed IPv4 address and port number which has been configured in the mappings.

After redirection the request is routed to the AFTR. AFTR would translate it from public IPv4 address and port number into private IPv4 address and port. The request finished AFTR translation and is encapsulated into a IPv4-in-IPv6 package until CPE.

At last the request would be decapsulated to an IPv4 package and is sent to the service provider. And the Web response would return to

the customer as requested routine. The whole communication process is finished successfully.

From the view of Web visitor, the location of Web Server is on

DDNS, just like a virtual host. It at least has three advantages.

Firstly, hackers and other attackers couldn't reach the real host and do something bad. The security is assured. Secondly, many domain name or space ISPs also provide service of domain and port mapping. However, some companies may use iframe or 301 redirection technology. Those means could lead to lower speed and affect PR weights to the search engine. Click-through rate and visits was 'stolen'. That could not be introduced into Carrier Scale Network. Hence, generation of DDNS has its unique meaning. Thirdly, DDNS solution could solve the problems of IP address + port mapping almost perfectly. Under DS-Lite network environment normal DNS resolution couldn't point a domain name to a IP address and a port. Because of designing defect of traditional DNS protocol a DNS request just could be resolve to be a A/AAAA record (the services have their own specific port. Such as web is 80 and ftp is 21, etc.). So DDNS as a supplementary was introduced into DS-Lite to play a role of mapping between domain name and IP address and port number.

4. References

4.1. Normative References

[RFC2136]

P. Vixie, et. al. " Dynamic Updates in the Domain Name System (DNS UPDATE)", April 1997.

[RFC3007]

B. Wellington, " Secure Domain Name System (DNS) Dynamic Update", November 2000.

[RFC3986]

T. Berners-Lee, et. al. " Uniform Resource Identifier (URI): Generic Syntax", January 2005.

4.2. Informative References

[I-D.ietf-pcp-base]

D. Wing, et. al. " Port Control Protocol (PCP)", June 5, 2012.

5. Authors' Addresses

Xiaohong Deng
France Telecom
Rennes, 35000 France
Email: dxhbupt@gmail.com

Mohamed BOUCADAIR
France Telecom
Rennes, 35000 France

Email: mohamed.boucadair@orange.com

Xu Wang
Beijing University of Posts and Telecommunications, China
Email: cngesaint@gmail.com

Network Working Group
Internet-Draft
Intended status: Experimental
Expires: December 29, 2012

M. Wasserman
S. Hartman
Painless Security
D. Zhang
Huawei
June 27, 2012

Port Control Protocol (PCP) Authentication Mechanism
draft-ietf-pcp-authentication-00.txt

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 December 29, 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. Terminology	4
3. Separate vs. Inline Key Management	5
4. Separate Key Management	5
5. Inline Key Management	5
6. Protocol Details	6
6.1. Session Initiation	6
6.2. Session Termination	8
7. PA Security Association	8
8. Packet Format	9
8.1. Authentication OpCode Format	9
8.2. Nonce Option	10
8.3. Authentication Tag Option	11
8.4. EAP Payload Option	12
8.5. PRF Option	12
8.6. Hash Algorithm Option	13
8.7. Session Lifetime Option	13
9. Processing Rules	13
9.1. Authentication Data Generation	13
9.2. Authentication Data Validation	14
9.3. Sequence Number	14
9.4. Retransmission Policies	15
9.5. MTU Considerations	16
10. IANA Considerations	16
11. Security Considerations	16
12. Acknowledgements	17
13. Change Log	17
13.1. Changes from wasserman-pcp-authentication-02 to ietf-pcp-authentication-00	17
13.2. Changes from wasserman-pcp-authentication-01 to -02 . . .	17
13.3. Changes from wasserman-pcp-authentication-00 to -01 . . .	17
14. References	17
14.1. Normative References	17
14.2. Informative References	18
Authors' Addresses	18

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 their clients with Extensible Authentication Protocol (EAP). The following issues are considered in the design of this extension:

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

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

- o On security infrastructure equipment, such as corporate firewalls, that does not create implicit mappings.
- o On equipment (such as CGNs or service provider firewalls) that serve multiple administrative domains and do not have a mechanism to securely partition traffic from those domains.
- o For any implementation that wants to be more permissive in authorizing explicit mappings than it is in authorizing implicit mappings.
- o For implementations that support the THIRD_PARTY Option (unless they can meet the constraints outlined in Section 14.1.2.2).
- o 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 (e.g., the one defined in [RFC5448]) .

PA (PCP for Authentication) message: A PCP message containing an Authentication OpCode for EAP authentication.

non-PA message: A PCP message which is not a PA message.

3. Separate vs. Inline Key Management

There is an open question in the working group regarding what approach should be used for PCP key management. The precursor to this document originally proposed an inline key management approach using EAP directly over PCP. There was an alternative proposal on the list to standardize a separate key management approach using PANA [RFC5191] (with EAP). The WG will need to make a decision between these two approaches before this document can be completed.

Both approaches for key management could be used with the integrity protection mechanism and options described later in this document.

4. Separate Key Management

The separate key management proposal involves running PANA between the end-points to dynamically generate a security association, and then using that security association to authenticate PCP message exchanges.

For this approach we would define an AVP for PANA to indicate that the PANA session was being used for PCP authentication, not for network access purposes.

A PANA server would be implemented on each PCP server that support authenticated requests, or another mechanism would need to be specified to locate a PANA server that can be used for PCP-related PANA requests. It may be possible to define a subset of the PANA protocol that can be run on PCP Servers if the same PANA server will not be used for network access. For example, it would not be necessary for these servers to support IP Address Reconfiguration.

Once a secure session has been established using PANA, the Secure OpCode option described in this draft could be used to associate PCP requests with a particular PANA session. Some discussion may be needed on how the PCP session will be securely bound to the PANA session initiation.

Although a separate key management approach using PANA has been discussed on the PCP mailing list, this approach would require further documentation if the WG decides to pursue it.

5. Inline Key Management

The inline key management approach is described in this document in the sections Section 6.1 and Section 6.2.

6. Protocol Details

6.1. Session Initiation

To carry out an EAP authentication process between two PCP devices, a set of PA messages need to be exchanged. A PA message contains an Authentication OpCode and associated Options. The Authentication OpCode consists of three fields: Session ID, Flag, and Sequence Number. The Session ID field is used to identify the session to which the message belongs. The Flag field indicates the type of the PCP 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. In the message, the Session ID and Sequence Number fields of the Authentication OpCode are set as 0; the I bit is set. The PCC-Initiation message is also attached with a nonce option which consists of a random nonce selected by the PCC to tolerate off-line attacks. 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 a nonce option which is learned from the PCC. From now on, every PA message within this session must be attached with the session identifier. When receiving a PA message from an unknown session, a PCP device MUST 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 inform the PCS that it would like to perform EAP authentication and skip over the step of waiting for the EAP Identity Request.

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 un-necessarily 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. Therefore, after a successful authentication procedure, a Master Session Key (MSK) will be generated. 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 contains a PRF that the PCS supports. Similarly, each MAC Algorithm Option contains a MAC (Message Authentication Code) algorithm that the PCS supports. After receiving the request, the PCC selects a PRF and a MAC algorithm which it would like to use, 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 succeeds, 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 successes 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.

6.2. 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.

7. 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:

- o IP address and UDP port number of the PCC
- o IP address and UDP port number of the PCS
- o Session Identifier
- o Sequence number for the next outgoing PCP message
- o Sequence number for the next incoming PCP message
- o Last outgoing message payload
- o Retransmission interval
- o MSK
- o 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:

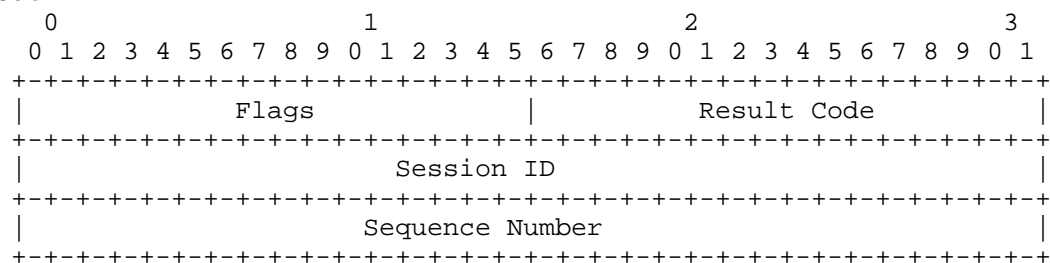
Transport key = prf(MSK, "IETF PCP" | Session_ID, key ID), where:

- 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

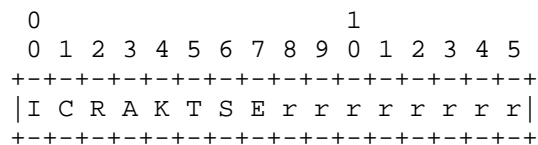
8. Packet Format

8.1. Authentication OpCode Format

The following figure illustrates the format of an authentication OpCode:



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.

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.

Result Code: This field is two octets. Following result code values are defined:

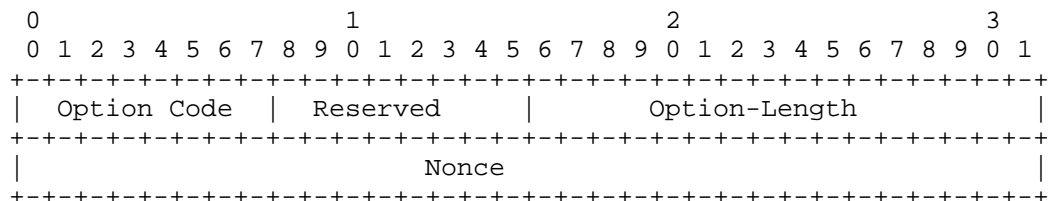
- 1 AUTHENTICATION-REQUIRED
- 2 AUTHENTICATION-FAILED
- 3 AUTHENTICATION-SUCCESS
- 4 AUTHORIZATION-FAILED

8.2. Nonce Option

Question: Would it be possible to remove this option from the PCP authentication draft, and use the nonce from the main PCP header instead?

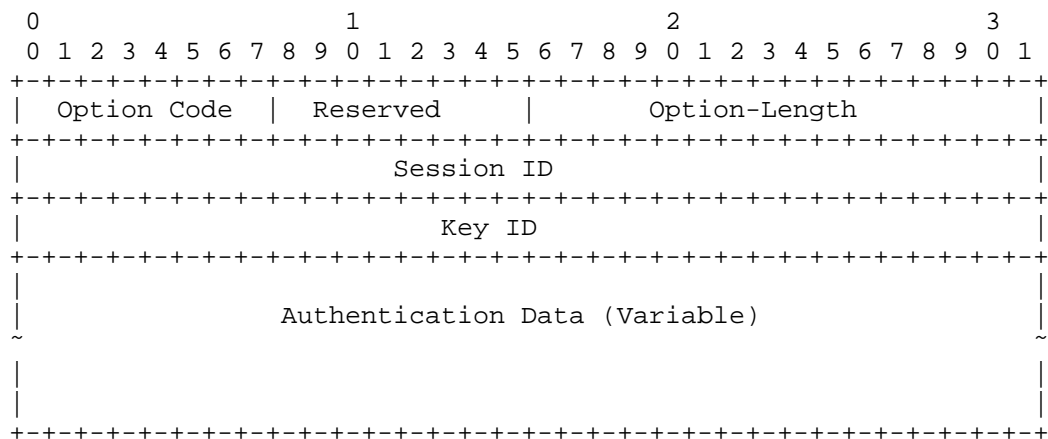
Because the session identifier of PA session is determined by the PCS, a PCS does not know the session identifier which will be used when it sends out a PCC-Initiation message. In order to prevent an attacker from interrupting the authentication process by sending off-line generated PA-Request messages, the PCS needs to generate a

random number as nonce in the PCC-Initiation message. The PCS will append the nonce within the initial PA-Request message. If the PA-Request message does not carry the correct nonce, the message will be discarded silently.



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

8.3. Authentication Tag Option



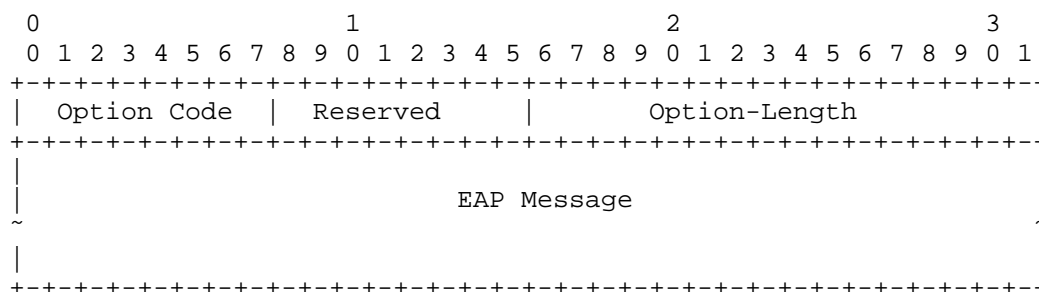
Option-Length: The length of the Authentication Tag Option (in octet), including the 8 octet fixed header and the variable length of the authentication data.

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.

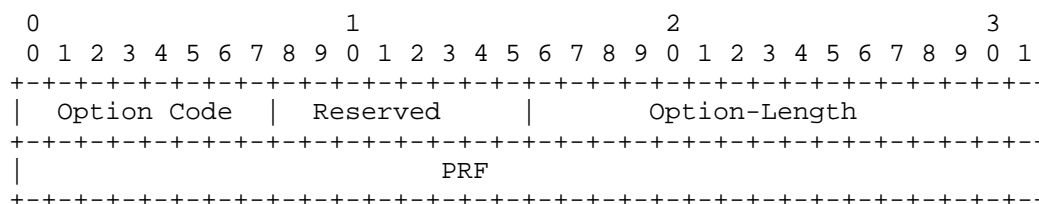
Authentication Data: A Variable length field that carries the Message Authentication Code for the PCP packet. The generation of the digest can be various according to the algorithms specified in different PCP SAs. This field MUST end on a 32-bit boundary, padded with 0's when necessary

8.4. EAP Payload Option



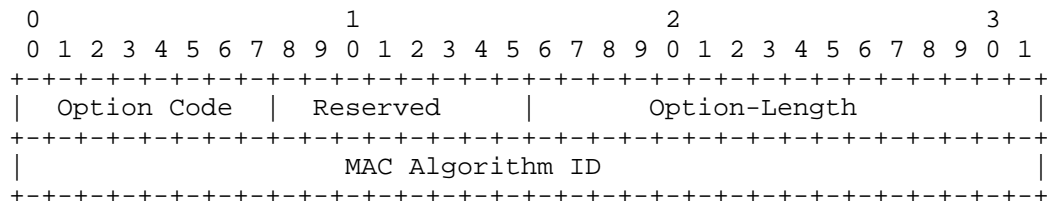
EAP Message: The EAP message transferred. Note this field MUST end on a 32-bit boundary, padded with 0's when necessary.

8.5. PRF Option



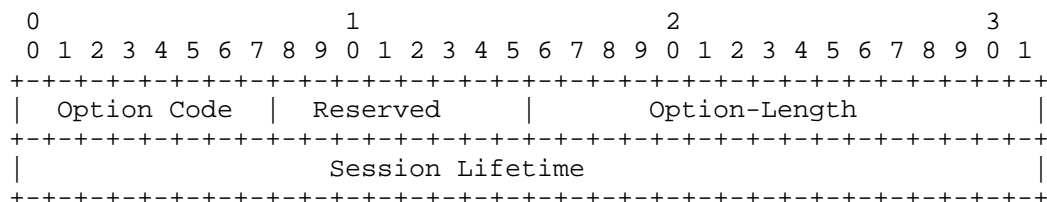
PRF: The pseudo-random Function which the sender supports to generate a MSK. This field contains an IKEv2 Transform ID of Transform Type 2 [RFC4306].

8.6. Hash Algorithm Option



MAC Algorithm ID: Indicate the MAC algorithm which the sender supports to generate authentication data. The MAC Algorithm ID field contains an IKEv2 Transform ID of Transform Type 3 [RFC4306].

8.7. Session Lifetime Option



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

9. Processing Rules

9.1. Authentication Data Generation

If a PCP SA is generated as the result of an successful EAP authentication process, every subsequent PCP message within the session MUST 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 entire PCP message (including the PCP header and Authentication Tag Option) with the MAC algorithm and the selected traffic key, and input the generated digest into the Authentication Data field.

9.2. 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 (using key ID) and the MAC algorithm. If no proper SA is found, the PCP packet MUST be discarded silently. 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 (including the PCP header and Authentication Tag Option) 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 successes. Otherwise, the packet MUST be discarded.

9.3. 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 carries 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 maintained in the SA 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 maintained in the SA 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 stored in the SA. 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. In such cases, the maintained incoming and outgoing sequence numbers will not

be affected by the message retransmission.

9.4. 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.

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):

$RT = (2 + RAND) * RT_{prev}$

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:

if $(RT > MRT)$

$RT = (1 + RAND) * MRT$

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.

9.5. MTU Considerations

EAP methods are responsible for MTU handling, so no special facilities are required in this protocol to deal with MTU issues.

10. IANA Considerations

TBD

11. Security Considerations

This section applies only to the in-band key management mechanism. It will need to be updated if the WG choose to pursue the out-of-band key management mechanism discussed above.

In this work, after a successful EAP authentication process performed between two PCP devices, a MSK will be exported. The MSK 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 a insecure network environment, and the user-identity confidentiality, protection against dictionary attacks, and session-key establishment must be supported.

12. Acknowledgements

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

13. Change Log

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

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

13.2. Changes from wasserman-pcp-authentication-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.

13.3. Changes from wasserman-pcp-authentication-00 to -01

- o Editorial changes, added use cases to introduction.

14. References

14.1. Normative References

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

14.2. Informative References

- [I-D.ietf-pcp-base]
Wing, D., Cheshire, S., Boucadair, M., Penno, R., and P. Selkirk, "Port Control Protocol (PCP)", draft-ietf-pcp-base-26 (work in progress), June 2012.
- [RFC2629] Rose, M., "Writing I-Ds and RFCs using XML", RFC 2629, June 1999.
- [RFC3748] Aboba, B., Blunk, L., Vollbrecht, J., Carlson, J., and H. Levkowetz, "Extensible Authentication Protocol (EAP)", RFC 3748, June 2004.
- [RFC4306] Kaufman, C., "Internet Key Exchange (IKEv2) Protocol", RFC 4306, December 2005.
- [RFC5191] Forsberg, D., Ohba, Y., Patil, B., Tschofenig, H., and A. Yegin, "Protocol for Carrying Authentication for Network Access (PANA)", RFC 5191, May 2008.
- [RFC5448] Arkko, J., Lehtovirta, V., and P. Eronen, "Improved Extensible Authentication Protocol Method for 3rd Generation Authentication and Key Agreement (EAP-AKA')", RFC 5448, May 2009.

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:

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

D. Wing, Ed.
Cisco
S. Cheshire
Apple
M. Boucadair
France Telecom
R. Penno
Cisco
P. Selkirk
ISC
November 7, 2012

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

Abstract

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

Status of this Memo

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

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

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

This Internet-Draft will expire on May 11, 2013.

Copyright Notice

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

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

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

Table of Contents

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

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

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

1. Introduction

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

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

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

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

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

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

2. Scope

2.1. Deployment Scenarios

PCP can be used in various deployment scenarios, including:

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

2.2. Supported Protocols

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

2.3. Single-homed Customer Premises Network

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

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

3. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in "Key words for use in RFCs to Indicate Requirement Levels" [RFC2119].

Internal Host:

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

Remote Peer Host:

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

Internal Address:

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

External Address:

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

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

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

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

Remote Peer Address:

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

Third Party:

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

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

Mapping, Port Mapping, Port Forwarding:

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

Mapping Types:

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

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

- * Implicit dynamic mappings are created implicitly as a side-effect of traffic such as an outgoing TCP SYN or outgoing UDP packet. Such packets were not originally designed explicitly for creating NAT (or firewall) state, but they can have that effect when they pass through a NAT (or firewall) device. Implicit dynamic mappings usually have a finite lifetime, though this lifetime is generally not known to the client using them.
- * Explicit dynamic mappings are created as a result of explicit PCP MAP and PEER requests. Like a DHCP address lease, explicit dynamic mappings have finite lifetime, and this lifetime is communicated to the client. As with a DHCP address lease, if the client wants a mapping to persist the client must prove that it is still present by periodically renewing the mapping to prevent it from expiring. If a PCP client goes away, then any mappings it created will be automatically cleaned up when they expire.
- * Explicit static mappings are created by manual configuration (e.g., via command-line interface or other user interface) and persist until the user changes that manual configuration.

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

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

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

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

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

PCP Client:

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

PCP-Controlled Device:

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

PCP Server:

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

Subscriber:

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

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

4. Relationship between PCP Server and its NAT/firewall

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

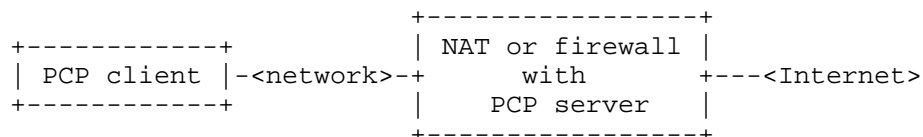


Figure 1: PCP-Enabled NAT or Firewall

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

5. Note on Fixed-Size Addresses

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

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

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

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

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

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

6. Protocol Design Note

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

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

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

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

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

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

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

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

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

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

7. Common Request and Response Header Format

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

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

7.1. Request Header

All requests have the following format:

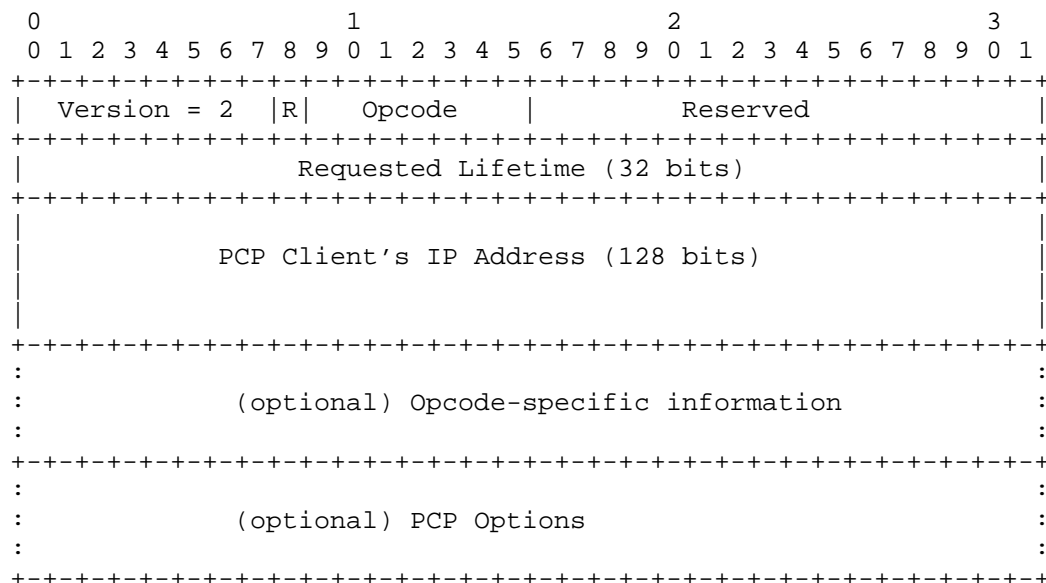


Figure 2: Common Request Packet Format

These fields are described below:

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

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

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

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

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

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

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

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

7.2. Response Header

All responses have the following format:

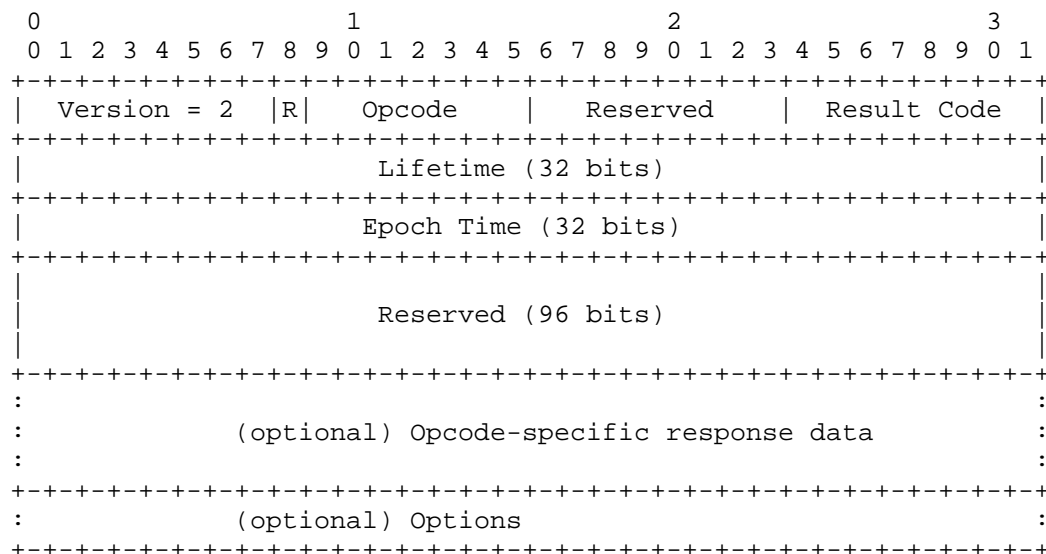


Figure 3: Common Response Packet Format

These fields are described below:

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

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

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

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

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

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

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

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

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

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

7.3. Options

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

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

Options use the following Type-Length-Value format:

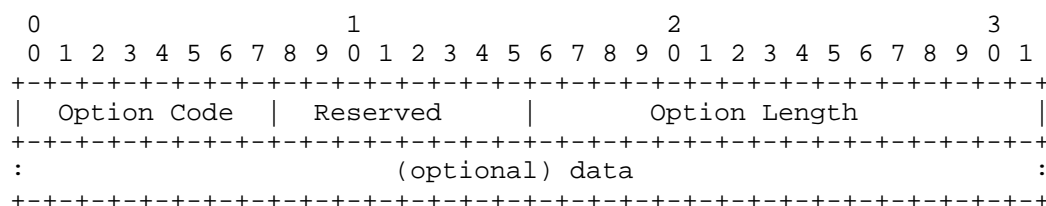


Figure 4: Options Header

The description of the fields is as follows:

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

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

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

data: Option data.

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

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

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

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

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

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

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

Different options are valid for different Opcodes. For example:

- o The THIRD_PARTY Option is valid for both MAP and PEER Opcodes.
- o The FILTER Option is valid only for the MAP Opcode (for the PEER Opcode it would have no meaning).
- o The PREFER_FAILURE Option is valid only for the MAP Opcode (for the PEER Opcode, similar semantics are automatically implied).

7.4. Result Codes

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

- 0 SUCCESS: Success.
- 1 UNSUPP_VERSION: The version number at the start of the PCP Request header is not recognized by this PCP server. This is a long lifetime error. This document describes PCP version 2.
- 2 NOT_AUTHORIZED: The requested operation is disabled for this PCP client, or the PCP client requested an operation that cannot be fulfilled by the PCP server's security policy. This is a long lifetime error.
- 3 MALFORMED_REQUEST: The request could not be successfully parsed. This is a long lifetime error.
- 4 UNSUPP_OPCODE: Unsupported Opcode. This is a long lifetime error.
- 5 UNSUPP_OPTION: Unsupported Option. This error only occurs if the Option is in the mandatory-to-process range. This is a long lifetime error.
- 6 MALFORMED_OPTION: Malformed Option (e.g., appears too many times, invalid length). This is a long lifetime error.

- 7 NETWORK_FAILURE: The PCP server or the device it controls are experiencing a network failure of some sort (e.g., has not obtained an External IP address). This is a short lifetime error.
- 8 NO_RESOURCES: Request is well-formed and valid, but the server has insufficient resources to complete the requested operation at this time. For example, the NAT device cannot create more mappings at this time, is short of CPU cycles or memory, or is unable to handle the request due to some other temporary condition. The same request may succeed in the future. This is a system-wide error, different from USER_EX_QUOTA. This can be used as a catch-all error, should no other error message be suitable. This is a short lifetime error.
- 9 UNSUPP_PROTOCOL: Unsupported transport protocol, e.g. SCTP in a NAT that handles only UDP and TCP. This is a long lifetime error.
- 10 USER_EX_QUOTA: This attempt to create a new mapping would exceed this subscriber's port quota. This is a short lifetime error.
- 11 CANNOT_PROVIDE_EXTERNAL: The suggested external port and/or external address cannot be provided. This error MUST only be returned for:
- * MAP requests that included the PREFER_FAILURE Option (normal MAP requests will return an available external port)
 - * MAP requests for the SCTP protocol (PREFER_FAILURE is implied)
 - * PEER requests
- See Section 13.2 for processing details. The error lifetime depends on the reason for the failure.
- 12 ADDRESS_MISMATCH: The source IP address of the request packet does not match the contents of the PCP Client's IP Address field, due to an unexpected NAT on the path between the PCP client and the PCP-controlled NAT or firewall. This is a long lifetime error.
- 13 EXCESSIVE_REMOTE_PEERS: The PCP server was not able to create the filters in this request. This result code MUST only be returned if the MAP request contained the FILTER Option. See Section 13.3 for processing information. This is a long lifetime error.

8. General PCP Operation

PCP messages MUST be sent over UDP [RFC0768]. Every PCP request generates at least one response, so PCP does not need to run over a reliable transport protocol.

When receiving multiple identical requests, the PCP server will generate identical responses, provided the PCP server's state did not change between those requests due to other activity. For example, if a request is received while the PCP-controlled device has no mappings available, it will generate an error response. If mappings become available and then a (duplicated or re-transmitted) request is seen by the server, it will generate a non-error response. A PCP client MUST handle such updated responses for any request it sends, most notably to support Rapid Recovery (Section 14). Also see the Protocol Design Note (Section 6).

8.1. General PCP Client: Generating a Request

This section details operation specific to a PCP client, for any Opcode. Procedures specific to the MAP Opcode are described in Section 11, and procedures specific to the PEER Opcode are described in Section 12.

Prior to sending its first PCP message, the PCP client determines which server to use. The PCP client performs the following steps to determine its PCP server:

1. if a PCP server is configured (e.g., in a configuration file or via DHCP), that single configuration source is used as the list of PCP Server(s), else;
2. the default router list (for IPv4 and IPv6) is used as the list of PCP Server(s). Thus, if a PCP client has both an IPv4 and IPv6 address, it will have an IPv4 PCP server (its IPv4 default router) for its IPv4 mappings, and an IPv6 PCP server (its IPv6 default router) for its IPv6 mappings.

For the purposes of this document, only a single PCP server address is supported. Should future specifications define configuration methods that provide a longer list of PCP server addresses, those specifications will define how clients select one or more addresses from that list.

With that PCP server address, the PCP client formulates its PCP request. The PCP request contains a PCP common header, PCP Opcode and payload, and (possibly) Options. As with all UDP client software on any operating system, when several independent PCP clients exist on the same host, each uses a distinct source port number to disambiguate their requests and replies. The PCP client's source port SHOULD be randomly generated [RFC6056].

The PCP client MUST include the source IP address of the PCP message in the PCP request. This is typically its own IP address; see

Section 16.4 for how this can be coded. This is used to detect an unexpected NAT on the path between the PCP client and the PCP-controlled NAT or firewall device, to avoid wasting state on the PCP-controlled NAT creating pointless non-functional mappings. When such an intervening non-PCP-aware inner NAT is detected, mappings must first be created by some other means in the inner NAT, before mappings can be usefully created in the outer PCP-controlled NAT. Having created mappings in the inner NAT by some other means, the PCP client should then use the inner NAT's External Address as the Client IP Address, to signal to the outer PCP-controlled NAT that the client is aware of the inner NAT, and has taken steps to create mappings in it by some other means, so that mappings created in the outer NAT will not be a pointless waste of state.

8.1.1. PCP Client Retransmission

PCP clients are responsible for reliable delivery of PCP request messages. If a PCP client fails to receive an expected response from a server, the client must retransmit its message. The retransmissions **MUST** use the same Mapping Nonce value (see Section 11.1 and Section 12.1). The client begins the message exchange by transmitting a message to the server. The message exchange continues for as long as the client wishes to maintain the mapping, and terminates when the PCP client is no longer interested in the PCP transaction (e.g., the application that requested the mapping is no longer interested in the mapping) or (optionally) when the message exchange is considered to have failed according to the retransmission mechanism described below.

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

- RT: Retransmission timeout, calculated as described below
- IRT: Initial retransmission time, **SHOULD** be 3 seconds
- MRC: Maximum retransmission count, **SHOULD** be 0 (0 indicates no maximum)
- MRT: Maximum retransmission time, **SHOULD** be 1024 seconds
- MRD: Maximum retransmission duration, **SHOULD** be 0 (0 indicates no maximum)
- RAND: Randomization factor, calculated as described below

With each message transmission or retransmission, the client sets RT according to the rules given below. If RT expires before a response

is received, the client recomputes RT and retransmits the request.

Each of the computations of a new RT include a new randomization factor (RAND), which is a random number chosen with a uniform distribution between -0.1 and +0.1. The randomization factor is included to minimize synchronization of messages transmitted by PCP clients. The algorithm for choosing a random number does not need to be cryptographically sound. The algorithm SHOULD produce a different sequence of random numbers from each invocation of the PCP client.

The RT value is initialized based on IRT:

$$RT = (1 + RAND) * IRT$$

RT for each subsequent message transmission is based on the previous value of RT, subject to the upper bound on the value of RT specified by MRT. If MRT has a value of 0, there is no upper limit on the value of RT, and MRT is treated as "infinity":

$$RT = (1 + RAND) * \text{MIN} (2 * RT_{\text{prev}}, \text{MRT})$$

MRC specifies an upper bound on the number of times a client may retransmit a message. Unless MRC is zero, the message exchange fails once the client has transmitted the message MRC times.

MRD specifies an upper bound on the length of time a client may retransmit a message. Unless MRD is zero, the message exchange fails once MRD seconds have elapsed since the client first transmitted the message.

If both MRC and MRD are non-zero, the message exchange fails whenever either of the conditions specified in the previous two paragraphs are met. If both MRC and MRD are zero, the client continues to transmit the message until it receives a response, or the client no longer wants a mapping.

Once a PCP client has successfully received a response from a PCP server on that interface, it resets RT to a value randomly selected in the range 1/2 to 5/8 of the mapping lifetime, as described in Section 11.2.1, and sends subsequent PCP requests for that mapping to that same server.

Note: If the server's state changes between retransmissions and the server's response is delayed or lost, the state in the PCP client and server may not be synchronized. This is not unique to PCP, but also occurs with other network protocols (e.g., TCP). In the unlikely event that such de-synchronization occurs, PCP heals itself after Lifetime seconds.

8.2. General PCP Server: Processing a Request

This section details operation specific to a PCP server. Processing SHOULD be performed in the order of the following paragraphs.

A PCP server MUST only accept normal (non-THIRD_PARTY) PCP requests from a client on the same interface it would normally receive packets from that client, and MUST silently ignore PCP requests arriving on any other interface. For example, a residential NAT gateway accepts PCP requests only when they arrive on its (LAN) interface connecting to the internal network, and silently ignores any PCP requests arriving on its external (WAN) interface. A PCP server which supports THIRD_PARTY requests MAY be configured to accept THIRD_PARTY requests on other configured interfaces (see Section 13.1).

Upon receiving a request, the PCP server parses and validates it. A valid request contains a valid PCP common header, one valid PCP Opcode, and zero or more Options (which the server might or might not comprehend). If an error is encountered during processing, the server generates an error response which is sent back to the PCP client. Processing an Opcode and the Options are specific to each Opcode.

Error responses have the same packet layout as success responses, with certain fields from the request copied into the response, and other fields assigned by the PCP server set as indicated in Figure 3.

Copying request fields into the response is important because this is what enables a client to identify to which request a given response pertains. For Opcodes that are understood by the PCP server, it follows the requirements of that Opcode to copy the appropriate fields. For Opcodes that are not understood by the PCP server, it simply generates the UNSUPP_OPCODE response and copies fields from the PCP header and copies the rest of the PCP payload as-is (without attempting to interpret it).

All responses (both error and success) contain the same Opcode as the request, but with the "R" bit set.

Any error response has a nonzero Result Code, and is created by:

- o Copying the entire UDP payload, or 1100 octets, whichever is less, and zero-padding the response to a multiple of 4 octets if necessary
- o Setting the R bit
- o Setting the Result Code
- o Setting the Lifetime, Epoch Time and Reserved fields
- o Updating other fields in the response, as indicated by 'set by the server' in the PCP response field description.

A success response has a zero Result Code, and is created by:

- o Copying the first four octets of request packet header
- o Setting the R bit
- o Setting the Result Code to zero
- o Setting the Lifetime, Epoch Time and Reserved fields
- o Possibly setting opcode-specific response data if appropriate
- o Adding any processed options to the response message

If the received PCP request message is less than two octets long it is silently dropped.

If the R bit is set the message is silently dropped.

If the first octet (version) is a version that is not supported, a response is generated with the UNSUPP_VERSION result code, and the other steps detailed in Section 9 are followed.

Otherwise, if the version is supported but the received message is shorter than 24 octets, the message is silently dropped.

If the server is overloaded by requests (from a particular client or from all clients), it MAY simply silently discard requests, as the requests will be retried by PCP clients, or it MAY generate the NO_RESOURCES error response.

If the length of the message exceeds 1100 octets, is not a multiple of 4 octets, or is too short for the opcode in question, it is invalid and a MALFORMED_REQUEST response is generated, and the response message is truncated to 1100 octets.

The PCP server compares the source IP address (from the received IP header) with the field PCP Client IP Address. If they do not match, the error ADDRESS_MISMATCH MUST be returned. This is done to detect and prevent accidental use of PCP where a non-PCP-aware NAT exists between the PCP client and PCP server. If the PCP client wants such a mapping it needs to ensure the PCP field matches its apparent IP address from the perspective of the PCP server.

8.3. General PCP Client: Processing a Response

The PCP client receives the response and verifies that the source IP address and port belong to the PCP server of a previously-sent PCP request. If not, the response is silently dropped.

If the received PCP response message is less than four octets long it is silently dropped.

If the R bit is clear the message is silently dropped.

If the error code is UNSUPP_VERSION processing continues as described in Section 9.

The PCP client then validates that the Opcode matches a previous PCP request. Responses shorter than 24 octets, longer than 1100 octets, or not a multiple of 4 octets are invalid and ignored, likely causing the request to be re-transmitted. The response is further matched by comparing fields in the response Opcode-specific data to fields in the request Opcode-specific data, as described by the processing for that Opcode.

After these matches are successful, the PCP client checks the Epoch Time field to determine if it needs to restore its state to the PCP server (see Section 8.5). A PCP client SHOULD be prepared to receive multiple responses from the PCP Server at any time after a single request is sent. This allows the PCP server to inform the client of mapping changes such as an update or deletion. For example, a PCP Server might send a SUCCESS response and, after a configuration change on the PCP Server, later send a NOT_AUTHORIZED response. A PCP client MUST be prepared to receive responses for requests it never sent (which could have been sent by a previous PCP instance on this same host, or by a previous host that used the same client IP address, or by a malicious attacker) by simply ignoring those unexpected messages.

If the error ADDRESS_MISMATCH is received, it indicates the presence of a NAT between the PCP client and PCP server. Procedures to resolve this problem are beyond the scope of this document.

For both success and error responses a Lifetime value is returned. The Lifetime indicates how long this request is considered valid by the server. The PCP client SHOULD impose an upper limit on this returned value (to protect against absurdly large values, e.g., 5 years), detailed in Section 15.

If the result code is 0 (SUCCESS), the request succeeded.

If the result code is not 0, the request failed, and the PCP client SHOULD NOT resend the same request for the indicated Lifetime of the error (as limited by the sanity checking detailed in Section 15).

If the PCP client has discovered a new PCP server (e.g., connected to a new network), the PCP client MAY immediately begin communicating with this PCP server, without regard to hold times from communicating with a previous PCP server.

8.4. Multi-Interface Issues

Hosts that desire a PCP mapping might be multi-interfaced (i.e., own several logical/physical interfaces). Indeed, a host can be configured with several IPv4 addresses (e.g., Wi-Fi and Ethernet) or dual-stacked. These IP addresses may have distinct reachability scopes (e.g., if IPv6 they might have global reachability scope as for Global Unicast Address (GUA, [RFC3587]) or limited scope as for Unique Local Address (ULA) [RFC4193]).

IPv6 addresses with global reachability (e.g., GUA) SHOULD be used as the source address when generating a PCP request. IPv6 addresses without global reachability (e.g., ULA [RFC4193]), SHOULD NOT be used as the source interface when generating a PCP request. If IPv6 privacy addresses [RFC4941] are used for PCP mappings, a new PCP request will need to be issued whenever the IPv6 privacy address is changed. This PCP request SHOULD be sent from the IPv6 privacy address itself. It is RECOMMENDED that the client delete its mappings to the previous privacy address after it no longer needs those old mappings.

Due to the ubiquity of IPv4 NAT, IPv4 addresses with limited scope (e.g., private addresses [RFC1918]) MAY be used as the source interface when generating a PCP request.

8.5. Epoch

Every PCP response sent by the PCP server includes an Epoch time field. This time field increments by one every second. Anomalies in the received Epoch time value provide a hint to PCP clients that a PCP server state loss may have occurred. Clients respond to such state loss hints by promptly renewing their mappings, so as to quickly restore any lost state at the PCP server.

If the PCP server resets or loses the state of its explicit dynamic Mappings (that is, those mappings created by PCP requests), due to reboot, power failure, or any other reason, it MUST reset its Epoch time to its initial starting value (usually zero) to provide this hint to PCP clients. After resetting its Epoch time, the PCP server

resumes incrementing the Epoch time value by one every second. Similarly, if the External IP Address(es) of the NAT (controlled by the PCP server) changes, the Epoch time MUST be reset. A PCP server MAY maintain one Epoch time value for all PCP clients, or MAY maintain distinct Epoch time values (per PCP client, per interface, or based on other criteria); this choice is implementation-dependent.

Whenever a client receives a PCP response, the client validates the received Epoch time value according to the procedure below, using integer arithmetic:

- o If this is the first PCP response the client has received from this PCP server, the Epoch time value is treated as necessarily valid, otherwise
 - * If the current PCP server Epoch time (`curr_server_time`) is less than the previously received PCP server Epoch time (`prev_server_time`) by more than one second, then the client treats the Epoch time as obviously invalid (time should not go backwards). The server Epoch time apparently going backwards by *up to* one second is not deemed invalid, so that minor packet re-ordering on the path from PCP Server to PCP Client does not trigger a cascade of unnecessary mapping renewals. If the server Epoch time passes this check, then further validation checks are performed:
 - + The client computes the difference between its current local time (`curr_client_time`) and the time the previous PCP response was received from this PCP server (`prev_client_time`):
`client_delta = curr_client_time - prev_client_time;`
 - + The client computes the difference between the current PCP server Epoch time (`curr_server_time`) and the previously received Epoch time (`prev_server_time`):
`server_delta = curr_server_time - prev_server_time;`
 - + If `client_delta+2 < server_delta - server_delta/16`
or `server_delta+2 < client_delta - client_delta/16`
then the client treats the Epoch time value as invalid,
else the client treats the Epoch time value as valid
- o The client records the current time values for use in its next comparison:
`prev_client_time = curr_client_time`
`prev_server_time = curr_server_time`

If the PCP client determined that the Epoch time value it received

was invalid then it concludes that the PCP server may have lost state, and promptly renews all its active port mapping leases as described in Section 16.3.1.

Notes:

- o The client clock MUST never go backwards. If `curr_client_time` is found to be less than `prev_client_time` then this is a client bug, and how the client deals with this client bug is implementation specific.
- o The calculations above are constructed to allow `client_delta` and `server_delta` to be computed as unsigned integer values.
- o The "+2" in the calculations above is to accommodate quantization errors in client and server clocks (up to one second quantization error each in server and client time intervals).
- o The "/16" in the calculations above is to accommodate inaccurate clocks in low-cost devices. This allows for a total discrepancy of up to 1/16 (6.25%) to be considered benign, e.g., if one clock were to run too fast by 3% while the other clock ran too slow by 3% then the client would not consider this difference to be anomalous or indicative of a restart having occurred. This tolerance is strict enough to be effective at detecting reboots, while not being so strict as to generate false alarms.

9. Version Negotiation

A PCP client sends its requests using PCP version number 2. Should later updates to this document specify different message formats with a version number greater than 2 it is expected that PCP servers will still support version 2 in addition to the newer version(s). However, in the event that a server returns a response with result code `UNSUPP_VERSION`, the client MAY log an error message to inform the user that it is too old to work with this server.

Should later updates to this document specify different message formats with a version number greater than 2, and backwards compatibility is desired, this first octet can be used for forward and backward compatibility.

If future PCP versions greater than 2 are specified, version negotiation proceeds as follows:

1. The client sends its first request using the highest (i.e., presumably 'best') version number it supports.

2. If the server supports that version it responds normally.
3. If the server does not support that version it replies giving a result containing the result code UNSUPP_VERSION, and the closest version number it does support (if the server supports a range of versions higher than the client's requested version, the server returns the lowest of that supported range; if the server supports a range of versions lower than the client's requested version, the server returns the highest of that supported range).
4. If the client receives an UNSUPP_VERSION result containing a version it does support, it records this fact and proceeds to use this message version for subsequent communication with this PCP server (until a possible future UNSUPP_VERSION response if the server is later updated, at which point the version negotiation process repeats).
5. If the client receives an UNSUPP_VERSION result containing a version it does not support then the client SHOULD try the next-lower version supported by the client. The attempt to use the next-lower version repeats until the client has tried version 2. If using version 2 fails, the client MAY log an error message to inform the user that it is too old to work with this server, and the client SHOULD set a timer to retry its request in 30 minutes or the returned Lifetime value, whichever is smaller. By automatically retrying in 30 minutes, the protocol accommodates an upgrade of the PCP server.

10. Introduction to MAP and PEER Opcodes

There are four uses for the MAP and PEER Opcodes defined in this document:

- o a host operating a server and wanting an incoming connection (Section 10.1);
- o a host operating a client and server on the same port (Section 10.2);
- o a host operating a client and wanting to optimize the application keepalive traffic (Section 10.3);
- o and a host operating a client and wanting to restore lost state in its NAT (Section 10.4).

These are discussed in the following sections, and a (non-normative) state diagram is provided in Section 16.5.

When operating a server (Section 10.1 and Section 10.2) the PCP client knows if it wants an IPv4 listener, IPv6 listener, or both on the Internet. The PCP client also knows if it has an IPv4 address or IPv6 address configured on one of its interfaces. It takes the union of this knowledge to decide to which of its PCP servers to send the request (e.g., an IPv4 address or an IPv6 address), and if to send one or two MAP requests for each of its interfaces (e.g., if the PCP client has only an IPv4 address but wants both IPv6 and IPv4 listeners, it sends a MAP request containing the all-zeros IPv6 address in the Suggested External Address field, and sends a second MAP request containing the all-zeros IPv4 address in the Suggested External Address field. If the PCP client has both an IPv4 and IPv6 address, and only wants an IPv4 listener, it sends one MAP request from its IPv4 address (if the PCP server supports NAT44 or IPv4 firewall) or one MAP request from its IPv6 address (if the PCP server supports NAT64). The PCP client can simply request the desired mapping to determine if the PCP server supports the desired mapping. Applications that embed IP addresses in payloads (e.g., FTP, SIP) will find it beneficial to avoid address family translation, if possible.

The MAP and PEER requests include a Suggested External IP Address field. Some PCP-controlled devices, especially CGN but also multi-homed NPTv6 networks, have a pool of public-facing IP addresses. PCP allows the client to indicate if it wants a mapping assigned on a specific address of that pool or any address of that pool. Some applications will break if mappings are created on different IP addresses (e.g., active mode FTP), so applications should carefully consider the implications of using this capability. Static mappings for that Internal Address (e.g., those created by a command-line interface on the PCP server or PCP-controlled device) may exist to a certain External Address, and if the Suggested External IP Address is the all-zeros address, PCP SHOULD assign its mappings to the same External Address, as this can also help applications using a mix of both static mappings and PCP-created mappings. If, on the other hand, the Suggested External IP Address contains a non-zero IP address the PCP Server SHOULD create a mapping to that external address, even if there are other mappings from that same Internal Address to a different External Address. Once an Internal Address has no implicit dynamic mappings and no explicit dynamic mappings in the PCP-controlled device, a subsequent implicit or explicit mapping for that Internal Address MAY be assigned to a different External Address. Generally, this re-assignment would occur when a CGN device is load balancing newly-seen Internal Addresses to its public pool of External Addresses.

The following table summarizes how various common PCP deployments use IPv6 and IPv4 addresses.

The 'internal' address is implicitly the same as the source IP address of the PCP request, except when the THIRD_PARTY option is used.

The 'external' address is the Suggested External Address field of the MAP or PEER request, and 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.

	internal	external	PCP remote peer	actual remote peer
	-----	-----	-----	-----
IPv4 firewall	IPv4	IPv4	IPv4	IPv4
IPv6 firewall	IPv6	IPv6	IPv6	IPv6
NAT44	IPv4	IPv4	IPv4	IPv4
NAT46	IPv4	IPv6	IPv4	IPv6
NAT64	IPv6	IPv4	IPv6	IPv4
NPTv6	IPv6	IPv6	IPv6	IPv6

Figure 5: Address Families with MAP and PEER

10.1. For Operating a Server

A host operating a server (e.g., a web server) listens for traffic on a port, but the server never initiates traffic from that port. For this to work across a NAT or a firewall, the host needs to (a) create a mapping from a public IP address, protocol, and port to itself as described in Section 11, (b) publish that public IP address, protocol, and port via some sort of rendezvous server (e.g., DNS, a SIP message, a proprietary protocol), and (c) ensure that any other non-PCP-speaking packet filtering middleboxes on the path (e.g., host-based firewall, network-based firewall, or other NATs) will also allow the incoming traffic. Publishing the public IP address and port is out of scope of this specification. To accomplish (a), the host follows the procedures described in this section.

As normal, the application needs to begin listening on a port. Then, the application constructs a PCP message with the MAP Opcode, with the external address set to the appropriate all-zeroes address, depending on whether it wants a public IPv4 or IPv6 address.

The following pseudo-code shows how PCP can be reliably used to operate a server:

```
/* start listening on the local server port */
int s = socket(...);
bind(s, ...);
listen(s, ...);

getsockname(s, &internal_sockaddr, ...);
bzero(&external_sockaddr, sizeof(external_sockaddr));

while (1)
{
    /* Note: The "time_to_send_pcp_request()" check below includes:
     * 1. Sending the first request
     * 2. Retransmitting requests due to packet loss
     * 3. Resending a request due to impending lease expiration
     * 4. Resending a request due to server state loss
     * The PCP packet sent is identical in all four cases; from
     * the PCP server's point of view they are the same operation.
     * The Suggested External Address and Port may be updated
     * repeatedly during the lifetime of the mapping.
     * Other fields in the packet generally remain unchanged.
     */
    if (time_to_send_pcp_request())
        pcp_send_map_request(internal_sockaddr.sin_port,
                             internal_sockaddr.sin_addr,
                             &external_sockaddr, /* will be zero the first time */
                             requested_lifetime, &assigned_lifetime);

    if (pcp_response_received())
        update_rendezvous_server("Client Ident", external_sockaddr);

    if (received_incoming_connection_or_packet())
        process_it(s);

    if (other_work_to_do())
        do_it();

    /* ... */

    block_until_we_need_to_do_something_else();
}
```

Figure 6: Pseudo-code for using PCP to operate a server

10.2. For Operating a Symmetric Client/Server

A host operating a client and server on the same port (e.g., Symmetric RTP [RFC4961] or SIP Symmetric Response Routing (rport) [RFC3581]) first establishes a local listener, (usually) sends the local and public IP addresses, protocol, and ports to a rendezvous service (which is out of scope of this document), and initiates an outbound connection from that same source address and same port. To accomplish this, the application uses the procedure described in this section.

An application that is using the same port for outgoing connections as well as incoming connections MUST first signal its operation of a server using the PCP MAP Opcode, as described in Section 11, and receive a positive PCP response before it sends any packets from that port.

Discussion: In general, a PCP client doesn't know in advance if it is behind a NAT or firewall. On detecting the host has connected to a new network, the PCP client can attempt to request a mapping using PCP, and if that succeeds then the client knows it has successfully created a mapping. If after multiple retries it has received no PCP response, then either the client is **not** behind a NAT or firewall and has unfettered connectivity, or the client **is** behind a NAT or firewall which doesn't support PCP (and the client may still have working connectivity by virtue of static mappings previously created manually by the user). Retransmitting PCP requests multiple times before giving up and assuming unfettered connectivity adds delay in that case. Initiating outbound TCP connections immediately without waiting for PCP avoids this delay, and will work if the NAT has endpoint-independent mapping EIM behavior, but may fail if the NAT has endpoint-dependent mapping EDM behavior. Waiting enough time to allow an explicit PCP MAP Mapping to be created (if possible) first ensures that the same External Port will then be used for all subsequent implicit dynamic mappings (e.g., TCP SYNs) sent from the specified Internal Address, Protocol, and Port. PCP supports both EIM and EDM NATs, so clients need to assume they may be dealing with an EDM NAT. In this case, the client will experience more reliable connectivity if it attempts explicit PCP MAP requests first, before initiating any outbound TCP connections from that Internal Address and Port. See also Section 16.1.

The following pseudo-code shows how PCP can be used to operate a symmetric client and server:

```
/* start listening on the local server port */
int s = socket(...);
bind(s, ...);
listen(s, ...);

getsockname(s, &internal_sockaddr, ...);
bzero(&external_sockaddr, sizeof(external_sockaddr));

while (1)
{
    /* Note: The "time_to_send_pcp_request()" check below includes:
     * 1. Sending the first request
     * 2. Retransmitting requests due to packet loss
     * 3. Resending a request due to impending lease expiration
     * 4. Resending a request due to server state loss
     */
    if (time_to_send_pcp_request())
        pcp_send_map_request(internal_sockaddr.sin_port,
                             internal_sockaddr.sin_addr,
                             &external_sockaddr, /* will be zero the first time */
                             requested_lifetime, &assigned_lifetime);

    if (pcp_response_received())
        update_rendezvous_server("Client Ident", external_sockaddr);

    if (received_incoming_connection_or_packet())
        process_it(s);

    if (need_to_make_outgoing_connection())
        make_outgoing_connection(s, ...);

    if (data_to_send())
        send_it(s);

    if (other_work_to_do())
        do_it();

    /* ... */

    block_until_we_need_to_do_something_else();
}
```

Figure 7: Pseudo-code for using PCP to operate a symmetric client/
server

10.3. For Reducing NAT or Firewall Keepalive Messages

A host operating a client (e.g., XMPP client, SIP client) sends from a port, and may receive responses, but never accepts incoming connections from other Remote Peers on this port. It wants to ensure the flow to its Remote Peer is not terminated (due to inactivity) by an on-path NAT or firewall. To accomplish this, the application uses the procedure described in this section.

Middleboxes such as NATs or firewalls need to see occasional traffic or will terminate their session state, causing application failures. To avoid this, many applications routinely generate keepalive traffic for the primary (or sole) purpose of maintaining state with such middleboxes. Applications can reduce such application keepalive traffic by using PCP.

Note: For reasons beyond NAT, an application may find it useful to perform application-level keepalives, such as to detect a broken path between the client and server, keep state alive on the Remote Peer, or detect a powered-down client. These keepalives are not related to maintaining middlebox state, and PCP cannot do anything useful to reduce those keepalives.

To use PCP for this function, the application first connects to its server, as normal. Afterwards, it issues a PCP request with the PEER Opcode as described in Section 12.

The following pseudo-code shows how PCP can be reliably used with a dynamic socket, for the purposes of reducing application keepalive messages:

```
int s = socket(...);
connect(s, &remote_peer, ...);

getsockname(s, &internal_sockaddr, ...);
bzero(&external_sockaddr, sizeof(external_sockaddr));

while (1)
{
    /* Note: The "time_to_send_pcp_request()" check below includes:
    * 1. Sending the first request
    * 2. Retransmitting requests due to packet loss
    * 3. Resending a request due to impending lease expiration
    * 4. Resending a request due to server state loss
    */
    if (time_to_send_pcp_request())
        pcp_send_peer_request(internal_sockaddr.sin_port,
                               internal_sockaddr.sin_addr,
                               &external_sockaddr, /* will be zero the first time */
                               remote_peer, requested_lifetime, &assigned_lifetime);

    if (data_to_send())
        send_it(s);

    if (other_work_to_do())
        do_it();

    /* ... */

    block_until_we_need_to_do_something_else();
}
```

Figure 8: Pseudo-code using PCP with a dynamic socket

10.4. For Restoring Lost Implicit TCP Dynamic Mapping State

After a NAT loses state (e.g., because of a crash or power failure), it is useful for clients to re-establish TCP mappings on the NAT. This allows servers on the Internet to see traffic from the same IP address and port, so that sessions can be resumed exactly where they were left off. This can be useful for long-lived connections (e.g., instant messaging) or for connections transferring a lot of data (e.g., FTP). This can be accomplished by first establishing a TCP connection normally and then sending a PEER request/response and remembering the External Address and External Port. Later, when the

NAT has lost state, the client can send a PEER request with the Suggested External Port and Suggested External Address remembered from the previous session, which will create a mapping in the NAT that functions exactly as an implicit dynamic mapping. The client then resumes sending TCP data to the server.

Note: This procedure works well for TCP, provided the NAT creates a new implicit dynamic outbound mapping only for TCP segments with the SYN bit set (i.e., the newly-booted NAT drops the re-transmitted data segments from the client because the NAT does not have an active mapping for those segments), and if the server is not sending data that elicits a RST from the NAT. This is not the case for UDP, because a new UDP mapping will be created (probably on a different port) as soon as UDP traffic is seen by the NAT.

11. MAP Opcode

This section defines an Opcode which controls forwarding from a NAT (or firewall) to an Internal Host.

MAP: Create an explicit dynamic mapping between an Internal Address + Port and an External Address + Port.

PCP Servers SHOULD provide a configuration option to allow administrators to disable MAP support if they wish.

Mappings created by PCP MAP requests are, by definition, Endpoint Independent Mappings (EIM) with Endpoint Independent Filtering (EIF) (unless the FILTER Option is used), even on a NAT that usually creates Endpoint Dependent Mappings (EDM) or Endpoint Dependent Filtering (EDF) for outgoing connections, since the purpose of an (unfiltered) MAP mapping is to receive inbound traffic from any remote endpoint, not from only one specific remote endpoint.

Note also that all NAT mappings (created by PCP or otherwise) are by necessity bidirectional and symmetric. For any packet going in one direction (in or out) that is translated by the NAT, a reply going in the opposite direction needs to have the corresponding opposite translation done so that the reply arrives at the right endpoint. This means that if a client creates a MAP mapping, and then later sends an outgoing packet using the mapping's Internal Address, Protocol and Port, the NAT should translate that packet's Internal Address and Port to the mapping's External Address and Port, so that replies addressed to the External Address and Port are correctly translated back to the mapping's Internal Address and Port.

On Operating Systems that allow multiple listening servers to bind to

the same internal address, protocol and port, servers **MUST** ensure that they have exclusive use of that internal address, protocol and port (e.g., by binding the port using `INADDR_ANY`, or using `SO_EXCLUSIVEADDRUSE` or similar) before sending their PCP MAP request, to ensure that no other PCP clients on the same machine are also listening on the same internal protocol and internal port.

As a side-effect of creating a mapping, ICMP messages associated with the mapping **MUST** be forwarded (and also translated, if appropriate) for the duration of the mapping's lifetime. This is done to ensure that ICMP messages can still be used by hosts, without application programmers or PCP client implementations needing to use PCP separately to create ICMP mappings for those flows.

The operation of the MAP Opcode is described in this section.

11.1. MAP Operation Packet Formats

The MAP Opcode has a similar packet layout for both requests and responses. If the Assigned External IP address and Port in the PCP response always match the Internal IP Address and Port from the PCP request, then the functionality is purely a firewall; otherwise it pertains to a network address translator which might also perform firewall-like functions.

The following diagram shows the format of the Opcode-specific information in a request for the MAP Opcode.

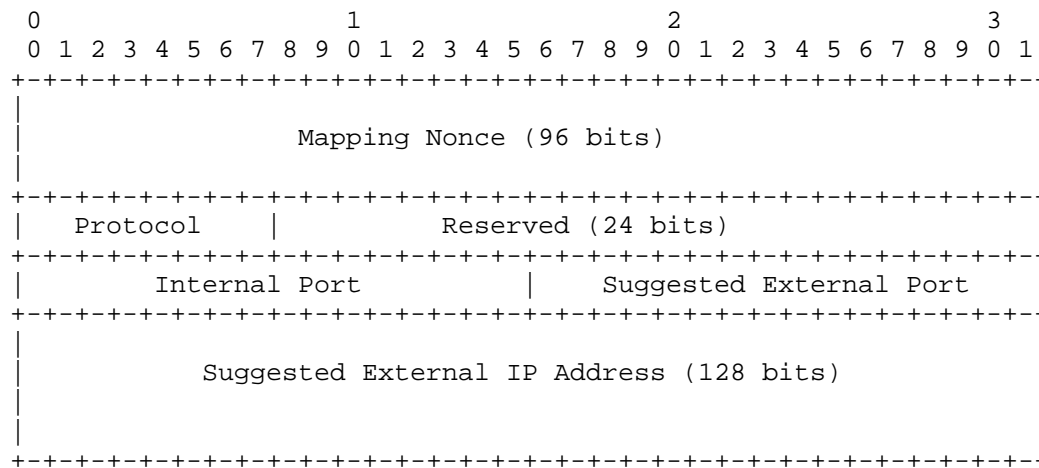


Figure 9: MAP Opcode Request

These fields are described below:

Requested lifetime (in common header): Requested lifetime of this mapping, in seconds. The value 0 indicates "delete".

Mapping Nonce: Random value chosen by the PCP client. See Section 11.2. Zero is a legal value (but unlikely, occurring in roughly one in 2^{96} requests).

Protocol: Upper-layer protocol associated with this Opcode. Values are taken from the IANA protocol registry [proto_numbers]. For example, this field contains 6 (TCP) if the Opcode is intended to create a TCP mapping. The value 0 has a special meaning for 'all protocols'.

Reserved: 24 reserved bits, MUST be sent as 0 and MUST be ignored when received.

Internal Port: Internal port for the mapping. The value 0 indicates 'all ports', and is legal when the lifetime is zero (a delete request), if the Protocol does not use 16-bit port numbers, or the client is requesting 'all ports'. If Protocol is zero (meaning 'all protocols'), then Internal Port MUST be zero on transmission and MUST be ignored on reception.

Suggested External Port: Suggested external port for the mapping. This is useful for refreshing a mapping, especially after the PCP server loses state. If the PCP client does not know the external port, or does not have a preference, it MUST use 0.

Suggested External IP Address: Suggested external IPv4 or IPv6 address. This is useful for refreshing a mapping, especially after the PCP server loses state. If the PCP client does not know the external address, or does not have a preference, it MUST use the address-family-specific all-zeroes address (see Section 5).

The internal address for the request is the source IP address of the PCP request message itself, unless the THIRD_PARTY Option is used.

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

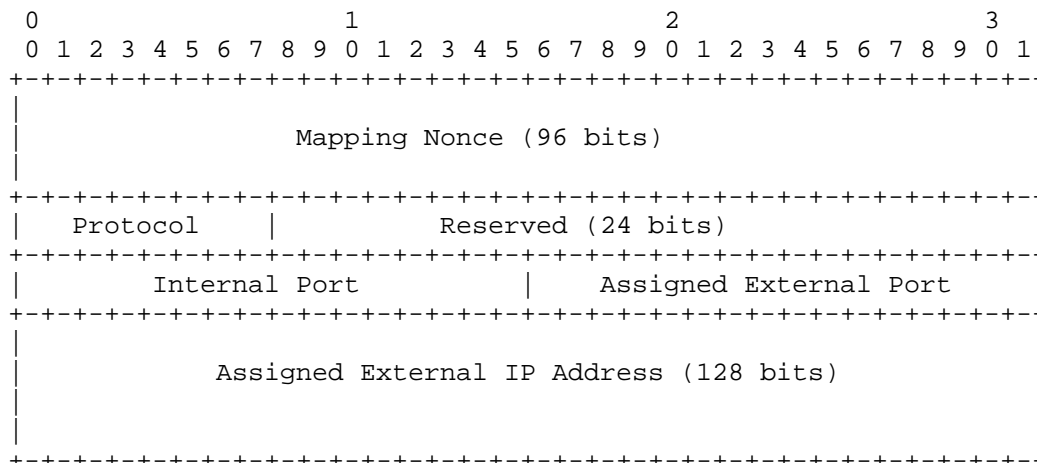


Figure 10: MAP Opcode Response

These fields are described below:

Lifetime (in common header): On an error response, this indicates how long clients should assume they'll get the same error response from the PCP server if they repeat the same request. On a success response, this indicates the lifetime for this mapping, in seconds.

Mapping Nonce: Copied from the request.

Protocol: Copied from the request.

Reserved: 24 reserved bits, MUST be sent as 0 and MUST be ignored when received.

Internal Port: Copied from the request.

Assigned External Port: On a success response, this is the assigned external port for the mapping. On an error response, the Suggested External Port is copied from the request.

Assigned External IP Address: On a success response, this is the assigned external IPv4 or IPv6 address for the mapping. An IPv4 address is encoded using IPv4-mapped IPv6 address. On an error response, the Suggested External IP Address is copied from the request.

11.2. Generating a MAP Request

This section describes the operation of a PCP client when sending requests with the MAP Opcode.

The request MAY contain values in the Suggested External Port and Suggested External IP Address fields. This allows the PCP client to attempt to rebuild lost state on the PCP server, which improves the chances of existing connections surviving, and helps the PCP client avoid having to change information maintained at its rendezvous server. Of course, due to other activity on the network (e.g., by other users or network renumbering), the PCP server may not be able to grant the suggested External IP Address, Protocol, and Port, and in that case it will assign a different External IP Address and Port.

A PCP client MUST be written assuming that it may **never** be assigned the external port it suggests. In the case of recreating state after a NAT gateway crash, the Suggested External Port, being one that was previously allocated to this client, is likely to be available for this client to continue using. In all other cases, the client MUST assume that it is unlikely that its Suggested External Port will be granted. For example, when many subscribers are sharing a Carrier-Grade NAT, popular ports such as 80, 443 and 8080 are likely to be in high demand. At most one client can have each of those popular ports for each External IP Address, and all the other clients will be assigned other, dynamically allocated, External Ports. Indeed, some ISPs may, by policy, choose not to grant those External Ports to **anyone**, so that none of their clients are **ever** assigned External Ports 80, 443 or 8080.

If the Protocol does not use 16-bit port numbers (e.g., RSVP, IP protocol number 46), the port number MUST be zero. This will cause all traffic matching that protocol to be mapped.

If the client wants all protocols mapped it uses Protocol 0 (zero) and Internal Port 0 (zero).

The Mapping Nonce value is randomly chosen by the PCP client, following accepted practices for generating unguessable random numbers [RFC4086], and is used as part of the validation of PCP responses (see below) by the PCP client, and validation for mapping refreshes by the PCP server. The client MUST use a different Mapping Nonce for each PCP server it communicates with, and it is RECOMMENDED to choose a new random Mapping Nonce whenever the PCP client is initialized. The client MAY use a different Mapping Nonce for every mapping.

11.2.1. Renewing a Mapping

An existing mapping can have its lifetime extended by the PCP client. To do this, the PCP client sends a new MAP request indicating the internal port. The PCP MAP request SHOULD also include the currently assigned external IP address and port in the Suggested External IP address and Suggested External Port fields, so if the PCP server has lost state it can recreate the lost mapping with the same parameters.

The PCP client SHOULD renew the mapping before its expiry time, otherwise it will be removed by the PCP server (see Section 15). To reduce the risk of inadvertent synchronization of renewal requests, a random jitter component should be included. It is RECOMMENDED that PCP clients send a single renewal request packet at a time chosen with uniform random distribution in the range $1/2$ to $5/8$ of expiration time. If no SUCCESS response is received, then the next renewal request should be sent $3/4$ to $3/4 + 1/16$ to expiration, and then another $7/8$ to $7/8 + 1/32$ to expiration, and so on, subject to the constraint that renewal requests MUST NOT be sent less than four seconds apart (a PCP client MUST NOT send a flood of ever-closer-together requests in the last few seconds before a mapping expires).

11.3. Processing a MAP Request

This section describes the operation of a PCP server when processing a request with the MAP Opcode. Processing SHOULD be performed in the order of the following paragraphs.

The Protocol, Internal Port, and Mapping Nonce fields from the MAP request are copied into the MAP response. If present and processed by the PCP server the THIRD_PARTY Option is also copied into the MAP response.

If the Requested Lifetime is non-zero then:

- o If both the protocol and internal port are non-zero, it indicates a request to create a mapping or extend the lifetime of an existing mapping. If the PCP server or PCP-controlled device does not support the Protocol, the UNSUPP_PROTOCOL error MUST be returned.
- o If the protocol is non-zero and the internal port is zero, it indicates a request to create or extend a mapping for all incoming traffic for that entire Protocol. If this request cannot be fulfilled in its entirety, the UNSUPP_PROTOCOL error MUST be returned.

- o If both the protocol and internal port are zero, it indicates a request to create or extend a mapping for all incoming traffic for all protocols (commonly called a "DMZ host"). If this request cannot be fulfilled in its entirety, the UNSUPP_PROTOCOL error MUST be returned.
- o If the protocol is zero and the internal port is non-zero, then the request is invalid and the PCP Server MUST return a MALFORMED_REQUEST error to the client.

If the requested lifetime is zero, it indicates a request to delete an existing mapping.

Further processing of the lifetime is described in Section 15.

If operating in the Simple Threat Model (Section 18.1), and the Internal port, Protocol, and Internal Address match an existing explicit dynamic mapping, but the Mapping Nonce does not match, the request MUST be rejected with a NOT_AUTHORIZED error with the Lifetime of the error indicating duration of that existing mapping. The PCP server only needs to remember one Mapping Nonce value for each explicit dynamic mapping.

If the Internal port, Protocol, and Internal Address match an existing static mapping (which will have no nonce) then a PCP reply is sent giving the External Address and Port of that static mapping, using the nonce from the PCP request. The server does not record the nonce.

If an Option with value less than 128 exists (i.e., mandatory to process) but that Option does not make sense (e.g., the PREFER_FAILURE Option is included in a request with lifetime=0), the request is invalid and generates a MALFORMED_OPTION error.

If the PCP-controlled device is stateless (that is, it does not establish any per-flow state, and simply rewrites the address and/or port in a purely algorithmic fashion), the PCP server simply returns an answer indicating the external IP address and port yielded by this stateless algorithmic translation. This allows the PCP client to learn its external IP address and port as seen by remote peers. Examples of stateless translators include stateless NAT64, 1:1 NAT44, and NPTv6 [RFC6296], all of which modify addresses but not port numbers.

It is possible that a mapping might already exist for a requested Internal Address, Protocol, and Port. If so, the PCP server takes the following actions:

1. If the MAP request contains the PREFER_FAILURE Option, but the Suggested External Address and Port do not match the External Address and Port of the existing mapping, the PCP server MUST return CANNOT_PROVIDE_EXTERNAL.
2. If the existing mapping is static (created outside of PCP), the PCP server MUST return the External Address and Port of the existing mapping in its response and SHOULD indicate a Lifetime of $2^{32}-1$ seconds, regardless of the Suggested External Address and Port in the request.
3. If the existing mapping is explicit dynamic inbound (created by a previous MAP request), the PCP server MUST return the existing External Address and Port in its response, regardless of the Suggested External Address and Port in the request. Additionally, the PCP server MUST update the lifetime of the existing mapping, in accordance with section 10.5.
4. If the existing mapping is dynamic outbound (created by outgoing traffic or a previous PEER request), the PCP server SHOULD create a new explicit inbound mapping, replicating the ports and addresses from the outbound mapping (but the outbound mapping continues to exist, and remains in effect if the explicit inbound mapping is later deleted).

If no mapping exists for the Internal Address, Protocol, and Port, and the PCP server is able to create a mapping using the Suggested External Address and Port, it SHOULD do so. This is beneficial for re-establishing state lost in the PCP server (e.g., due to a reboot). There are, however, cases where the PCP server is not able to create a new mapping using the Suggested External Address and Port:

- o The Suggested External Address, Protocol, and Port is already assigned to another existing explicit or implicit mapping (i.e., is already forwarding traffic to some other internal address and port).
- o The Suggested External Address, Protocol, and Port is already used by the NAT gateway for one of its own services. For example, TCP port 80 for the NAT gateway's own configuration web pages, or UDP ports 5350 and 5351, used by PCP itself. A PCP server MUST NOT create client mappings for External UDP ports 5350 or 5351.
- o The Suggested External Address, Protocol, and Port is otherwise prohibited by the PCP server's policy.
- o The Suggested External IP Address, Protocol, or Suggested Port are invalid or invalid combinations (e.g., External Address 127.0.0.1,

:::1, a multicast address, or the Suggested Port is not valid for the Protocol).

- o The Suggested External Address does not belong to the NAT gateway.
- o The Suggested External Address is not configured to be used as an external address of the firewall or NAT gateway.

If the PCP server cannot assign the Suggested External Address, Protocol, and Port, then:

- o If the request contained the PREFER_FAILURE Option, then the PCP server MUST return CANNOT_PROVIDE_EXTERNAL.
- o If the request did not contain the PREFER_FAILURE Option, and the PCP server can assign some other External Address and Port for that protocol, then the PCP server MUST do so and return the newly assigned External Address and Port in the response. In no case is the client penalized for a 'poor' choice of Suggested External Address and Port. The Suggested External Address and Port may be used by the server to guide its choice of what External Address and Port to assign, but in no case do they cause the server to fail to allocate an External Address and Port where otherwise it would have succeeded. The presence of a non-zero Suggested External Address or Port is merely a hint; it never does any harm.

By default, a PCP-controlled device MUST NOT create mappings for a protocol not indicated in the request. For example, if the request was for a TCP mapping, a UDP mapping MUST NOT be created.

Mappings typically consume state on the PCP-controlled device, and it is RECOMMENDED that a per-host and/or per-subscriber limit be enforced by the PCP server to prevent exhausting the mapping state. If this limit is exceeded, the result code USER_EX_QUOTA is returned.

If all of the preceding operations were successful (did not generate an error response), then the requested mapping is created or refreshed as described in the request and a SUCCESS response is built.

11.4. Processing a MAP Response

This section describes the operation of the PCP client when it receives a PCP response for the MAP Opcode.

After performing common PCP response processing, the response is further matched with a previously-sent MAP request by comparing the Internal IP Address (the destination IP address of the PCP response,

or other IP address specified via the THIRD_PARTY option), the Protocol, the Internal Port, and the Mapping Nonce. Other fields are not compared, because the PCP server sets those fields. The PCP server will send a Mapping Update (Section 14.2) if the mapping changes (e.g., due to IP renumbering).

If the result code is NO_RESOURCES and the request was for the creation or renewal of a mapping, then the PCP client SHOULD NOT send further requests for any new mappings to that PCP server for the (limited) value of the Lifetime. If the result code is NO_RESOURCES and the request was for the deletion of a mapping, then the PCP client SHOULD NOT send further requests of *any kind* to that PCP server for the (limited) value of the Lifetime.

On a success response, the PCP client can use the External IP Address and Port as needed. Typically the PCP client will communicate the External IP Address and Port to another host on the Internet using an application-specific rendezvous mechanism such as DNS SRV records.

As long as renewal is desired, the PCP client MUST also set a timer or otherwise schedule an event to renew the mapping before its lifetime expires. Renewing a mapping is performed by sending another MAP request, exactly as described in Section 11.2, except that the Suggested External Address and Port SHOULD be set to the values received in the response. From the PCP server's point of view a MAP request to renew a mapping is identical to a MAP request to create a new mapping, and is handled identically. Indeed, in the event of PCP server state loss, a renewal request from a PCP client will appear to the server to be a request to create a new mapping, with a particular Suggested External Address and Port, which happens to be what the PCP server previously assigned. See also Section 16.3.1.

On an error response, the client SHOULD NOT repeat the same request to the same PCP server within the lifetime returned in the response.

11.5. Address Change Events

A customer premises router might obtain a new External IP address, for a variety of reasons including a reboot, power outage, DHCP lease expiry, or other action by the ISP. If this occurs, traffic forwarded to the host's previous address might be delivered to another host which now has that address. This affects all mapping types, whether implicit or explicit. This same problem already occurs today when a host's IP address is re-assigned, without PCP and without an ISP-operated CGN. The solution is the same as today: the problems associated with host renumbering are caused by host renumbering, and are eliminated if host renumbering is avoided. PCP defined in this document does not provide machinery to reduce the

host renumbering problem.

When an Internal Host changes its Internal IP address (e.g., by having a different address assigned by the DHCP server) the NAT (or firewall) will continue to send traffic to the old IP address. Typically, the Internal Host will no longer receive traffic sent to that old IP address. Assuming the Internal Host wants to continue receiving traffic, it needs to install new mappings for its new IP address. The suggested external port field will not be fulfilled by the PCP server, in all likelihood, because it is still being forwarded to the old IP address. Thus, a mapping is likely to be assigned a new External Port number and/or External IP Address. Note that such host renumbering is not expected to happen routinely on a regular basis for most hosts, since most hosts renew their DHCP leases before they expire (or re-request the same address after reboot) and most DHCP servers honor such requests and grant the host the same address it was previously using before the reboot.

A host might gain or lose interfaces while existing mappings are active (e.g., Ethernet cable plugged in or removed, joining/leaving a Wi-Fi network). Because of this, if the PCP client is sending a PCP request to maintain state in the PCP server, it SHOULD ensure those PCP requests continue to use the same interface (e.g., when refreshing mappings). If the PCP client is sending a PCP request to create new state in the PCP server, it MAY use a different source interface or different source address.

11.6. Learning the External IP Address Alone

NAT-PMP [I-D.cheshire-nat-pmp] includes a mechanism to allow clients to learn the External IP Address alone, without also requesting a port mapping. NAT-PMP was designed for residential NAT gateways, where such an operation makes sense because the residential NAT gateway has only one External IP Address. PCP has broader scope, and also supports Carrier-Grade NATs (CGN) which may have a pool of External IP Addresses, not just one. A client may not be assigned any particular External IP Address from that pool until it has at least one implicit, explicit or static port mapping, and even then only for as long as that mapping remains valid. Client software that just wishes to display the user's External IP Address for cosmetic purposes can achieve that by requesting a short-lived mapping (e.g., to the Discard service (TCP/9 or UDP/9) or some other port) and then displaying the resulting External IP Address. However, once that mapping expires a subsequent implicit or explicit dynamic mapping might be mapped to a different external IP address.

12. PEER Opcode

This section defines an Opcode for controlling dynamic mappings.

PEER: Create a new dynamic outbound mapping to a remote peer's IP address and port, or extend the lifetime of an existing outbound mapping.

The use of this Opcodes is described in this section.

PCP Servers SHOULD provide a configuration option to allow administrators to disable PEER support if they wish.

Because a mapping created or managed by PEER behaves almost exactly like an implicit dynamic mapping created as a side-effect of a packet (e.g., TCP SYN) sent by the host, mappings created or managed using PCP PEER requests may be Endpoint Independent Mappings (EIM) or Endpoint Dependent Mappings (EDM), with Endpoint Independent Filtering (EIF) or Endpoint Dependent Filtering (EDF), consistent with the existing behavior of the NAT gateway or firewall in question for implicit outbound mappings it creates automatically as a result of observing outgoing traffic from Internal Hosts.

12.1. PEER Operation Packet Formats

The PEER Opcode allows a PCP client to create a new explicit dynamic outbound mapping (which functions similarly to an outbound mapping created implicitly when a host sends an outbound TCP SYN) or to extend the lifetime of an existing outbound mapping.

The following diagram shows the Opcode layout for the PEER Opcode. This packet format is aligned with the response packet format:

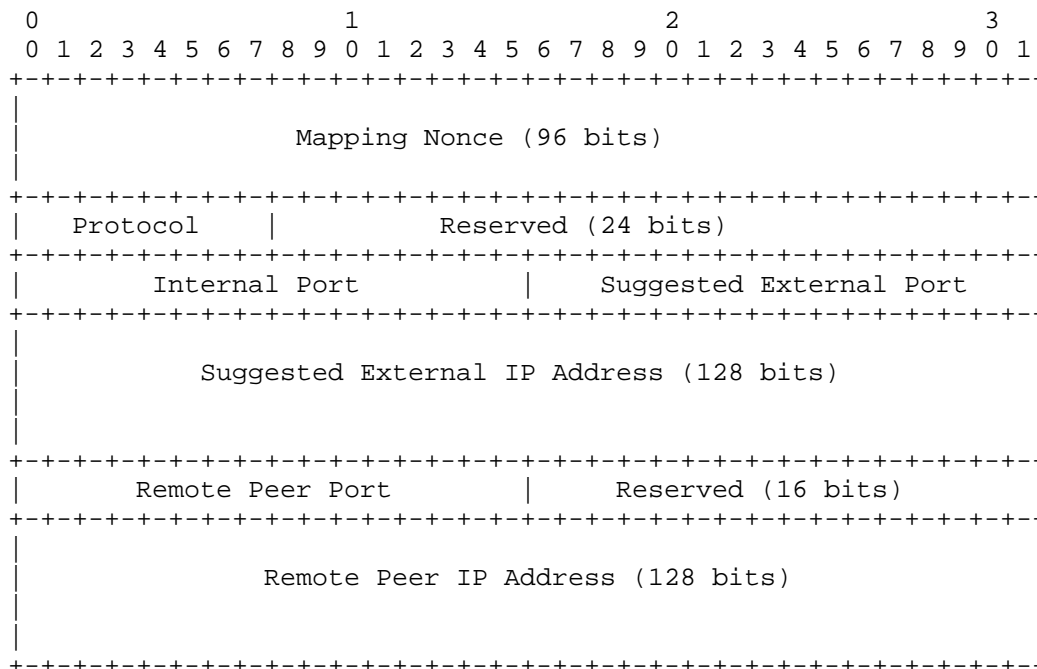


Figure 11: PEER Opcode Request

These fields are described below:

Requested Lifetime (in common header): Requested lifetime of this mapping, in seconds. Note that it is not possible to reduce the lifetime of a mapping (or delete it, with requested lifetime=0) using PEER.

Mapping Nonce: Random value chosen by the PCP client. See Section 12.2. Zero is a legal value (but unlikely, occurring in roughly one in 2^{96} requests).

Protocol: Upper-layer protocol associated with this Opcode. Values are taken from the IANA protocol registry [proto_numbers]. For example, this field contains 6 (TCP) if the Opcode is describing a TCP mapping. Protocol MUST NOT be zero.

Reserved: 24 reserved bits, MUST be set to 0 on transmission and MUST be ignored on reception.

Internal Port: Internal port for the mapping. Internal Port MUST NOT be zero.

Suggested External Port: Suggested external port for the mapping. If the PCP client does not know the external port, or does not have a preference, it MUST use 0.

Suggested External IP Address: Suggested External IP Address for the mapping. If the PCP client does not know the external address, or does not have a preference, it MUST use the address-family-specific all-zeroes address (see Section 5).

Remote Peer Port: Remote peer's port for the mapping. Remote Peer Port MUST NOT be zero.

Reserved: 16 reserved bits, MUST be set to 0 on transmission and MUST be ignored on reception.

Remote Peer IP Address: Remote peer's IP address. This is from the perspective of the PCP client, so that the PCP client does not need to concern itself with NAT64 or NAT46 (which both cause the client's idea of the remote peer's IP address to differ from the remote peer's actual IP address). This field allows the PCP client and PCP server to disambiguate multiple connections from the same port on the Internal Host to different servers. An IPv6 address is represented directly, and an IPv4 address is represented using the IPv4-mapped address syntax (Section 5).

When attempting to re-create a lost mapping, the Suggested External IP Address and Port are set to the External IP Address and Port fields received in a previous PEER response from the PCP server. On an initial PEER request, the External IP Address and Port are set to zero.

Note that semantics similar to the PREFER_FAILURE option are automatically implied by PEER requests. If the Suggested External IP Address or Suggested External Port fields are non-zero, and the PCP server is unable to honor the Suggested External IP Address, Protocol, or Port, then the PCP server MUST return a CANNOT_PROVIDE_EXTERNAL error response. The PREFER_FAILURE Option is neither required nor allowed in PEER requests, and if PCP server receives a PEER request containing the PREFER_FAILURE Option it MUST return a MALFORMED_REQUEST error response.

The following diagram shows the Opcode response for the PEER Opcode:

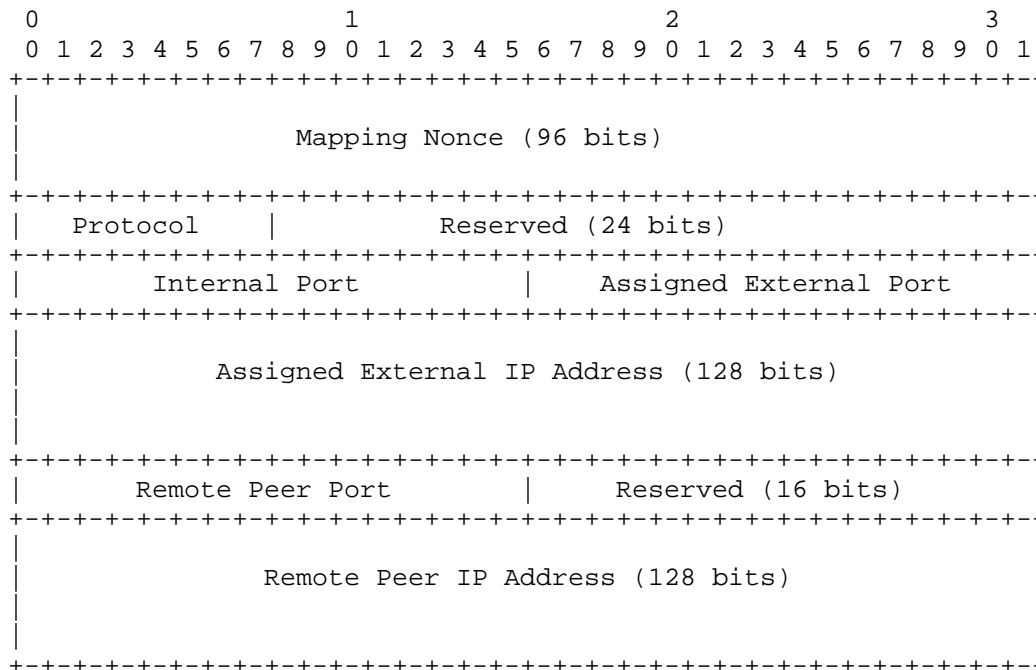


Figure 12: PEER Opcode Response

Lifetime (in common header): On a success response, this indicates the lifetime for this mapping, in seconds. On an error response, this indicates how long clients should assume they'll get the same error response from the PCP server if they repeat the same request.

Mapping Nonce: Copied from the request.

Protocol: Copied from the request.

Reserved: 24 reserved bits, MUST be set to 0 on transmission, MUST be ignored on reception.

Internal Port: Copied from request.

Assigned External Port: On a success response, this is the assigned external port for the mapping. On an error response, the Suggested External Port is copied from the request.

Assigned External IP Address: On a success response, this is the assigned external IPv4 or IPv6 address for the mapping. On an error response, the Suggested External IP Address is copied from the request.

Remote Peer port: Copied from request.

Reserved: 16 reserved bits, MUST be set to 0 on transmission, MUST be ignored on reception.

Remote Peer IP Address: Copied from the request.

12.2. Generating a PEER Request

This section describes the operation of a client when generating a message with the PEER Opcode.

The PEER Opcode MAY be sent before or after establishing bi-directional communication with the remote peer.

If sent before, this is considered a PEER-created mapping which creates a new dynamic outbound mapping in the PCP-controlled device. This is useful for restoring a mapping after a NAT has lost its mapping state (e.g., due to a crash).

If sent after, this allows the PCP client to learn the IP address, port, and lifetime of the assigned External Address and Port for the existing implicit dynamic outbound mapping, and potentially to extend this lifetime (for the purpose described in Section 10.3).

The Mapping Nonce value is randomly chosen by the PCP client, following accepted practices for generating unguessable random numbers [RFC4086], and is used as part of the validation of PCP responses (see below) by the PCP client, and validation for mapping refreshes by the PCP server. The client MUST use a different Mapping Nonce for each PCP server it communicates with, and it is RECOMMENDED to choose a new random Mapping Nonce whenever the PCP client is initialized. The client MAY use a different Mapping Nonce for every mapping.

The PEER Opcode contains a Remote Peer Address field, which is always from the perspective of the PCP client. Note that when the PCP-controlled device is performing address family translation (NAT46 or NAT64), the remote peer address from the perspective of the PCP client is different from the remote peer address on the other side of the address family translation device.

12.3. Processing a PEER Request

This section describes the operation of a server when receiving a request with the PEER Opcode. Processing SHOULD be performed in the order of the following paragraphs.

The following fields from a PEER request are copied into the response: Protocol, Internal Port, Remote Peer IP Address, Remote Peer Port, and Mapping Nonce.

When an implicit dynamic mapping is created, some NATs and firewalls validate destination addresses and will not create an implicit dynamic mapping if the destination address is invalid (e.g., 127.0.0.1). If a PCP-controlled device does such validation for implicit dynamic mappings, it SHOULD also do a similar validation of the Remote Peer IP Address, Protocol, and Port for PEER-created explicit dynamic mappings. If the validation determines the Remote Peer IP Address of a PEER request is invalid, then no mapping is created, and a MALFORMED_REQUEST error result is returned.

On receiving the PEER Opcode, the PCP server examines the mapping table for a matching five-tuple { Protocol, Internal Address, Internal Port, Remote Peer Address, Remote Peer Port }.

If no matching mapping is found, and the Suggested External Address and Port are either zero or can be honored for the specified Protocol, a new mapping is created. By having PEER create such a mapping, we avoid a race condition between the PEER request or the initial outgoing packet arriving at the NAT or firewall device first, and allow PEER to be used to recreate an outbound dynamic mapping (see last paragraph of Section 16.3.1). Thereafter, this PEER-created mapping is treated as if it was an implicit dynamic outbound mapping (e.g., as if the PCP client sent a TCP SYN) and a Lifetime appropriate to such a mapping is returned (note: on many NATs and firewalls, such mapping lifetimes are very short until the bi-directional traffic is seen by the NAT or firewall).

If no matching mapping is found, and the Suggested External Address and Port cannot be honored, then no new state is created, and the error CANNOT_PROVIDE_EXTERNAL is returned.

If a matching mapping is found, but no previous PEER Opcode was successfully processed for this mapping, then the Suggested External Address and Port values in the request are ignored, Lifetime of that mapping is adjusted as described below, and information about the existing mapping is returned. This allows a client to explicitly extend the lifetime of an existing mapping and/or to learn an existing mapping's External Address, Port and lifetime. The Mapping

Nonce is remembered for this mapping.

If operating in the Simple Threat Model (Section 18.1), and the Internal port, Protocol, and Internal Address match a mapping that already exists, but the Mapping Nonce does not match (that is, a previous PEER request was processed), the request **MUST** be rejected with a NOT_AUTHORIZED error with the Lifetime of the error indicating duration of that existing mapping. The PCP server only needs to remember one Mapping Nonce value for each mapping.

Processing the lifetime value of the PEER Opcode is described in Section 15. Sending a PEER request with a very short Requested Lifetime can be used to query the lifetime of an existing mapping.

If all of the preceding operations were successful (did not generate an error response), then a SUCCESS response is generated, with the Lifetime field containing the lifetime of the mapping.

If a PEER-created or PEER-managed mapping is not renewed using PEER, then it reverts to the NAT's usual behavior for implicit mappings, e.g., continued outbound traffic keeps the mapping alive, as per the NAT or firewall device's existing policy. A PEER-created or PEER-managed mapping may be terminated at any time by action of the TCP client or server (e.g., due to TCP FIN or TCP RST), as per the NAT or firewall device's existing policy.

12.4. Processing a PEER Response

This section describes the operation of a client when processing a response with the PEER Opcode.

After performing common PCP response processing, the response is further matched with an outstanding PEER request by comparing the Internal IP Address (the destination IP address of the PCP response, or other IP address specified via the THIRD_PARTY option), the Protocol, the Internal Port, the Remote Peer Address, the Remote Peer Port, and the Mapping Nonce. Other fields are not compared, because the PCP server sets those fields to provide information about the mapping created by the Opcode. The PCP server will send a Mapping Update (Section 14.2) if the mapping changes (e.g., due to IP renumbering).

If the result code is NO_RESOURCES and the request was for the creation or renewal of a mapping, then the PCP client **SHOULD NOT** send further requests for any new mappings to that PCP server for the (limited) value of the Lifetime.

On a successful response, the application can use the assigned

lifetime value to reduce its frequency of application keepalives for that particular NAT mapping. Of course, there may be other reasons, specific to the application, to use more frequent application keepalives. For example, the PCP assigned lifetime could be one hour but the application may want to maintain state on its server (e.g., "busy" / "away") more frequently than once an hour. If the response indicates an unexpected IP address or port (e.g., due to IP renumbering), the PCP client will want to re-establish its connection to its remote server.

If the PCP client wishes to keep this mapping alive beyond the indicated lifetime, it MAY rely on continued inside-to-outside traffic to ensure the mapping will continue to exist, or it MAY issue a new PCP request prior to the expiration. The recommended timings for renewing PEER mappings are the same as for MAP mappings, as described in Section 11.2.1.

Note: Implementations need to expect the PEER response may contain an External IP Address with a different family than the Remote Peer IP Address, e.g., when NAT64 or NAT46 are being used.

13. Options for MAP and PEER Opcodes

This section describes Options for the MAP and PEER Opcodes. These Options MUST NOT appear with other Opcodes, unless permitted by those other Opcodes.

13.1. THIRD_PARTY Option for MAP and PEER Opcodes

This Option is used when a PCP client wants to control a mapping to an Internal Host other than itself. This is used with both MAP and PEER Opcodes.

Due to security concerns with the THIRD_PARTY option, this Option MUST NOT be implemented or used unless the network on which the PCP messages are to be sent is fully trusted. For example if access control lists are installed on the PCP client, PCP server, and the network between them, so those ACLs allow only communications from a trusted PCP client to the PCP server.

A management device would use this Option to control a PCP server on behalf of users. For example, a management device located in a network operations center, which presents a user interface to end users or to network operations staff, and issues PCP requests with the THIRD_PARTY option to the appropriate PCP server.

The THIRD_PARTY Option is formatted as follows:

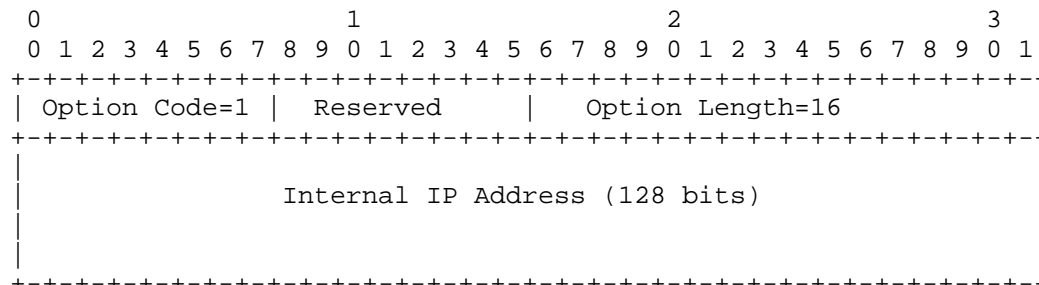


Figure 13: THIRD_PARTY Option

The fields are described below:

Internal IP Address: Internal IP address for this mapping.

Option Name: THIRD_PARTY

Number: 1

Purpose: Indicates the MAP or PEER request is for a host other than the host sending the PCP Option.

Valid for Opcodes: MAP, PEER

Length: 16 octets

May appear in: request. May appear in response only if it appeared in the associated request.

Maximum occurrences: 1

A THIRD_PARTY Option MUST NOT contain the same address as the source address of the packet. This is because many PCP servers may not implement the THIRD_PARTY Option at all, and with those servers a client redundantly using the THIRD_PARTY Option to specify its own IP address would cause such mapping requests to fail where they would otherwise have succeeded. A PCP server receiving a THIRD_PARTY Option specifying the same address as the source address of the packet MUST return a MALFORMED_REQUEST result code.

A PCP server MAY be configured to permit or to prohibit the use of the THIRD_PARTY Option. If this Option is permitted, properly authorized clients may perform these operations on behalf of other hosts. If this Option is prohibited, and a PCP server receives a PCP MAP request with a THIRD_PARTY Option, it MUST generate a UNSUPP_OPTION response.

It is RECOMMENDED that customer premises equipment implementing a PCP Server be configured to prohibit third party mappings by default. With this default, if a user wants to create a third party mapping,

the user needs to interact out-of-band with their customer premises router (e.g., using its HTTP administrative interface).

It is RECOMMENDED that service provider NAT and firewall devices implementing a PCP Server be configured to permit the THIRD_PARTY Option, when sent by a properly authorized host. If the packet arrives from an unauthorized host, the PCP server MUST generate an UNSUPP_OPTION error.

Note that the THIRD_PARTY Option is not needed for today's common scenario of an ISP offering a single IP address to a customer who is using NAT to share that address locally, since in this scenario all the customer's hosts appear, from the point of view of the ISP, to be a single host.

When a PCP client is using the THIRD_PARTY Option to make and maintain mappings on behalf of some other device, it may be beneficial if, where possible, the PCP client verifies that the other device is actually present and active on the network. Otherwise the PCP client risks maintaining those mappings forever, long after the device that required them has gone. This would defeat the purpose of PCP mappings having a finite lifetime so that they can be automatically deleted after they are no longer needed.

13.2. PREFER_FAILURE Option for MAP Opcode

This Option is only used with the MAP Opcode.

This Option indicates that if the PCP server is unable to map both the Suggested External Port and Suggested External Address, the PCP server should not create a mapping. This differs from the behavior without this Option, which is to create a mapping.

The PREFER_FAILURE Option is formatted as follows:

0										1										2										3									
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
Option Code=2										Reserved										Option Length=0																			

Figure 14: PREFER_FAILURE Option

Option Name: PREFER_FAILURE

Number: 2

Purpose: indicates that the PCP server should not create an alternative mapping if the suggested external port and address cannot be mapped.

Valid for Opcodes: MAP

Length: 0

May appear in: request. May appear in response only if it appeared in the associated request.

Maximum occurrences: 1

The result code CANNOT_PROVIDE_EXTERNAL is returned if the Suggested External Address, Protocol, and Port cannot be mapped. This can occur because the External Port is already mapped to another host's outbound dynamic mapping, an inbound dynamic mapping, a static mapping, or the same Internal Address, Protocol, and Port already has an outbound dynamic mapping which is mapped to a different External Port than suggested. This can also occur because the External Address is no longer available (e.g., due to renumbering). The server MAY set the Lifetime in the response to the remaining lifetime of the conflicting mapping + TIME_WAIT [RFC0793], rounded up to the next larger integer number of seconds.

PREFER_FAILURE is never necessary for a PCP client to manage mappings for itself, and its use causes additional work in the PCP client and in the PCP server. This Option exists for interworking with non-PCP mapping protocols that have different semantics than PCP (e.g., UPnP IGDv1 interworking [I-D.ietf-pcp-upnp-igd-interworking], where the semantics of UPnP IGDv1 only allow the UPnP IGDv1 client to dictate mapping a specific port), or separate port allocation systems which allocate ports to a subscriber (e.g., a subscriber-accessed web portal operated by the same ISP that operates the PCP server). A PCP server MAY support this Option, if its designers wish to support such downstream devices or separate port allocation systems. PCP servers that are not intended to interface with such systems are not required to support this Option. PCP clients other than UPnP IGDv1 interworking clients or other than a separate port allocation system SHOULD NOT use this Option because it results in inefficient operation, and they cannot safely assume that all PCP servers will implement it. It is anticipated that this Option will be deprecated in the future as more clients adopt PCP natively and the need for this Option declines.

If a PCP request contains the PREFER_FAILURE option and has zero in the Suggested External Port field, or has the all-zeros IPv4 or all-zeros IPv6 address in the Suggested External Address field, it is invalid. The PCP server MUST reject such a message with the MALFORMED_OPTION error code.

PCP servers MAY choose to rate-limit their handling of PREFER_FAILURE requests, to protect themselves from a rapid flurry of 65535 consecutive PREFER_FAILURE requests from clients probing to discover which external ports are available.

There can exist a race condition between the MAP Opcode using the PREFER_FAILURE option and Mapping Update (Section 14.2). For example, a previous host on the local network could have previously had the same Internal Address, with a mapping for the same Internal Port. At about the same moment that the current host sends a MAP Request using the PREFER_FAILURE option, the PCP server could send a spontaneous mapping update for the old mapping due to an external configuration change, which could appear to be a reply to the new mapping request. Because of this, the PCP client MUST validate that the External IP Address, Protocol, Port and Nonce in a success response matches the associated suggested values from the request. If they don't match, it is because the Mapping Update was sent before the MAP request was processed.

13.3. FILTER Option for MAP Opcode

This Option is only used with the MAP Opcode.

This Option indicates that filtering incoming packets is desired. The protocol being filtered is indicated by the Protocol field in the MAP Request, and the Remote Peer IP Address and Remote Peer Port of the FILTER Option indicate the permitted remote peer's source IP address and source port for packets from the Internet; other traffic from other addresses is blocked. The remote peer prefix length indicates the length of the remote peer's IP address that is significant; this allows a single Option to permit an entire subnet. After processing this MAP request containing the FILTER Option and generating a successful response, the PCP-controlled device will drop packets received on its public-facing interface that don't match the filter fields. After dropping the packet, if its security policy allows, the PCP-controlled device MAY also generate an ICMP error in response to the dropped packet.

The use of the FILTER Option can be seen as a performance optimization. Since all software using PCP to receive incoming connections also has to deal with the case where it may be directly connected to the Internet and receive unrestricted incoming TCP connections and UDP packets, if it wishes to restrict incoming traffic to a specific source address or group of source addresses such software already needs to check the source address of incoming traffic and reject unwanted traffic. However, the FILTER Option is a particularly useful performance optimization for battery powered wireless devices, because it can enable them to conserve battery

power by not having to wake up just to reject unwanted traffic.

The FILTER Option is formatted as follows:

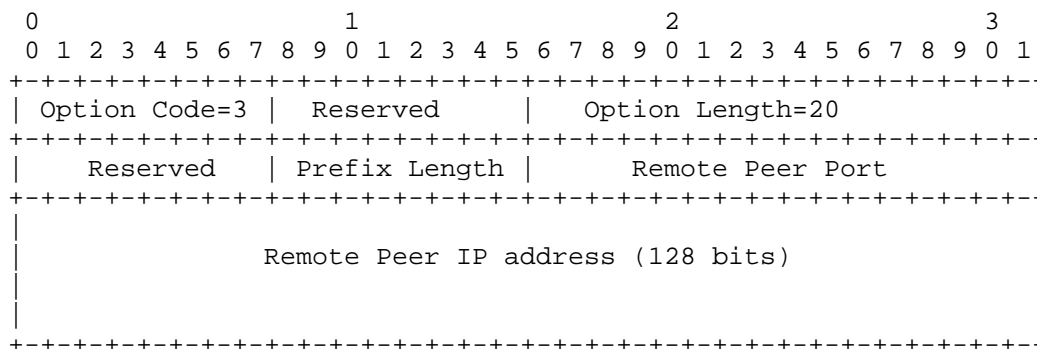


Figure 15: FILTER Option layout

These fields are described below:

Reserved: 8 reserved bits, MUST be sent as 0 and MUST be ignored when received.

Prefix Length: indicates how many bits of the IPv4 or IPv6 address are relevant for this filter. The value 0 indicates "no filter", and will remove all previous filters. See below for detail.

Remote Peer Port: the port number of the remote peer. The value 0 indicates "all ports".

Remote Peer IP address: The IP address of the remote peer.

Option Name: FILTER

Number: 3

Purpose: specifies a filter for incoming packets

Valid for Opcodes: MAP

Length: 20 octets

May appear in: request. May appear in response only if it appeared in the associated request.

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

The Prefix Length indicates how many bits of the address are used for the filter. For IPv4 addresses (which are encoded using the IPv4-mapped address format (::FFFF:0:0/96)), this means valid prefix lengths are between 96 and 128 bits, inclusive. That is, add 96 to the IPv4 prefix length. For IPv6 addresses, valid prefix lengths are

between 0 and 128 bits, inclusive. Values outside those ranges cause the PCP server to return the MALFORMED_OPTION result code.

If multiple occurrences of the FILTER Option exist in the same MAP request, they are processed in the order received (as per normal PCP Option processing) and they MAY overlap the filtering requested. If an existing mapping exists (with or without a filter) and the server receives a MAP request with FILTER, the filters indicated in the new request are added to any existing filters. If a MAP request has a lifetime of 0 and contains the FILTER Option, the error MALFORMED_OPTION is returned.

If any occurrences of the FILTER Option in a request packet are not successfully processed then an error is returned (e.g., MALFORMED_OPTION if one of the Options was malformed) and as with other PCP errors, returning an error causes no state to be changed in the PCP server or in the PCP-controlled device.

To remove all existing filters, the Prefix Length 0 is used. There is no mechanism to remove a specific filter.

To change an existing filter, the PCP client sends a MAP request containing two FILTER Options, the first Option containing a Prefix Length of 0 (to delete all existing filters) and the second containing the new remote peer's IP address, protocol, and port. Other FILTER Options in that PCP request, if any, add more allowed Remote Peers.

The PCP server or the PCP-controlled device is expected to have a limit on the number of remote peers it can support. This limit might be as small as one. If a MAP request would exceed this limit, the entire MAP request is rejected with the result code EXCESSIVE_REMOTE_PEERS, and the state on the PCP server is unchanged.

All PCP servers MUST support at least one filter per MAP mapping.

14. Rapid Recovery

PCP includes a rapid recovery feature, which allows PCP clients to repair failed mappings within seconds, rather than the minutes or hours it might take if they relied solely on waiting for the next routine renewal of the mapping. Mapping failures may occur when a NAT gateway is rebooted and loses its mapping state, or when a NAT gateway has its external IP address changed so that its current mapping state becomes invalid.

The PCP rapid recovery feature enables users to, for example, connect

to remote machines using ssh, and then reboot their NAT or firewall device (or even replace it with completely new hardware) without losing their established ssh connections.

Use of PCP rapid recovery is a performance optimization to PCP's routine self-healing. Without rapid recovery, PCP clients will still recreate their correct state when they next renew their mappings, but this routine self-healing process may take hours rather than seconds, and will probably not happen fast enough to prevent active TCP connections from timing out.

There are two mechanisms to perform rapid recovery, described below. A PCP server that can lose state (e.g., due to reboot) or might have a mapping change (e.g., due to IP renumbering) MUST implement either the Announce Opcode or the Mapping Update mechanism and SHOULD implement both mechanisms. Failing to implement and deploy a rapid recovery mechanism will encourage application developers to feel the need to refresh their PCP state more frequently than necessary, causing more network traffic.

14.1. ANNOUNCE Opcode

This rapid recovery mechanism uses the ANNOUNCE Opcode. When the PCP server loses its state (e.g., it lost its state when rebooted), it sends the ANNOUNCE response to the link-scoped multicast address (specific address explained below) if a multicast network exists on its local interface or, if configured with the IP address(es) and port(s) of PCP client(s), sends unicast ANNOUNCE responses to those address(es) and port(s). This means ANNOUNCE may not be available on all networks (such as networks without a multicast link between the PCP server and its PCP clients). Additionally, an ANNOUNCE request can be sent (unicast) by a PCP client which elicits a unicast ANNOUNCE response like any other Opcode.

14.1.1. ANNOUNCE Operation

The PCP ANNOUNCE Opcode requests and responses have no Opcode-specific payload (that is, the length of the Opcode-specific data is zero). The Requested Lifetime field of requests and Lifetime field of responses are both set to 0 on transmission and ignored on reception.

If a PCP server receives an ANNOUNCE request, it first parses it and generates a SUCCESS if parsing and processing of ANNOUNCE is successful. An error is generated if the Client's IP Address field does not match the packet source address, or the request packet is otherwise malformed, such as packet length less than 24 octets. Note that, in the future, Options MAY be sent with the PCP ANNOUNCE Opcode; PCP clients and servers need to be prepared to receive

Options with the ANNOUNCE Opcode.

Discussion: Client-to-server request messages are sent to listening UDP port 5351 on the server; server-to-client multicast notifications are sent to listening UDP port 5350 on the client. The reason the same UDP port is not used for both purposes is that a single device may have multiple roles. For example, a multi-function home gateway that provides NAT service (PCP server) may also provide printer sharing (which wants a PCP client), or a home computer (PCP client) may also provide "Internet Sharing" (NAT) functionality (which needs to offer PCP service). Such devices need to act as both a PCP Server and a PCP Client at the same time, and the software that implements the PCP Server on the device may not be the same software component that implements the PCP Client. The software that implements the PCP Server needs to listen for unicast client requests, whereas the software that implements the PCP Client needs to listen for multicast restart announcements. In many networking APIs it is difficult or impossible to have two independent clients listening for both unicasts and multicasts on the same port at the same time. For this reason, two ports are used.

14.1.2. Generating and Processing a Solicited ANNOUNCE Message

The PCP ANNOUNCE Opcode MAY be sent (unicast) by a PCP client. The Requested Lifetime value MUST be set to zero.

When the PCP server receives the ANNOUNCE Opcode and successfully parses and processes it, it generates SUCCESS response with an Assigned Lifetime of zero.

This functionality allows a PCP client to determine a server's Epoch, or to determine if a PCP server is running, without changing the server's state.

14.1.3. Generating and Processing an Unsolicited ANNOUNCE Message

When sending unsolicited responses, the ANNOUNCE Opcode MUST have Result Code equal to zero (SUCCESS), and the packet MUST be sent from the unicast IP address and UDP port number on which PCP requests are received (so PCP response processing accepts the message, see Section 8.3). This message is most typically multicast, but can also be unicast. Multicast PCP restart announcements are sent to 224.0.0.1:5350 and/or [ff02::1]:5350, as described below. Sending PCP restart announcements via unicast requires that the PCP server know the IP address(es) and port(s) of its listening clients, which means that sending PCP restart announcements via unicast is only applicable to PCP servers that retain knowledge of the IP address(es)

and port(s) of their clients even after they otherwise lose the rest of their state.

When a PCP server device that implements this functionality reboots, restarts its NAT engine, or otherwise enters a state where it may have lost some or all of its previous mapping state (or enters a state where it doesn't even know whether it may have had prior state that it lost) it **MUST** inform PCP clients of this fact by unicasting or multicasting a gratuitous PCP ANNOUNCE Opcode response packet, as shown below, via paths over which it accepts PCP requests. If sending a multicast ANNOUNCE message, a PCP server device which accepts PCP requests over IPv4 sends the Restart Announcement to the IPv4 multicast address 224.0.0.1:5350 (224.0.0.1 is the All Hosts multicast group address), and a PCP server device which accepts PCP requests over IPv6 sends the Restart Announcement to the IPv6 multicast address [ff02::1]:5350 (ff02::1 is for all nodes on the local segment). A PCP server device which accepts PCP requests over both IPv4 and IPv6 sends a pair of Restart Announcements, one to each multicast address. If sending a unicast ANNOUNCE messages, it sends ANNOUNCE response message to the IP address(es) and port(s) of its PCP clients. To accommodate packet loss, the PCP server device **MAY** transmit such packets (or packet pairs) up to ten times (with an appropriate Epoch time value in each to reflect the passage of time between transmissions) provided that the interval between the first two notifications is at least 250ms, and the interval between subsequent notification at least doubles.

A PCP client that sends PCP requests to a PCP Server via a multicast-capable path, and implements the Restart Announcement feature, and wishes to receive these announcements, **MUST** listen to receive these PCP Restart Announcements (gratuitous PCP ANNOUNCE Opcode response packets) on the appropriate multicast-capable interfaces on which it sends PCP requests, and **MAY** also listen for unicast announcements from the server too, (using the UDP port it already uses to issue unicast PCP requests to, and receive unicast PCP responses from, that server). A PCP client device which sends PCP requests using IPv4 listens for packets sent to the IPv4 multicast address 224.0.0.1:5350. A PCP client device which sends PCP requests using IPv6 listens for packets sent to the IPv6 multicast address [ff02::1]:5350. A PCP client device which sends PCP requests using both IPv4 and IPv6 listens for both types of Restart Announcement. The `SO_REUSEPORT` socket option or equivalent should be used for the multicast UDP port, if required by the host OS to permit multiple independent listeners on the same multicast UDP port.

Upon receiving a unicasted or multicasted PCP ANNOUNCE Opcode response packet, a PCP client **MUST** (as it does with all received PCP response packets) inspect the Announcement's source IP address, and

if the Epoch time value is outside the expected range for that server, it MUST wait a random amount of time between 0 and 5 seconds (to prevent synchronization of all PCP clients), then for all PCP mappings it made at that server address the client issues new PCP requests to recreate any lost mapping state. The use of the Suggested External IP Address and Suggested External Port fields in the client's renewal requests allows the client to remind the restarted PCP server device of what mappings the client had previously been given, so that in many cases the prior state can be recreated. For PCP server devices that reboot relatively quickly it is usually possible to reconstruct lost mapping state fast enough that existing TCP connections and UDP communications do not time out, and continue without failure. As for all PCP response messages, if the Epoch time value is within the expected range for that server, the PCP client does not recreate its mappings. As for all PCP response messages, after receiving and validating the ANNOUNCE message, the client updates its own Epoch time for that server, as described in Section 8.5.

14.2. PCP Mapping Update

This rapid recovery mechanism is used when the PCP server remembers its state and determines its existing mappings are invalid (e.g., IP renumbering changes the External IP Address of a PCP-controlled NAT).

It is anticipated that servers which are routinely reconfigured by an administrator or have their WAN address changed frequently will implement this feature (e.g., residential CPE routers). It is anticipated that servers which are not routinely reconfigured will not implement this feature (e.g., service provider-operated CGN).

If a PCP server device has not forgotten its mapping state, but for some other reason has determined that some or all of its mappings have become unusable (e.g., when a home gateway is assigned a different external IPv4 address by the upstream DHCP server) then the PCP server device automatically repairs its mappings and notifies its clients by following the procedure described below.

For PCP-managed mappings, for each one the PCP server device should update the External IP Address and External Port to appropriate available values, and then send unicast PCP MAP or PEER responses (as appropriate for the mapping) to inform the PCP client of the new External IP Address and External Port. Such unsolicited responses are identical to the MAP or PEER responses normally returned in response to client MAP or PEER requests, containing newly updated External IP Address and External Port values, and are sent to the same client IP address and port that the PCP server used to send the prior response for that mapping. If the earlier associated request

contained the THIRD_PARTY Option, the THIRD_PARTY Option MUST also appear in the Mapping Update as it is necessary for the PCP client to disambiguate the response. If the earlier associated request contained the PREFER_FAILURE option, and the same external IP address, protocol, and port cannot be provided, the error CANNOT_PROVIDE_EXTERNAL SHOULD be sent. If the earlier associated request contained the FILTER option, the filters are moved to the new mapping and the FILTER Option is sent in the Mapping Update response. Non-mandatory Options SHOULD NOT be sent in the Mapping Update response.

Discussion: It could have been possible to design this so that the PCP server (1) sent an ANNOUNCE Opcode to the PCP client, the PCP client reacted by (2) sending a new MAP request and (3) receiving a MAP response. Instead, that design is short-cutted by the server simply sending the message it would have sent in (3).

To accommodate packet loss, the PCP server device SHOULD transmit such packets 3 times, with an appropriate Epoch time value in each to reflect the passage of time between transmissions. The interval between the first two notifications MUST be at least 250ms, and the third packet after a 500ms interval. Once the PCP server has received a refreshed state for that mapping, the PCP server SHOULD cease those retransmissions for that mapping, as it serves no further purpose to continue sending messages regarding that mapping.

Upon receipt of such an updated MAP or PEER response, a PCP client uses the information in the response to adjust rendezvous servers or re-connect to servers, respectively. For MAP, this would mean updating the DNS entries or other address and port information recorded with some kind of application-specific rendezvous server. For PEER responses giving a CANNOT_PROVIDE_EXTERNAL error, this would typically mean establishing new connections to servers. Any time the external address or port changes, existing TCP and UDP connections will be lost; PCP can't avoid that, but does provide immediate notification of the event to lessen the impact.

15. Mapping Lifetime and Deletion

The PCP client requests a certain lifetime, and the PCP server responds with the assigned lifetime. The PCP server MAY grant a lifetime smaller or larger than the requested lifetime. The PCP server SHOULD be configurable for permitted minimum and maximum lifetime, and the minimum value SHOULD be 120 seconds. The maximum value SHOULD be the remaining lifetime of the IP address assigned to the PCP client if that information is available (e.g., from the DHCP server), or half the lifetime of IP address assignments on that

network if the remaining lifetime is not available, or 24 hours. Excessively long lifetimes can cause consumption of ports even if the Internal Host is no longer interested in receiving the traffic or is no longer connected to the network. These recommendations are not strict, and deployments should evaluate the trade offs to determine their own minimum and maximum lifetime values.

Once a PCP server has responded positively to a MAP request for a certain lifetime, the port mapping is active for the duration of the lifetime unless the lifetime is reduced by the PCP client (to a shorter lifetime or to zero) or until the PCP server loses its state (e.g., crashes). Mappings created by PCP MAP requests are not special or different from mappings created in other ways. In particular, it is implementation-dependent if outgoing traffic extends the lifetime of such mappings beyond the PCP-assigned lifetime. PCP clients **MUST NOT** depend on this behavior to keep mappings active, and **MUST** explicitly renew their mappings as required by the Lifetime field in PCP response messages.

Upon receipt of a PCP response with an absurdly long Assigned Lifetime the PCP client **SHOULD** behave as if it received a more sane value (e.g., 24 hours), and renew the mapping accordingly, to ensure that if the static mapping is removed the client will continue to maintain the mapping it desires.

An application that forgets its PCP-assigned mappings (e.g., the application or OS crashes) will request new PCP mappings. This may consume port mappings, if the application binds to a different Internal Port every time it runs. The application will also likely initiate new implicit dynamic outbound mappings without using PCP, which will also consume port mappings. If there is a port mapping quota for the Internal Host, frequent restarts such as this may exhaust the quota and using the same Mapping Nonce can help alleviate such exhaustion.

To help clean PCP state, it is **RECOMMENDED** that devices which combine IP address assignment (e.g., DHCP server) with the PCP server function (e.g., such as a residential CPE) flush PCP state when an IP address is allocated to a new host, because the new host will be unable perform the functions described in the previous paragraph because the new host does not know the previous host's Mapping Nonce value. It is good hygiene to also flush TCP and UDP flow state of NAT or firewall functions, although out of scope of this document.

To reduce unwanted traffic and data corruption for both TCP and UDP, the Assigned External Port created by the MAP Opcode or PEER Opcode **SHOULD NOT** be re-used for the same interval enforced by NAT for implicitly creating mappings, which is typically the maximum segment

lifetime interval of 120 seconds [RFC0793]. To reduce port stealing attacks, the Assigned External Port SHOULD NOT be re-used by the same Client IP Address (or Internal IP Address if using the THIRD_PARTY Option) for the duration the PCP-controlled device keeps a mapping for active bi-directional traffic (e.g., 2 minutes for UDP [RFC4787], 2 hours 4 minutes for TCP [RFC5382]). However, within the above times, the PCP server SHOULD allow a request using the same Client IP Address (and same Internal IP Address if using the THIRD_PARTY Option), Internal Port, and Mapping Nonce to re-acquire the same External Port.

The assigned lifetime is calculated by subtracting (a) zero or the number of seconds since the internal host sent a packet for this mapping from (b) the lifetime the PCP-controlled device uses for transitory connection idle-timeout (e.g., a NAT device might use 2 minutes for UDP [RFC4787] or 4 minutes for TCP [RFC5382]). If the result is a negative number, the assigned lifetime is 0.

15.1. Lifetime Processing for the MAP Opcode

If the the requested lifetime is zero then:

- o If both the protocol and internal port are non-zero, it indicates a request to delete the indicated mapping immediately.
- o If the protocol is non-zero and the internal port is zero, it indicates a request to delete a previous 'wildcard' (all-ports) mapping for that protocol.
- o If both the protocol and internal port are zero, it indicates a request to delete all mappings for this Internal Address for all transport protocols. Such a request is rejected with a NOT_AUTHORIZED error. To delete all mappings the client has to send separate MAP requests with appropriate Mapping Nonce values.
- o If the protocol is zero and the internal port is non-zero, then the request is invalid and the PCP Server MUST return a MALFORMED_REQUEST error to the client.

In requests where the requested Lifetime is 0, the Suggested External Address and Suggested External Port fields MUST be set to zero on transmission and MUST be ignored on reception, and these fields MUST be copied into the Assigned External IP Address and Assigned External Port of the response.

PCP MAP requests can only delete or shorten lifetimes of MAP-created mappings. If the PCP client attempts to delete a static mapping (i.e., a mapping created outside of PCP itself), or an outbound

(implicit or PEER-created) mapping, the PCP server MUST return NOT_AUTHORIZED. If the PCP client attempts to delete a mapping that does not exist, the SUCCESS result code is returned (this is necessary for PCP to return the same response for the same request). If the deletion request was properly formatted and successfully processed, a SUCCESS response is generated with the assigned lifetime of the mapping and the server copies the protocol and internal port number from the request into the response. An inbound mapping (i.e., static mapping or MAP- created dynamic mapping) MUST NOT have its lifetime reduced by transport protocol messages (e.g., TCP RST, TCP FIN). Note the THIRD_PARTY Option, if authorized, can also delete PCP-created mappings (see Section 13.1).

16. Implementation Considerations

Section 16 provides non-normative guidance that may be useful to implementers.

16.1. Implementing MAP with EDM port-mapping NAT

For implicit dynamic outbound mappings, some existing NAT devices have endpoint-independent mapping (EIM) behavior while other NAT devices have endpoint-dependent mapping (EDM) behavior. NATs which have EIM behavior do not suffer from the problem described in this section. The IETF strongly encourages EIM behavior [RFC4787][RFC5382].

In EDM NAT devices, the same external port may be used by an outbound dynamic mapping and an inbound dynamic mapping (from the same Internal Host or from a different Internal Host). This complicates the interaction with the MAP Opcode. With such NAT devices, there are two ways envisioned to implement the MAP Opcode:

1. Have outbound mappings use a different set of External ports than inbound mappings (e.g., those created with MAP), thus reducing the interaction problem between them; or
2. On arrival of a packet (inbound from the Internet or outbound from an Internal Host), first attempt to use a dynamic outbound mapping to process that packet. If none match, attempt to use an inbound mapping to process that packet. This effectively 'prioritizes' outbound mappings above inbound mappings.

16.2. Lifetime of Explicit and Implicit Dynamic Mappings

No matter if a NAT is EIM or EDM, it is possible that one (or more) outbound mappings, using the same internal port on the Internal Host, might be created before or after a MAP request. When this occurs, it is important that the NAT honor the Lifetime returned in the MAP response. Specifically, if a mapping was created with the MAP Opcode, the implementation needs to ensure that termination of an outbound mapping (e.g., via a TCP FIN handshake) does not prematurely destroy the MAP-created inbound mapping.

16.3. PCP Failure Recovery

If an event occurs that causes the PCP server to lose dynamic mapping state (such as a crash or power outage), the mappings created by PCP are lost. Occasional loss of state may be unavoidable in a residential NAT device which does not write transient information to non-volatile memory. Loss of state is expected to be rare in a service provider environment (due to redundant power, disk drives for storage, etc.). Of course, due to outright failure of service provider equipment (e.g., software malfunction), state may still be lost.

The Epoch Time allows a client to deduce when a PCP server may have lost its state. When the Epoch Time value is observed to be outside the expected range, the PCP client can attempt to recreate the mappings following the procedures described in this section.

Further analysis of PCP failure scenarios is in [I-D.boucadair-pcp-failure].

16.3.1. Recreating Mappings

A mapping renewal packet is formatted identically to an original mapping request; from the point of view of the client it is a renewal of an existing mapping, but from the point of view of a newly rebooted PCP server it appears as a new mapping request. In the normal process of routinely renewing its mappings before they expire, a PCP client will automatically recreate all its lost mappings.

When the PCP server loses state and begins processing new PCP messages, its Epoch time is reset and begins counting again. As the result of receiving a packet where the Epoch time field is outside the expected range (Section 8.5), indicating that a reboot or similar loss of state has occurred, the client can renew its port mappings sooner, without waiting for the normal routine renewal time.

16.3.2. Maintaining Mappings

A PCP client refreshes a mapping by sending a new PCP request containing information from the earlier PCP response. The PCP server will respond indicating the new lifetime. It is possible, due to reconfiguration or failure of the PCP server, that the External IP Address and/or External Port, or the PCP server itself, has changed (due to a new route to a different PCP server). Such events are rare, but not an error. The PCP server will simply return a new External Address and/or External Port to the client, and the client should record this new External Address and Port with its rendezvous service. To detect such events more quickly, a server that requires extremely high availability may find it beneficial to use shorter lifetimes in its PCP mappings requests, so that it communicates with the PCP server more often. This is an engineering trade-off based on (i) the acceptable downtime for the service in question, (ii) the expected likelihood of NAT or firewall state loss, and (iii) the amount of PCP maintenance traffic that is acceptable.

If the PCP client has several mappings, the Epoch Time value only needs to be retrieved for one of them to determine whether or not it appears the PCP server may have suffered a catastrophic loss of state. If the client wishes to check the PCP server's Epoch Time, it sends a PCP request for any one of the client's mappings. This will return the current Epoch Time value. In that request the PCP client could extend the mapping lifetime (by asking for more time) or maintain the current lifetime (by asking for the same number of seconds that it knows are remaining of the lifetime).

If a PCP client changes its Internal IP Address (e.g., because the Internal Host has moved to a new network), and the PCP client wishes to still receive incoming traffic, it needs create new mappings on that new network. New mappings will typically also require an update to the application-specific rendezvous server if the External Address or Port are different from the previous values (see Section 10.1 and Section 11.5).

16.3.3. SCTP

Although SCTP has port numbers like TCP and UDP, SCTP works differently when behind an address-sharing NAT, in that SCTP port numbers are not changed [I-D.ietf-behave-sctpnat]. Outbound dynamic SCTP mappings use the verification tag of the association instead of the local and remote peer port numbers. As with TCP, explicit outbound mappings can be made to reduce keepalive intervals, and explicit inbound mappings can be made by passive listeners expecting to receive new associations at the external port.

Because an SCTP-aware NAT does not (currently) rewrite SCTP port numbers, it will not be able to assign an External Port that is different from the client's Internal Port. A PCP client making a MAP request for SCTP should be aware of this restriction. The PCP client SHOULD make its SCTP MAP request just as it would for a TCP MAP request: in its initial PCP MAP request it SHOULD specify zero for the External Address and Port, and then in subsequent renewals it SHOULD echo the assigned External Address and Port. However, since a current SCTP-aware NAT can only assign an External Port that is the same as the Internal Port, it may not be able to do that if the External Port is already assigned to a different PCP client. This is likely if there is more than one instance of a given SCTP service on the local network, since both instances are likely to listen on the same well-known SCTP port for that service on their respective hosts, but they can't both have the same External Port on the NAT gateway's External Address. A particular External Port may not be assignable for other reasons, such as when it is already in use by the NAT device itself, or otherwise prohibited by policy, as described in Section 11.3. In the event that the External Port matching the Internal Port cannot be assigned (and the SCTP-aware NAT does not perform SCTP port rewriting) then the SCTP-aware NAT MUST return a CANNOT_PROVIDE_EXTERNAL error to the requesting PCP client. Note that this restriction places extra burden on the SCTP server whose MAP request failed, because it then has to tear down its exiting listening socket and try again with a different Internal Port, repeatedly until it is successful in finding an External Port it can use.

The SCTP complications described above occur because of address sharing. The SCTP complications are avoided when address sharing is avoided (e.g., 1:1 NAT, firewall).

16.4. Source Address Replicated in PCP Header

All PCP requests include the PCP client's IP address replicated in the PCP header. This is used to detect address rewriting (NAT) between the PCP client and its PCP server. On operating systems that support the sockets API, the following steps are RECOMMENDED for a PCP client to insert the correct source address and port in the PCP header:

1. Create a UDP socket.
2. Call "connect" on this UDP socket using the address and port of the desired PCP server.
3. Call the getsockname() function to retrieve a sockaddr containing the source address the kernel will use for UDP packets sent through this socket.

4. If the IP address is an IPv4 address, encode the address into an IPv4-mapped IPv6 address. Place the native IPv6 address or IPv4-mapped IPv6 address into the PCP Client's IP Address field in the PCP header.
5. Send PCP requests using this connected UDP socket.

16.5. State Diagram

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

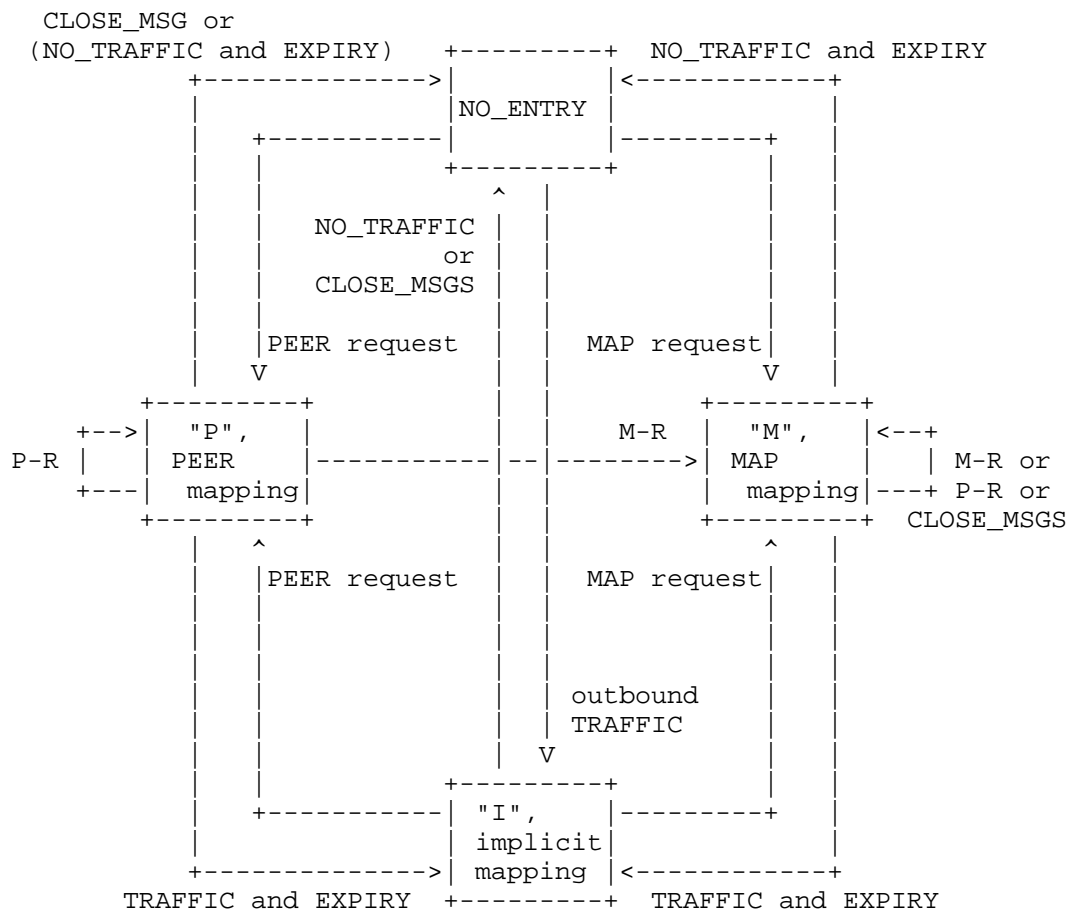


Figure 16: PCP State Diagram

The meanings of the states and events are:

NO_ENTRY: Invalid state represents Entry does not exist. This is the only possible start state.

M-R: MAP request

P-R: PEER request

M: Mapping entry when created by MAP request

P: Mapping entry when created/managed by PEER request

I: Implicit mapping created by an outgoing packet from the client (e.g., TCP SYN), and also the state when a PCP-created mapping's lifetime expires while there is still active traffic.

EXPIRY: PEER or MAP lifetime expired

TRAFFIC: Traffic seen by PCP-controlled device using this entry within the expiry time for that entry. This traffic may be inbound or outbound.

NO_TRAFFIC: Indicates that there is no TRAFFIC.

CLOSE_MSG: Protocol messages from the client or server to close the session (e.g., TCP FIN or TCP RST), as per the NAT or firewall device's handling of such protocol messages.

Notes on the diagram:

1. The 'and' clause indicates the events on either side of 'and' are required for the state-transition. The 'or' clause indicates either one of the events are enough for the state-transition.
2. Transition from state M to state I is implementation dependent.

17. Deployment Considerations

17.1. Ingress Filtering

As with implicit dynamic mappings created by outgoing TCP SYN packets, explicit dynamic mappings created via PCP use the source IP address of the packet as the Internal Address for the mappings. Therefore ingress filtering [RFC2827] SHOULD be used on the path between the Internal Host and the PCP Server to prevent the injection

of spoofed packets onto that path.

17.2. Mapping Quota

On PCP-controlled devices that create state when a mapping is created (e.g., NAT), the PCP server SHOULD maintain per-host and/or per-subscriber quotas for mappings. It is implementation-specific whether the PCP server uses a separate quotas for implicit, explicit, and static mappings, a combined quota for all of them, or some other policy.

18. Security Considerations

The goal of the PCP protocol is to improve the ability of end nodes to control their associated NAT state, and to improve the efficiency and error handling of NAT mappings when compared to existing implicit mapping mechanisms in NAT boxes and stateful firewalls. It is the security goal of the PCP protocol to limit any new denial of service opportunities, and to avoid introducing new attacks that can result in unauthorized changes to mapping state. One of the most serious consequences of unauthorized changes in mapping state is traffic theft. All mappings that could be created by a specific host using implicit mapping mechanisms are inherently considered to be authorized. Confidentiality of mappings is not a requirement, even in cases where the PCP messages may transit paths that would not be travelled by the mapped traffic.

18.1. Simple Threat Model

PCP is secure against off-path attackers who cannot spoof a packet that the PCP Server will view as a packet received from the internal network. PCP is secure against off-path attackers who can spoof the PCP server's IP address.

Defending against attackers who can modify or drop packets between the internal network and the PCP server, or who can inject spoofed packets that appear to come from the internal network is out of scope. Such an attacker can re-direct traffic to a host of their choosing.

A PCP Server is secure under this threat model if the PCP Server is constrained so that it does not configure any explicit mapping that it would not configure implicitly. In most cases, this means that PCP Servers running on NAT boxes or stateful firewalls that support the PEER and MAP Opcodes can be secure under this threat model if (1) all of their hosts are within a single administrative domain (or if the internal hosts can be securely partitioned into separate

administrative domains, as in the DS-Lite B4 case), (2) explicit mappings are created with the same lifetime as implicit mappings, and (3) the THIRD_PARTY option is not supported. PCP Servers can also securely support the MAP Opcode under this threat model if the security policy on the device running the PCP Server would permit endpoint independent filtering of implicit mappings.

PCP Servers that comply with the Simple Threat Model and do not implement a PCP security mechanism described in Section 18.2 MUST enforce the constraints described in the paragraph above.

18.1.1. Attacks Considered

- o 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.
- o 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.
- o 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.
- o 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.
- o If the MAP Opcode is supported by the PCP server in cases where the security policy would not support endpoint independent filtering of implicit mappings, then the MAP Opcode changes the security properties of the device running the PCP Server by allowing explicit mappings that violate the security policy.

18.1.2. Deployment Examples Supporting the Simple Threat Model

This section offers two examples of how the Simple Threat Model can be supported in real-world deployment scenarios.

18.1.2.1. Residential Gateway Deployment

Parity with many currently-deployed residential gateways can be achieved using a PCP Server that is constrained as described in Section 18.1 above.

18.2. Advanced Threat Model

In the Advanced Threat Model the PCP protocol ensures that attackers (on- or off-path) cannot create unauthorized mappings or make unauthorized changes to existing mappings. The protocol must also limit the opportunity for on- or off-path attackers to perpetrate denial of service attacks.

The Advanced Threat Model security model will be needed in the following cases:

- o Security infrastructure equipment, such as corporate firewalls, that does not create implicit mappings.
- o Equipment (such as CGNs or service provider firewalls) that serve multiple administrative domains and do not have a mechanism to securely partition traffic from those domains.
- o Any implementation that wants to be more permissive in authorizing explicit mappings than it is in authorizing implicit mappings.
- o Implementations that wish to support any deployment scenario that does not meet the constraints described in Section 18.1.

To protect against attacks under this threat model, a PCP security mechanism that provides an authenticated, integrity-protected signaling channel would need to be specified.

PCP Servers that implement a PCP security mechanism MAY accept unauthenticated requests. PCP Servers implementing the PCP security mechanism MUST enforce the constraints described in Section 18.1 above, in their default configuration, when processing unauthenticated requests.

18.3. Residual Threats

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

18.3.1. Denial of Service

Because of the state created in a NAT or firewall, a per-host and/or per-subscriber quota will likely exist for both implicit dynamic mappings and explicit dynamic mappings. A host might make an excessive number of implicit or explicit dynamic mappings, consuming an inordinate number of ports, causing a denial of service to other hosts. Thus, Section 17.2 recommends that hosts be limited to a reasonable number of explicit dynamic mappings.

An attacker, on the path between the PCP client and PCP server, can drop PCP requests, drop PCP responses, or spoof a PCP error, all of which will effectively deny service. Through such actions, the PCP client might not be aware the PCP server might have actually processed the PCP request. An attacker sending a NO_RESOURCES error can cause the PCP client to not send messages to that server for a while. There is no mitigation to this on-path attacker.

18.3.2. Ingress Filtering

It is important to prevent a host from fraudulently creating, deleting, or refreshing a mapping (or filtering) for another host, because this can expose the other host to unwanted traffic, prevent it from receiving wanted traffic, or consume the other host's mapping quota. Both implicit and explicit dynamic mappings are created based on the source IP address in the packet, and hence depend on ingress filtering to guard against spoof source IP addresses.

18.3.3. Mapping Theft

In the time between when a PCP server loses state and the PCP client notices the lower-than-expected Epoch Time value, it is possible that the PCP client's mapping will be acquired by another host (via an explicit dynamic mapping or implicit dynamic mapping). This means incoming traffic will be sent to a different host ("theft"). Rapid Recovery reduces this interval, but would not completely eliminate this threat. The PCP client can reduce this interval by using a relatively short lifetime; however, this increases the amount of PCP chatter. This threat is reduced by using persistent storage of explicit dynamic mappings in the PCP server (so it does not lose explicit dynamic mapping state), or by ensuring the previous external IP address, protocol, and port cannot be used by another host (e.g.,

by using a different IP address pool).

18.3.4. Attacks Against Server Discovery

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

19. IANA Considerations

IANA is requested to perform the following actions:

19.1. Port Number

PCP will use ports 5350 and 5351 (currently assigned by IANA to NAT-PMP [I-D.cheshire-nat-pmp]). We request that IANA re-assign those ports to PCP, and relinquish UDP port 44323.

[Note to RFC Editor: Please remove the text about relinquishing port 44323 prior to publication.]

19.2. Opcodes

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

value	Opcode
-----	-----
0	ANNOUNCE
1	MAP
2	PEER
3-31	Standards Action [RFC5226]
32-63	Specification Required [RFC5226]
96-126	Private Use [RFC5226]
127	Reserved, Standards Action [RFC5226]

The value 127 is Reserved and may be assigned via Standards Action [RFC5226]. The values in the range 3-31 can be assigned via Standards Action [RFC5226], 32-63 via Specification Required [RFC5226], and 96-126 is for Private Use [RFC5226].

19.3. Result Codes

IANA shall create a new registry for PCP result codes, numbered 0-255, initially populated with the result codes from Section 7.4. The value 255 is Reserved and may be assigned via Standards Action [RFC5226].

The values in the range 14-127 can be assigned via Standards Action [RFC5226], 128-191 via Specification Required [RFC5226], and 191-254 is for Private Use [RFC5226].

19.4. Options

IANA shall create a new registry for PCP Options, numbered 0-255, each with an associated mnemonic. The values 0-127 are mandatory-to-process, and 128-255 are optional to process. The initial registry contains the Options described in Section 13. The Option values 0, 127 and 255 are Reserved and may be assigned via Standards Action [RFC5226].

Additional PCP Option codes in the ranges 4-63 and 128-191 can be created via Standards Action [RFC5226], the ranges 64-95 and 192-223 are for Specification Required [RFC5226] and the ranges 96-126 and 224-254 are for Private Use [RFC5226].

Documents describing an Option should describe if the processing for both the PCP client and server and the information below:

Option Name: <mnemonic>
Number: <value>
Purpose: <textual description>
Valid for Opcodes: <list of Opcodes>
Length: <rules for length>
May appear in: <requests/responses/both>
Maximum occurrences: <count>

20. Acknowledgments

Thanks to Xiaohong Deng, Alain Durand, Christian Jacquenet, Jacni Qin, Simon Perreault, James Yu, Tina TSOU (Ting ZOU), Felipe Miranda Costa, James Woodyatt, Dave Thaler, Masataka Ohta, Vijay K. Gurbani, Loa Andersson, Richard Barnes, Russ Housley, Adrian Farrel, Pete Resnick, Pasi Sarolahti, Robert Sparks, Wesley Eddy, Dan Harkins, Peter Saint-Andre, Stephen Farrell, Ralph Droms, Felipe Miranda Costa, Amit Jain, and Wim Henderickx for their comments and review.

Thanks to Simon Perreault for highlighting the interaction of dynamic connections with PCP-created mappings.

Thanks to Francis Dupont for his several thorough reviews of the specification, which improved the protocol significantly.

Thanks to T. S. Ranganathan for the state diagram.

Thanks to Peter Lothberg for clock skew information.

Thanks to Margaret Wasserman and Sam Hartman for writing the Security Considerations section.

Thanks to authors of DHCPv6 for retransmission text.

21. References

21.1. Normative References

- [RFC0768] Postel, J., "User Datagram Protocol", STD 6, RFC 768, August 1980.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2827] Ferguson, P. and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing", BCP 38, RFC 2827, May 2000.
- [RFC4086] Eastlake, D., Schiller, J., and S. Crocker, "Randomness Requirements for Security", BCP 106, RFC 4086, June 2005.
- [RFC4193] Hinden, R. and B. Haberman, "Unique Local IPv6 Unicast Addresses", RFC 4193, October 2005.
- [RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, February 2006.
- [RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.
- [RFC6056] Larsen, M. and F. Gont, "Recommendations for Transport-Protocol Port Randomization", BCP 156, RFC 6056, January 2011.
- [proto_numbers] IANA, "Protocol Numbers", 2011, <<http://www.iana.org/assignments/protocol-numbers/protocol-numbers.xml>>.

21.2. Informative References

- [I-D.boucadair-pcp-failure] Boucadair, M., Dupont, F., and R. Penno, "Port Control Protocol (PCP) Failure Scenarios", draft-boucadair-pcp-failure-04 (work in progress), August 2012.

- [I-D.cheshire-dnsext-dns-sd]
Cheshire, S. and M. Krochmal, "DNS-Based Service Discovery", draft-cheshire-dnsext-dns-sd-11 (work in progress), December 2011.
- [I-D.cheshire-nat-pmp]
Cheshire, S. and M. Krochmal, "NAT Port Mapping Protocol (NAT-PMP)", draft-cheshire-nat-pmp-05 (work in progress), September 2012.
- [I-D.ietf-behave-lsn-requirements]
Perreault, S., Yamagata, I., Miyakawa, S., Nakagawa, A., and H. Ashida, "Common requirements for Carrier Grade NATs (CGNs)", draft-ietf-behave-lsn-requirements-09 (work in progress), August 2012.
- [I-D.ietf-behave-sctpnaat]
Stewart, R., Tuexen, M., and I. Ruengeler, "Stream Control Transmission Protocol (SCTP) Network Address Translation", draft-ietf-behave-sctpnaat-07 (work in progress), October 2012.
- [I-D.ietf-pcp-upnp-igd-interworking]
Boucadair, M., Dupont, F., Penno, R., and D. Wing, "Universal Plug and Play (UPnP) Internet Gateway Device (IGD)-Port Control Protocol (PCP) Interworking Function", draft-ietf-pcp-upnp-igd-interworking-04 (work in progress), September 2012.
- [I-D.miles-behave-l2nat]
Miles, D. and M. Townsley, "Layer2-Aware NAT", draft-miles-behave-l2nat-00 (work in progress), March 2009.
- [IGDv1]
UPnP Gateway Committee, "WANIPConnection:1", November 2001, <<http://upnp.org/specs/gw/UPnP-gw-WANIPConnection-v1-Service.pdf>>.
- [RFC0793]
Postel, J., "Transmission Control Protocol", STD 7, RFC 793, September 1981.
- [RFC1918]
Rekhter, Y., Moskowitz, R., Karrenberg, D., Groot, G., and E. Lear, "Address Allocation for Private Internets", BCP 5, RFC 1918, February 1996.
- [RFC2136]
Vixie, P., Thomson, S., Rekhter, Y., and J. Bound, "Dynamic Updates in the Domain Name System (DNS UPDATE)", RFC 2136, April 1997.

- [RFC3007] Wellington, B., "Secure Domain Name System (DNS) Dynamic Update", RFC 3007, November 2000.
- [RFC3022] Srisuresh, P. and K. Egevang, "Traditional IP Network Address Translator (Traditional NAT)", RFC 3022, January 2001.
- [RFC3581] Rosenberg, J. and H. Schulzrinne, "An Extension to the Session Initiation Protocol (SIP) for Symmetric Response Routing", RFC 3581, August 2003.
- [RFC3587] Hinden, R., Deering, S., and E. Nordmark, "IPv6 Global Unicast Address Format", RFC 3587, August 2003.
- [RFC4303] Kent, S., "IP Encapsulating Security Payload (ESP)", RFC 4303, December 2005.
- [RFC4340] Kohler, E., Handley, M., and S. Floyd, "Datagram Congestion Control Protocol (DCCP)", RFC 4340, March 2006.
- [RFC4787] Audet, F. and C. Jennings, "Network Address Translation (NAT) Behavioral Requirements for Unicast UDP", BCP 127, RFC 4787, January 2007.
- [RFC4941] Narten, T., Draves, R., and S. Krishnan, "Privacy Extensions for Stateless Address Autoconfiguration in IPv6", RFC 4941, September 2007.
- [RFC4960] Stewart, R., "Stream Control Transmission Protocol", RFC 4960, September 2007.
- [RFC4961] Wing, D., "Symmetric RTP / RTP Control Protocol (RTCP)", BCP 131, RFC 4961, July 2007.
- [RFC5382] Guha, S., Biswas, K., Ford, B., Sivakumar, S., and P. Srisuresh, "NAT Behavioral Requirements for TCP", BCP 142, RFC 5382, October 2008.
- [RFC6092] Woodyatt, J., "Recommended Simple Security Capabilities in Customer Premises Equipment (CPE) for Providing Residential IPv6 Internet Service", RFC 6092, January 2011.
- [RFC6145] Li, X., Bao, C., and F. Baker, "IP/ICMP Translation Algorithm", RFC 6145, April 2011.
- [RFC6146] Bagnulo, M., Matthews, P., and I. van Beijnum, "Stateful NAT64: Network Address and Protocol Translation from IPv6

Clients to IPv4 Servers", RFC 6146, April 2011.

- [RFC6296] Wasserman, M. and F. Baker, "IPv6-to-IPv6 Network Prefix Translation", RFC 6296, June 2011.
- [RFC6333] Durand, A., Droms, R., Woodyatt, J., and Y. Lee, "Dual-Stack Lite Broadband Deployments Following IPv4 Exhaustion", RFC 6333, August 2011.
- [RFC6619] Arkko, J., Eggert, L., and M. Townsley, "Scalable Operation of Address Translators with Per-Interface Bindings", RFC 6619, June 2012.

Appendix A. NAT-PMP Transition

The Port Control Protocol (PCP) is a successor to the NAT Port Mapping Protocol, NAT-PMP [I-D.cheshire-nat-pmp], and shares similar semantics, concepts, and packet formats. Because of this NAT-PMP and PCP both use the same port, and use NAT-PMP and PCP's version negotiation capabilities to determine which version to use. This section describes how an orderly transition may be achieved.

A client supporting both NAT-PMP and PCP SHOULD send its request using the PCP packet format. This will be received by a NAT-PMP server or a PCP server. If received by a NAT-PMP server, the response will be as indicated by the NAT-PMP specification [I-D.cheshire-nat-pmp], which will cause the client to downgrade to NAT-PMP and re-send its request in NAT-PMP format. If received by a PCP server, the response will be as described by this document and processing continues as expected.

A PCP server supporting both NAT-PMP and PCP can handle requests in either format. The first octet of the packet indicates if it is NAT-PMP (first octet zero) or PCP (first octet non-zero).

A PCP-only gateway receiving a NAT-PMP request (identified by the first octet being zero) will interpret the request as a version mismatch. Normal PCP processing will emit a PCP response that is compatible with NAT-PMP, without any special handling by the PCP server.

Appendix B. Change History

[Note to RFC Editor: Please remove this section prior to publication.]

B.1. Changes from draft-ietf-pcp-base-28 to -29

- o Removed text suggesting PCP client can remove old mappings when it acquires a new IP address.

B.2. Changes from draft-ietf-pcp-base-27 to -28

- o When processing MAP request or processing PEER request, Mapping Nonce validation only applies to Basic Threat Model, and not to THIRD_PARTY.
- o A maximum payload size of 1100 keeps PCP packets below IPv6's 1280 MTU limit while still allowing some room for encapsulation. This accommodates EAP over PANA over PCP (EAP needs 1020 octets, per RFC3748), should PCP authentication decide to use EAP over PANA over PCP.
- o Both MAP and PEER-created mappings cannot have their lifetimes reduced beyond normal UDP/TCP timeouts.
- o Disallow re-assigning External Port to same internal host.

B.3. Changes from draft-ietf-pcp-base-26 to -27

- o For table, reverted the NAT64 remote peer to IPv6 -- because from the IPv6 PCP client's perspective, the remote peer really is IPv6.
- o "list of PCP server addresses" changed to "longer list of PCP server addresses"
- o Clarify that unsolicited ANNOUNCE messages are sent from the PCP server IP address and PCP port.
- o "1024 bytes" changed to "1024 octets".
- o Clarify that re-transmitted requests must use same Mapping Nonce value (beginning of Section 8.1.1).
- o Describe that de-synchronization that can occur (end of Section 8.1.1).
- o For devices that lose state or expect IP renumbering, Rapid Recovery is now a MUST, with SHOULD for implementing both multicast Announce mechanism and unicast mechanisms.
- o For refreshing MAP or PEER, Mapping Nonce has to match the previous MAP or PEER. This protects from off-path attackers stealing MAP or shortening PEER mappings.

- o With the Mapping Nonce change, we now allow PEER to reduce mapping lifetime to same lifetime as implicit mapping lifetime (but not shorter). Changes for this are in both PEER section and Security Considerations.
- o With Mapping Nonce change, can no longer delete a 'set of mappings' (because we cannot send multiple Mapping Nonce values), so removed text that allowed that.
- o Send Mapping Update only 3 times (used to be 10 times).
- o General PCP processing now requires validating Mapping Nonce, if the opcode uses a Mapping Nonce Section 8.3.
- o Moved text describing NO_RESOURCES handling from General Processing section to MAP and PEER processing sections, as it NO_RESOURCES processing should be done after validating Mapping Nonce.
- o Clarified SCTP NAT behavior (port numbers stay the same, causing grief).
- o added EIM definition.
- o Clarified Mapping Type definitions.
- o PCP Client definition simplified to no longer obliquely and erroneously reference UPnP IGD.
- o Clarified using network-byte order.
- o Epoch time comparison now allows slight packet re-ordering.
- o Encourage that when new address is assigned (e.g., DHCP) that PCP as well as non-PCP mappings be cleaned up.
- o Simplified formatting of retransmission, but no normative change.
- o Clarified how server chooses ports and how Suggested External Port can gently influence that decision.
- o Described how PCP client can use PCP Client Address with a non-PCP-aware inner NAT (Section 8.1.)
- o Clarified 1024 octet length applies to UDP payload itself, and that error responses copy 1024 of UDP payload.

- o Lifetime for both MAP and PEER should not exceed the remaining IP address lifetime of the PCP client (if known) or half the typical IP address lifetime (if the remaining lifetime is unknown).
 - o Lifetime section was (mistakenly) a subsection of the MAP section, but referenced by both MAP and PEER. It is now a top-level section.
 - o Clarified that PEER cannot reduce lifetime beyond normal implicit mapping lifetime, no matter what. This restriction prevents malicious or accidental deletion of a quiescent connection that was not using PCP.
 - o Clarified port re-use of PCP-created mappings should follow same port re-use algorithm used by the NAT for implicitly-created mappings (likely maximum segment lifetime).
 - o Other minor text changes; consult diffs.
- B.4. Changes from draft-ietf-pcp-base-25 to -26
- o Changed "internal address and port" to "internal address, protocol, and port" in several more places.
 - o Improved wording of THIRD_PARTY restrictions.
 - o Bump version number from 1 to 2, to accommodate pre-RFC PCP client implementations without needing a heuristic.
- B.5. Changes from draft-ietf-pcp-base-24 to -25
- o Clarified the port used by the PCP server when sending unsolicited unicast ANNOUNCE.
 - o Removed parenthetical comment implying ANNOUNCE was not a normal Opcode; it is a normal Opcode.
 - o Explain that non-PCP-speaking host-based and network-based firewalls need to allow incoming connections for MAP to work.
 - o For race condition with PREFER_FAILURE, clarified that it is the PCP client's responsibility to delete the mapping if the PCP client doesn't need the mapping.
 - o For table, the NAT64 remote peer is IPv4 (was IPv6).
 - o Added a Mapping Nonce field to both MAP and PEER requests and responses, to protect from off-path attackers spoofing the PCP

server's IP address.

- o Security considerations: added 'PCP is secure against off-path attackers who can spoof the PCP server's IP address', because of the addition of the Mapping Nonce.
- o Removed reference to DS-Lite from Security Considerations, as part of the changes to THIRD_PARTY from IESG review.
- o Rapid Recovery is now a SHOULD implement.
- o Clarify behavior of PREFER_FAILURE with zeros in Suggested External Port or Address fields.
- o PCP server is now more robust and insistent about informing PCP client of state changes.
- o When PCP server sends Mapping Update to a specific PCP client, and gets an update for a particular mapping, it doesn't need to send reminders about that mapping any more.
- o THIRD_PARTY is now prohibited on subscriber PCP clients.

B.6. Changes from draft-ietf-pcp-base-23 to -24

- o Explained common questions regarding PCP's design, such as lack of transaction identifiers and its request/response semantics and operation (Protocol Design Note (Section 6)).
- o added MUST for all-zeros IPv6 and IPv4 address formats.
- o included field definitions for Opcode-specific information and PCP options under both Figure 2 and Figure 3.
- o adopted retransmission mechanism from DHCPv6.
- o 1024 message size limit described in PCP message restriction.
- o Explained PCP server list, with example of host with IPv4 and IPv6 addresses having two PCP servers (one IPv4 PCP server for IPv4 mappings and one IPv6 PCP server for IPv6 mappings).
- o mention PCP client needs to expect unsolicited PCP responses from previous incarnations of itself (on the same host) or of this host (using same IP address as another PCP client).
- o eliminated overuse of 'packet format' when it was 'opcode format'.

- o for IANA registries, added code points assignable via Standards Action (previously was just Specification Required).
- o Version negotiation, added explanation that retrying after 30 minutes makes the protocol self-healing if the PCP server is upgraded.
- o Version negotiation now accomodates non-contiguous version numbers.
- o Tweaked definition of VERSION field (that "1" is for this version, but other values could of course appear in the future).
- o when receiving unsolicited ANNOUNCE, PCP client now waits random 0-5 seconds.
- o Removed 'interworking function' from list of terminology because we no longer use the term in this document.
- o tightened definitions of 'PCP client' and 'PCP server'.
- o For 'Requested Lifetime' definitions, removed text requiring its value be 0 for not-yet-defined opcodes.
- o Removed some unnecessary text suggesting logging (is an implementation detail).
- o Added active-mode FTP as example protocol that can break with mappings to different IP addresses.
- o Clarified that if PCP request contains a Suggested External Address, the PCP server should try to create a mapping to that address even if other mappings already exist to a different external address.
- o Changed "internal address and port" to "internal address, protocol, and port" in several places.
- o Clarified which 96 bits are copied into error response. Clarified that only error responses are copied verbatim from request.
- o a single PCP server can control multiple NATs or multiple firewalls (Section 4).
- o Clarified that sending unsolicited multicast ANNOUNCE is not always available on all networks.

- o Clarified option length error example is when option length exceeds UDP length
- o Explained that an on-path attacker that can spoof packets can re-direct traffic to a host of their choosing.
- o Instead of saying IPv4-mapped addresses won't appear on the wire, say they aren't used for mappings.
- o THIRD_PARTY is useful for management device (e.g., in a network operations center).
- o Clarified PCP responses have fields updated as indicated with 'set by the server' from field definitions.
- o Disallow using MAP to the PCP ports themselves and encourage implementations have policy control for other ports.
- o Instead of 'idempotent', now says 'identical requests generate identical response'.
- o Described which Options are included when sending Mapping Update (unsolicited responses), Section 14.2.
- o Dropped [RFC2136] and [RFC3007] to informative references.
- o Updated from 'should' to 'SHOULD' in Section 17.1.
- o Described 'hairpin' in terminology section.

B.7. Changes from draft-ietf-pcp-base-22 to -23

- o Instead of returning error NO_RESOURCES when requesting a MAP for all protocols or for all ports, return UNSUPP_PROTOCOL.
- o Clarify that PEER-created mappings are treated as if it was implicit dynamic outbound mapping (Section 12.3).
- o Point out that PEER-created mappings may be very short until bi-directional traffic is seen by the PCP-managed device.
- o Clarification that an existing implicit mapping (created e.g., by TCP SYN) can become managed by a MAP request (Section 11.3).
- o Clarified the ANNOUNCE Opcode is being defined in Section 14.1, and that the length of requests (as well as responses) is zero.

- o Clarify that ANNOUNCE has Lifetime=0 for requests and responses.
- o Clarify ANNOUNCE can be sent unicast by the client (to solicit a response), or can be multicasted (unsolicited) by the server.
- o Allow ANNOUNCE to be sent unicast by the server, to accomodate case where PCP server fails but knows the IP address of a PCP client (e.g., web portal).
- o Clarified ports used for unicast and multicast unsolicited ANNOUNCE.
- o Tweaked NO_RESOURCES handling, to just disallow *new* mappings.
- o State diagram is now non-normative, because it overly simplifies that implicit mappings become MAP (when they actually still retain their previous behavior when the MAP expires).
- o In section Section 15, clarified that PEER cannot delete or shorten any lifetime, and that MAP can only shorten or delete lifetimes of MAP-created mappings.
- o Clarified handling of MAP when mapping already exists (4 steps).
- o $2^{32}-1$
- o Randomize retry interval (1.5-2.5), and maximum retry interval is now 1024 seconds (was 15 minutes).
- o Remove MUST be 0 for Reserved field when sending error responses for un-parseable message.
- o Whenever PCP client includes Suggested IP Address (in MAP or PEER), the PCP server should try to fulfill that request, even if creating a mapping on that IP address means the internal host will have mappings on different IP addresses and ports.
- o For NO_RESOURCES error, the PCP client can attempt to renew and attempt to delete mappings (as they can help shed load) -- it just can't try to create new ones.
- o Removed the overly simplistic normative text regarding honoring Suggested External Address from Section 10 in favor of the text in Section 11.3 which has significantly more detail.

B.8. Changes from draft-ietf-pcp-base-21 to -22

- o Removed paragraph discussing multiple addresses on the same (physical) interface; those will work with PCP.
- o The FILTER Option's Prefix Length field redefined to simply be a count of the relevant bits (rather than 0-32 for IPv4-mapped addresses).
- o Point out NO_RESOURCES attack vector in security considerations.
- o Tighten up recommendation for client handling long Lifetimes, and moved from the MAP-specific section to the General PCP Processing section. Client should normalize to 24 hours maximum for success and 30 minute maximum for errors.

B.9. Changes from draft-ietf-pcp-base-20 to -21

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

B.10. Changes from draft-ietf-pcp-base-19 to -20

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

B.11. Changes from draft-ietf-pcp-base-18 to -19

- o Described race condition with MAP containing PREFER_FAILURE and Mapping Update.
- o Added state machine (Section 16.5).

- o Fully integrated Rapid Recovery, with a separate Opcode having its own processing description.
- o Clarified that due to Mapping Update, a single MAP or PEER request can receive multiple responses, each updating the previous request, and that the PCP client needs to handle MAP updates or PEER updates accordingly.

B.12. Changes from draft-ietf-pcp-base-17 to -18

- o Removed UNPROCESSED option. Instead, unprocessed options are simply not included in responses.
- o Updated terminology section for Implicit/Explicit and Outbound/Inbound.
- o PEER requests cannot delete or shorten the lifetime of a mapping.
- o Clarified that PCP clients only retransmit mapping requests for as long as they actually want the mapping.
- o Revised Epoch time calculations and explanation.
- o Renamed the announcement opcode from No-Op to ANNOUNCE.

B.13. Changes from draft-ietf-pcp-base-16 to -17

- o suggest acquiring a mapping to the Discard port if there is a desire to show the user their external address (Section 11.6).
- o Added Restart Announcement.
- o Tweaked terminology.
- o Detailed how error responses are generated.

B.14. Changes from draft-ietf-pcp-base-15 to -16

- o fixed mistake in PCP request format (had 32 bits of extraneous fields)
- o Allow MAP to request all ports (port=0) for a specific protocol (protocol!=0), for the same reason we added support for all ports (port=0) and all protocols (protocol=0) in -15
- o corrected text on Client Processing a Response related to receiving ADDRESS_MISMATCH error.

- o updated Epoch text.
- o Added text that MALFORMED_REQUEST is generated for MAP if Protocol is zero but Internal Port is non-zero.

B.15. Changes from draft-ietf-pcp-base-14 to -15

- o Softened and removed text that was normatively explaining how PEER is implemented within a NAT.
- o Allow a MAP request for protocol=0, which means "all protocols". This can work for an IPv6 or IPv4 firewall. Its use with a NAPT is undefined.
- o combined SERVER_OVERLOADED and NO_RESOURCES into one error code, NO_RESOURCES.
- o SCTP mappings have to use same internal and suggested external ports, and have implied PREFER_FAILURE semantics.
- o Re-instated ADDRESS_MISMATCH error, which only checks the client address (not its port).

B.16. Changes from draft-ietf-pcp-base-13 to -14

- o Moved discussion of socket operations for PCP source address into Implementation Considerations section.
- o Integrated numerous WGLC comments.
- o NPTv6 in scope.
- o Re-written security considerations section. Thanks, Margaret!
- o Reduced PEER4 and PEER6 Opcodes to just a single Opcode, PEER.
- o Reduced MAP4 and MAP6 Opcodes to just a single Opcode, MAP.
- o Rearranged the PEER packet formats to align with MAP.
- o Removed discussion of the "O" bit for Options, which was confusing. Now the text just discusses the most significant bit of the Option code which indicates mandatory/optional, so it is clearer the field is 8 bits.
- o The THIRD_PARTY Option from an unauthorized host generates UNSUPP_OPTION, so the PCP server doesn't disclose it knows how to process THIRD_PARTY Option.

- o Added table to show which fields of MAP or PEER need IPv6/IPv4 addresses for IPv4 firewall, DS-Lite, NAT64, NAT44, etc.
- o Accommodate the server's Epoch going up or down, to better detect switching to a different PCP server.
- o Removed ADDRESS_MISMATCH; the server always includes its idea of the Client's IP Address and Port, and it's up to the client to detect a mismatch (and rectify it).

B.17. Changes from draft-ietf-pcp-base-12 to -13

- o All addresses are 128 bits. IPv4 addresses are represented by IPv4-mapped IPv6 addresses (::FFFF/96)
- o PCP request header now includes PCP client's port (in addition to the client's IP address, which was in -12).
- o new ADDRESS_MISMATCH error.
- o removed PROCESSING_ERROR error, which was too similar to MALFORMED_REQUEST.
- o Tweaked text describing how PCP client deals with multiple PCP server addresses (Section 8.1)
- o clarified that when overloaded, the server can send SERVER_OVERLOADED (and drop requests) or simply drop requests.
- o Clarified how PCP client chooses MAP4 or MAP6, depending on the presence of its own IPv6 or IPv4 interfaces (Section 10).
- o compliant PCP server MUST support MAPx and PEERx, SHOULD support ability to disable support.
- o clarified that MAP-created mappings have no filtering, and PEER-created mappings have whatever filtering and mapping behavior is normal for that particular NAT / firewall.
- o Integrated WGLC feedback (small changes to abstract, definitions, and small edits throughout the document)
- o allow new Options to be defined with a specification (rather than standards action)

B.18. Changes from draft-ietf-pcp-base-11 to -12

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

B.19. Changes from draft-ietf-pcp-base-10 to -11

- o clarified what can cause CANNOT_PROVIDE_EXTERNAL error to be generated.

B.20. Changes from draft-ietf-pcp-base-09 to -10

- o Added External_AF field to PEER requests. Made PEER's Suggested External IP Address and Assigned External IP Address always be 128 bits long.

B.21. Changes from draft-ietf-pcp-base-08 to -09

- o Clarified in PEER Opcode introduction (Section 12) that they can also create mappings.
- o More clearly explained how PEER can re-create an implicit dynamic mapping, for purposes of rebuilding state to maintain an existing session (e.g., long-lived TCP connection to a server).
- o Added Suggested External IP Address to the PEER Opcodes, to allow more robust rebuilding of connections. Added related text to the PEER server processing section.
- o Removed text encouraging PCP server to statefully remember its mappings from Section 16.3.1, as it didn't belong there. Text in Security Considerations already encourages persistent storage.
- o More clearly discussed how PEER is used to re-establish TCP mapping state. Moved it to a new section, as well (it is now Section 10.4).
- o MAP errors now copy the Suggested Address (and port) fields to Assigned IP Address (and port), to allow PCP client to distinguish among many outstanding requests when using PREFER_FAILURE.
- o Mapping theft can also be mitigated by ensuring hosts can't re-use same IP address or port after state loss.
- o the UNPROCESSED option is renumbered to 0 (zero), which ensures no other option will be given 0 and be unable to be expressed by the UNPROCESSED option (due to its 0 padding).

- o created new Implementation Considerations section (Section 16) which discusses non-normative things that might be useful to implementers. Some new text is in here, and the Failure Scenarios text (Section 16.3) has been moved to here.
- o Tweaked wording of EDM NATs in Section 16.1 to clarify the problem occurs both inside->outside and outside->inside.
- o removed "Interference by Other Applications on Same Host" section from security considerations.
- o fixed zero/non-zero text in Section 15.
- o removed duplicate text saying MAP is allowed to delete an implicit dynamic mapping. It is still allowed to do that, but it didn't need to be said twice in the same paragraph.
- o Renamed error from UNAUTH_TARGET_ADDRESS to UNAUTH_THIRD_PARTY_INTERNAL_ADDRESS.
- o for FILTER option, removed unnecessary detail on how FILTER would be bad for PEER, as it is only allowed for MAP anyway.
- o In Security Considerations, explain that PEER can create a mapping which makes its security considerations the same as MAP.

B.22. Changes from draft-ietf-pcp-base-07 to -08

- o moved all MAP4-, MAP6-, and PEER-specific options into a single section.
- o discussed NAT port-overloading and its impact on MAP (new section Section 16.1), which allowed removing the IMPLICIT_MAPPING_EXISTS error.
- o eliminated NONEXIST_PEER error (which was returned if a PEER request was received without an implicit dynamic mapping already being created), and adjusted PEER so that it creates an implicit dynamic mapping.
- o Removed Deployment Scenarios section (which detailed NAT64, NAT44, Dual-Stack Lite, etc.).
- o Added Client's IP Address to PCP common header. This allows server to refuse a PCP request if there is a mismatch with the source IP address, such as when a non-PCP-aware NAT was on the path. This should reduce failure situations where PCP is deployed in conjunction with a non-PCP-aware NAT. This addition was

consensus at IETF80.

- o Changed UNSPECIFIED_ERROR to PROCESSING_ERROR. Clarified that MALFORMED_REQUEST is for malformed requests (and not related to failed attempts to process the request).
- o Removed MISORDERED_OPTIONS. Consensus of IETF80.
- o SERVER_OVERLOADED is now a common PCP error (instead of specific to MAP).
- o Tweaked PCP retransmit/retry algorithm again, to allow more aggressive PCP discovery if an implementation wants to do that.
- o Version negotiation text tweaked to soften NAT-PMP reference, and more clearly explain exactly what UNSUPP_VERSION should return.
- o 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.
- o New Appendix A to discuss NAT-PMP / PCP interworking.
- o improved pseudocode to be non-blocking.
- o clarified that PCP cannot delete a static mapping (i.e., a mapping created by CLI or other non-PCP means).
- o moved theft of mapping discussion from Epoch section to Security Considerations.

B.23. Changes from draft-ietf-pcp-base-06 to -07

- o tightened up THIRD_PARTY security discussion. Removed "highest numbered address", and left it as simply "the CPE's IP address".
- o removed UNABLE_TO_DELETE_ALL error.
- o renumbered Opcodes
- o renumbered some error codes
- o assigned value to IMPLICIT_MAPPING_EXISTS.
- o UNPROCESSED can include arbitrary number of option codes.
- o Moved lifetime fields into common request/response headers

- o We've noticed we're having to repeatedly explain to people that the "requested port" is merely a hint, and the NAT gateway is free to ignore it. Changed name to "suggested port" to better convey this intention.
- o Added NAT-PMP transition section
- o Separated Internal Address, External Address, Remote Peer Address definition
- o Unified Mapping, Port Mapping, Port Forwarding definition
- o adjusted so DHCP configuration is non-normative.
- o mentioned PCP refreshes need to be sent over the same interface.
- o renamed the REMOTE_PEER_FILTER option to FILTER.
- o Clarified FILTER option to allow sending an ICMP error if policy allows.
- o for MAP, clarified that if the PCP client changed its IP address and still wants to receive traffic, it needs to send a new MAP request.
- o clarified that PEER requests have to be sent from same interface as the connection itself.
- o for MAP opcode, text now requires mapping be deleted when lifetime expires (per consensus on 8-Mar interim meeting)
- o PEER Opcode: better description of remote peer's IP address, specifically that it does not control or establish any filtering, and explaining why it is 'from the PCP client's perspective'.
- o Removed latent text allowing DMZ for 'all protocols' (protocol=0). Which wouldn't have been legal, anyway, as protocol 0 is assigned by IANA to HOPOPT (thanks to James Yu for catching that one).
- o clarified that PCP server only listens on its internal interface.
- o abandoned 'target' term and reverted to simpler 'internal' term.

B.24. Changes from draft-ietf-pcp-base-05 to -06

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

- o defined THIRD_PARTY Option to work with the PEER Opcode, too. This meant moving it to its own section, and having both MAP and PEER Opcodes reference that common section.
- o used "target" instead of "internal", in the hopes that clarifies internal address used by PCP itself (for sending its packets) versus the address for Mappings.
- o Options are now required to be ordered in requests, and ordering has to be validated by the server. Intent is to ease server processing of mandatory-to-implement options.
- o Swapped Option values for the mandatory- and optional-to-process Options, so we can have a simple lowest..highest ordering.
- o added MISORDERED_OPTIONS error.
- o re-ordered some error messages to cause MALFORMED_REQUEST (which is PCP's most general error response) to be error 1, instead of buried in the middle of the error numbers.
- o clarified that, after successfully using a PCP server, that PCP server is declared to be non-responsive after 5 failed retransmissions.
- o tightened up text (which was inaccurate) about how long general PCP processing is to delay when receiving an error and if it should honor Opcode-specific error lifetime. Useful for MAP errors which have an error lifetime. (This all feels awkward to have only some errors with a lifetime.)
- o Added better discussion of multiple interfaces, including highlighting Wi-Fi+Ethernet. Added discussion of using IPv6 Privacy Addresses and RFC1918 as source addresses for PCP requests. This should finish the section on multi-interface issues.
- o added some text about why server might send SERVER_OVERLOADED, or might simply discard packets.
- o Dis-allow internal-port=0, which means we dis-allow using PCP as a DMZ-like function. Instead, ports have to be mapped individually.
- o Text describing server's processing of PEER is tightened up.
- o Server's processing of PEER now says it is implementation-specific if a PCP server continues to allow the mapping to exist after a PEER message. Client's processing of PEER says that if client

wants mapping to continue to exist, client has to continue to send recurring PEER messages.

B.25. Changes from draft-ietf-pcp-base-04 to -05

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

B.26. Changes from draft-ietf-pcp-base-03 to -04

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

- o Allow arbitrary Options to be included in PCP response, so that PCP server can indicate un-supported PCP Options. Satisfies PCP Issue #19
- o Reduced scope to only deal with mapping protocols that have port numbers.
- o Reduced scope to not support DMZ-style forwarding.
- o Clarified version negotiation.

B.27. Changes from draft-ietf-pcp-base-02 to -03

- o Adjusted abstract and introduction to make it clear PCP is intended to forward ports and intended to reduce application keepalives.
- o First bit in PCP common header is set. This allows DTLS and non-DTLS to be multiplexed on same port, should a future update to this specification add DTLS support.
- o Moved subscriber identity from common PCP section to MAP* section.
- o made clearer that PCP client can reduce mapping lifetime if it wishes.
- o Added discussion of host running a server, client, or symmetric client+server.
- o Introduced PEER4 and PEER6 Opcodes.
- o Removed REMOTE_PEER Option, as its function has been replaced by the new PEER Opcodes.
- o IANA assigned port 44323 to PCP.
- o Removed AMBIGUOUS error code, which is no longer needed.

B.28. Changes from draft-ietf-pcp-base-01 to -02

- o more error codes
- o PCP client source port number should be random
- o PCP message minimum 8 octets, maximum 1024 octets.
- o tweaked a lot of text in section 7.4, "Opcode-Specific Server Operation".

- o opening a mapping also allows ICMP messages associated with that mapping.
- o PREFER_FAILURE value changed to the mandatory-to-process range.
- o added text recommending applications that are crashing obtain short lifetimes, to avoid consuming subscriber's port quota.

B.29. Changes from draft-ietf-pcp-base-00 to -01

- o Significant document reorganization, primarily to split base PCP operation from Opcode operation.
- o packet format changed to move 'protocol' outside of PCP common header and into the MAP* opcodes
- o Renamed Informational Elements (IE) to Options.
- o Added REMOTE_PEER (for disambiguation with dynamic ports), REMOTE_PEER_FILTER (for simple packet filtering), and PREFER_FAILURE (to optimize UPnP IGDv1 interworking) options.
- o Is NAT or router behind B4 in scope?
- o PCP option MAY be included in a request, in which case it MUST appear in a response. It MUST NOT appear in a response if it was not in the request.
- o Result code most significant bit now indicates permanent/temporary error
- o PCP Options are split into mandatory-to-process ("P" bit), and into Specification Required and Private Use.
- o Epoch discussion simplified.

Authors' Addresses

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

Email: dwing@cisco.com

Stuart Cheshire
Apple Inc.
1 Infinite Loop
Cupertino, California 95014
USA

Phone: +1 408 974 3207
Email: cheshire@apple.com

Mohamed Boucadair
France Telecom
Rennes, 35000
France

Email: mohamed.boucadair@orange.com

Reinaldo Penno
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, California 95134
USA

Email: repenno@cisco.com

Paul Selkirk
Internet Systems Consortium
950 Charter Street
Redwood City, California 94063
USA

Email: pselkirk@isc.org

PCP Working Group
Internet-Draft
Intended status: Standards Track
Expires: July 16, 2012

M. Boucadair
France Telecom
R. Penno
Juniper Networks
D. Wing
Cisco
January 13, 2012

DHCP Options for the Port Control Protocol (PCP)
draft-ietf-pcp-dhcp-02

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

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

Table of Contents

1. Introduction	3
2. Terminology	3
3. Rationale	3
4. Consistent NAT and PCP Configuration	4
5. IP Address Selection	5
5.1. Serial Queries	5
5.2. Parallel Queries	5
6. DHCPv6 PCP Server Option	5
6.1. Format	6
6.2. Client Behaviour	6
7. DHCPv4 PCP Option	7
7.1. Format	7
7.2. Client Behaviour	8
8. Dual-Stack Hosts	8
9. Security Considerations	9
10. IANA Considerations	9
10.1. DHCPv6 Option	9
10.2. DHCPv4 Option	9
11. Acknowledgements	9
12. References	9
12.1. Normative References	9
12.2. Informative References	10
Authors' Addresses	11

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:

- o PCP Server denotes a functional element which receives and processes PCP requests from a PCP Client. A PCP Server can be co-located with or be separated from the function (e.g., NAT, Firewall) it controls. Refer to [I-D.ietf-pcp-base].
- o PCP Client denotes a PCP software instance responsible for issuing PCP requests to a PCP Server. Refer to [I-D.ietf-pcp-base].
- o DHCPv4 refers to the Dynamic Host Configuration Protocol [RFC2131] for IPv4.
- o DHCP refers to both DHCPv4 [RFC2131] and DHCPv6 [RFC3315].
- o DHCP client (or client) denotes a node that initiates requests to obtain configuration parameters from one or more DHCP servers [RFC3315].
- o DHCP server (or server) refers to a node that responds to requests from DHCP clients [RFC3315].
- o 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:

- o The use of the Name for NAT64 [RFC6146] might be suitable for load-balancing purposes;
- o 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.

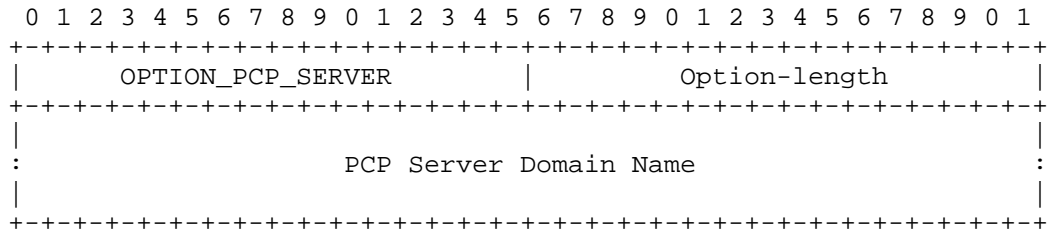


Figure 1: PCP Server Name DHCPv6 Option

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

- o Option-code: OPTION_PCP_SERVER (TBA, see Section 10.1)
- o Option-length: Length of the 'PCP Server Domain Name' field in octets.
- o 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.

Code	Length	PCP Server Domain Name					
-----	-----	-----	-----	-----	-----	-----	-----
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:

- o Code: OPTION_PCP_SERVER (TBA, see Section 10.2);
- o Length: Includes the length of the "PCP Server Domain Name" field in octets; The maximum length is 255 octets.
- o 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:

Option Name	Value
OPTION_PCP_SERVER	TBA

10.2. DHCPv4 Option

Authors of this document request the following DHCPv4 option code:

Option Name	Value
OPTION_PCP_SERVER	TBA

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]
Wing, D., Cheshire, S., Boucadair, M., Penno, R., and P. Selkirk, "Port Control Protocol (PCP)",
draft-ietf-pcp-base-21 (work in progress), January 2012.
- [RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, November 1987.

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2131] Droms, R., "Dynamic Host Configuration Protocol", RFC 2131, March 1997.
- [RFC2132] Alexander, S. and R. Droms, "DHCP Options and BOOTP Vendor Extensions", RFC 2132, March 1997.
- [RFC3315] Droms, R., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, July 2003.
- [RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, February 2006.

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 and Support", STD 3, RFC 1123, October 1989.
- [RFC3493] Gilligan, R., Thomson, S., Bound, J., McCann, J., and W. Stevens, "Basic Socket Interface Extensions for IPv6", RFC 3493, February 2003.
- [RFC6055] Thaler, D., Klensin, J., and S. Cheshire, "IAB Thoughts on Encodings for Internationalized Domain Names", RFC 6055, February 2011.
- [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.
- [RFC6296] Wasserman, M. and F. Baker, "IPv6-to-IPv6 Network Prefix Translation", RFC 6296, June 2011.
- [RFC6333] Durand, A., Droms, R., Woodyatt, J., and Y. Lee, "Dual-

Stack Lite Broadband Deployments Following IPv4
Exhaustion", RFC 6333, August 2011.

[RFC6334] Hankins, D. and T. Mrugalski, "Dynamic Host Configuration
Protocol for IPv6 (DHCPv6) Option for Dual-Stack Lite",
RFC 6334, 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

PCP Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 13, 2012

M. Boucadair
France Telecom
F. Dupont
Internet Systems Consortium
R. Penno
Juniper Networks
D. Wing
Cisco
March 12, 2012

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:

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

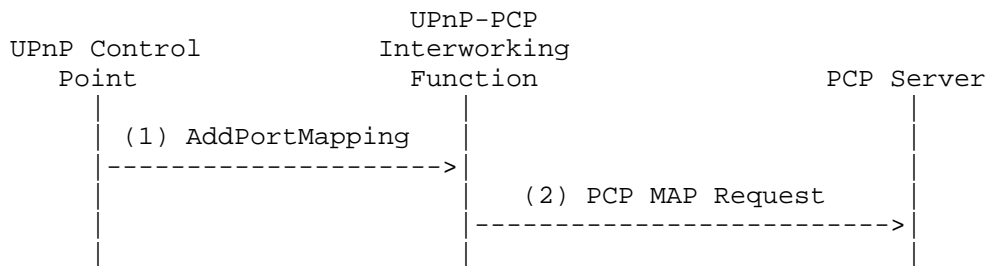


Figure 1: Flow Example

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:

CP router Customer Premise router
 DS-Lite Dual-Stack Lite
 IGD Internet Gateway Device
 IWF Interworking Function
 NAT Network Address Translation
 PCP Port Control Protocol
 UPnP Universal Plug and Play

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:

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

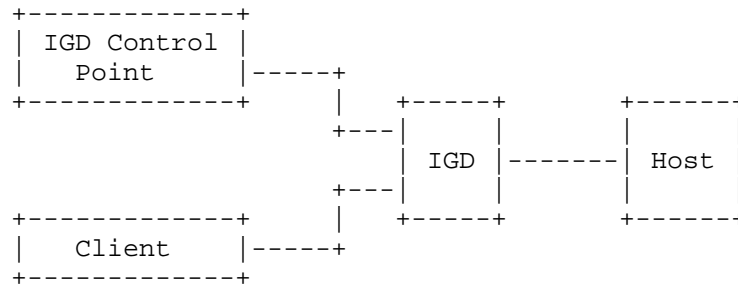


Figure 2: UPnP IGD Model

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.

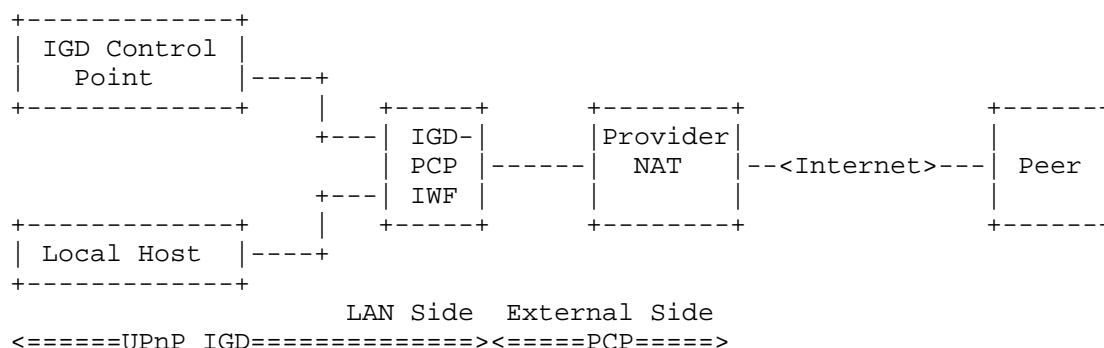


Figure 3: UPnP IGD-PCP Interworking Model

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).

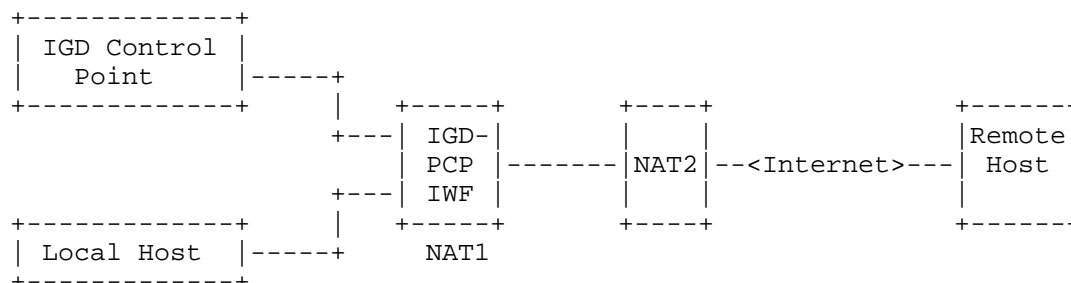


Figure 4: Cascaded NAT scenario

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).

ExternalIPAddress: 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 relaying 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:

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

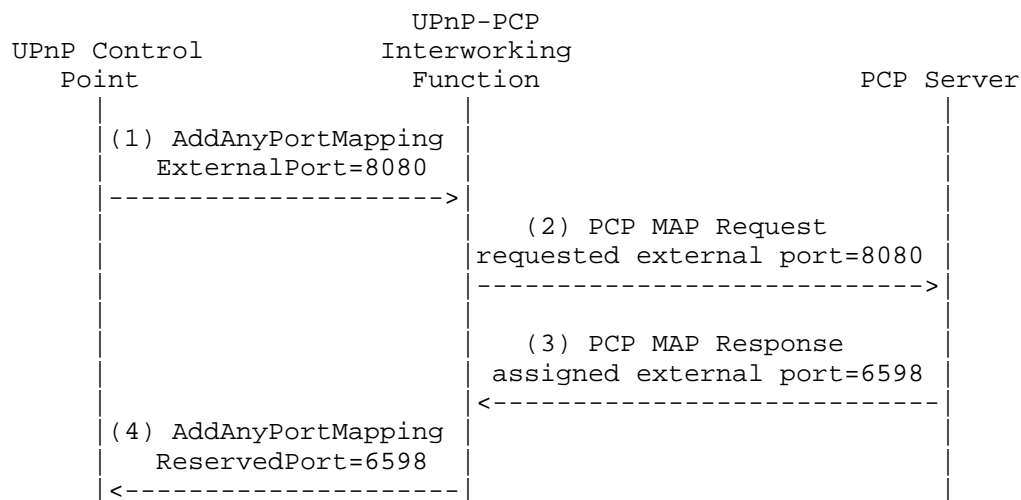
`AddAnyPortMapping()` is the RECOMMENDED method.

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.

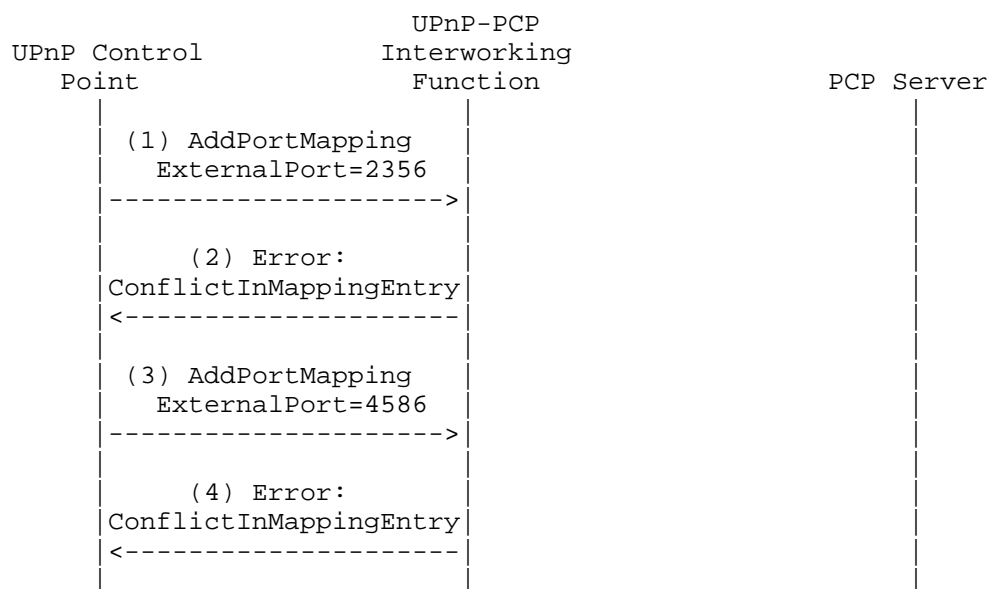
Justification: Some applications (e.g., uTorrent, Vuze, Emule) wait approximately 150s, 90s, 90s, respectively for a response after sending an UPnP request. If a short lifetime error occurs, re-sending the request may lead to a positive response from the PCP Server. UPnP Control Points are therefore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

Figure 5: Flow example when `AddAnyPortMapping()` is used

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 option 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 uses `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

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.

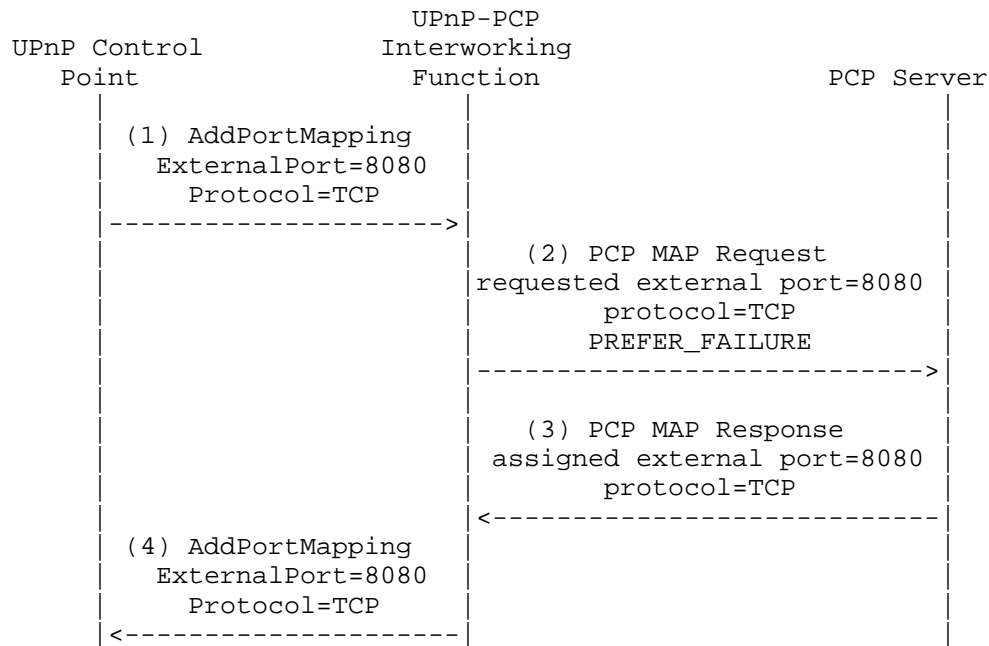


Figure 7: Flow Example (Positive Answer)

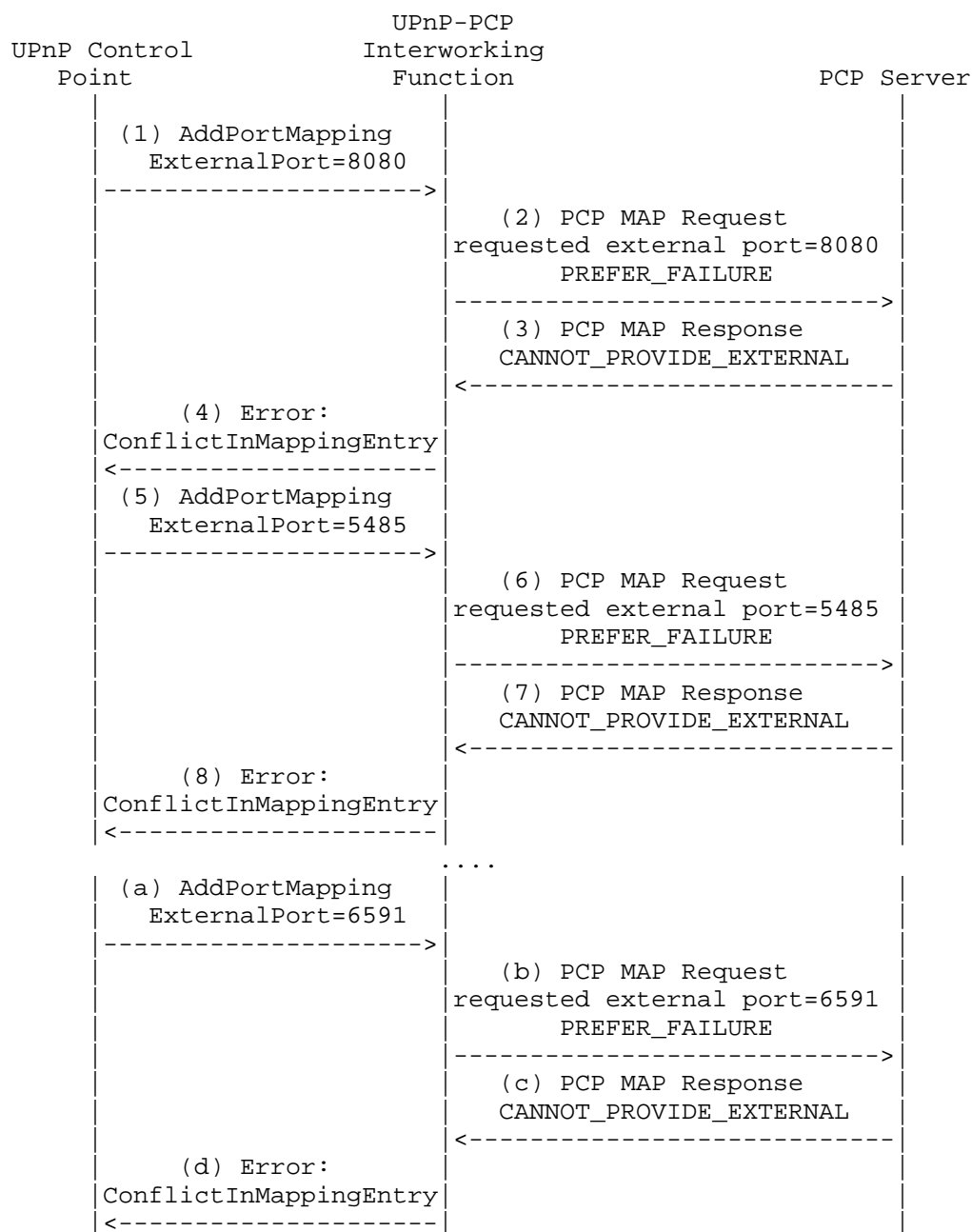


Figure 8: Flow Example (Negative Answer)

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 IWF MUST 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 IWF MUST send back a positive answer. If the external port is already in use by another UPnP Control Point, the IGD-PCP IWF MUST send back a `ConflictInMappingEntry` error to the requesting UPnP Control Point. If no mapping is found in the local mapping table, the IWF MUST 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 IWF.

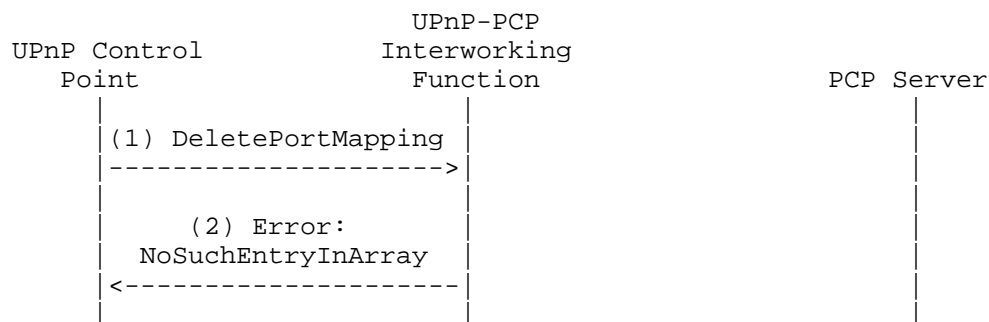


Figure 9: Local Delete (IGD-PCP IWF)

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.

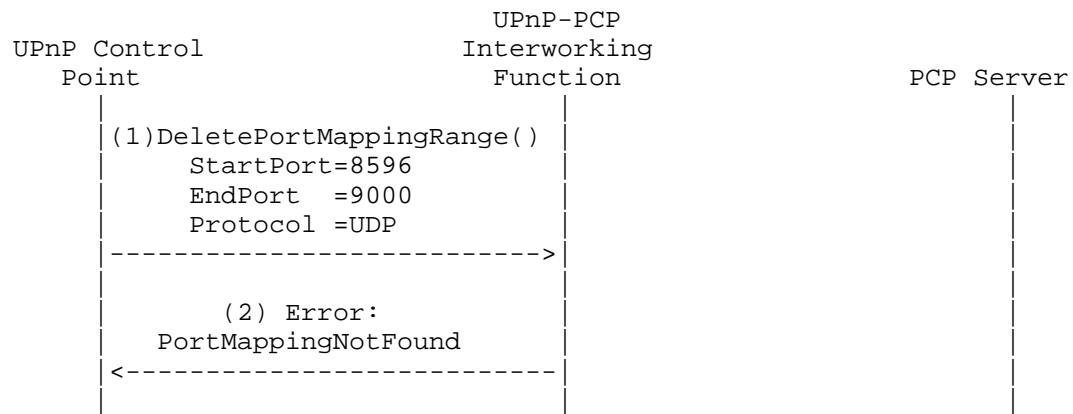


Figure 10: Flow example when an error encountered when processing DeletePortMappingRange()

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.

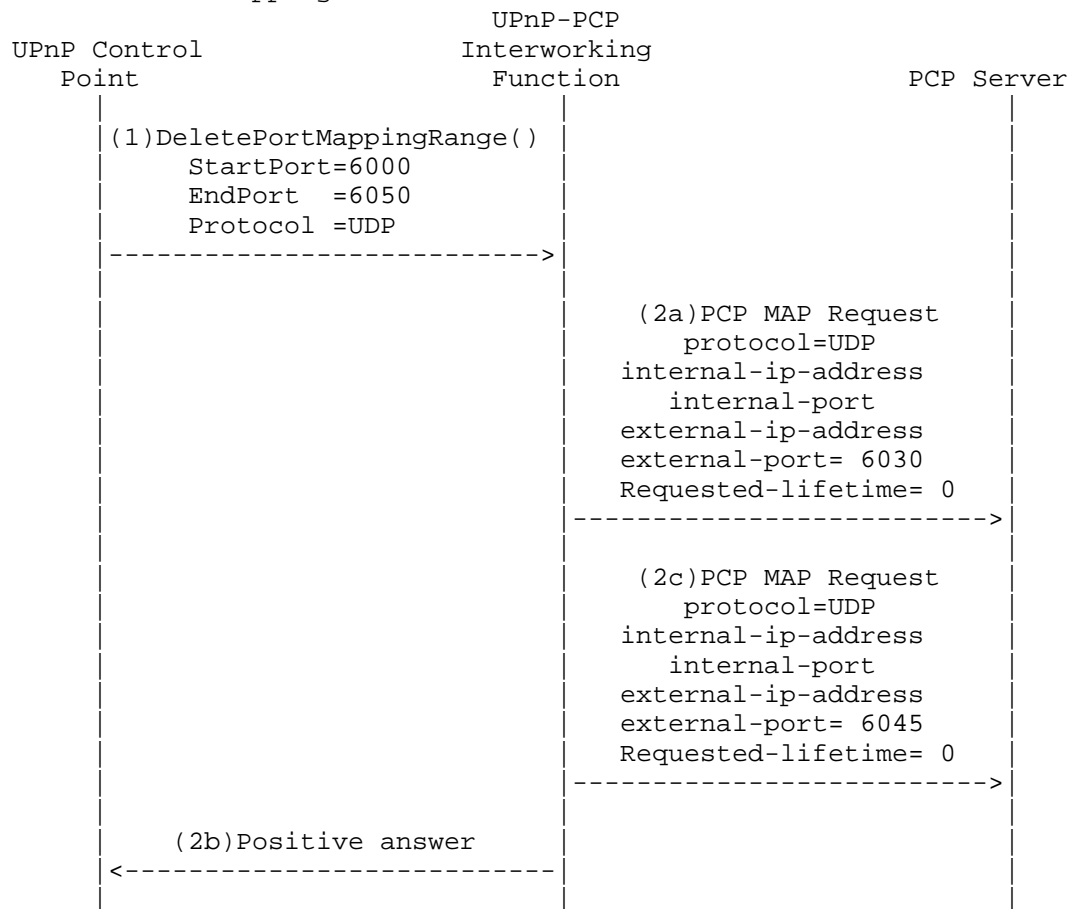


Figure 11: Example of DeletePortMappingRange()

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]
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 Requirement Levels", BCP 14, RFC 2119, March 1997.

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.
- [IGD2] UPnP Forum, "WANIPConnection:2 Service (<http://upnp.org/specs/gw/UPnP-gw-WANIPConnection-v2-Service.pdf>)", September 2010.
- [RFC6146] Bagnulo, M., Matthews, P., and I. van Beijnum, "Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers", RFC 6146, April 2011.

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

[Sec_DCP] UPnP Forum, "Device Protection:1", November 2009.

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

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 8, 2013

Y. Ohba
Y. Tanaka
Toshiba
S. Das
ACS
July 7, 2012

Provisioning Message Authentication Key for PCP using PANA
draft-ohba-pcp-pana-00

Abstract

This document specifies a mechanism for provisioning PCP (Port Control Protocol) message authentication key using PANA (Protocol for carrying Authentication for Network Access).

Status of this Memo

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

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

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

This Internet-Draft will expire on January 8, 2013.

Copyright Notice

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

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

Table of Contents

1. Introduction	3
1.1. Specification of Requirements	3
2. Establishing a PCP SA	3
3. Security Considerations	4
4. IANA Considerations	5
5. Acknowledgments	5
6. References	5
6.1. Normative References	5
6.2. Informative References	5
Authors' Addresses	5

1. Introduction

PCP (Port Control Protocol) [I-D.ietf-pcp-base] is used for an IPv6 or IPv4 host to control how incoming IPv6 or IPv4 packets are translated and forwarded by a network address translator (NAT) or simple firewall, and also allows a host to optimize its outgoing NAT keepalive messages.

In order to provide integrity protection for PCP messages, a message authentication mechanism for PCP is defined in [I-D.ietf-pcp-authentication]. A PCP Security Association (SA) used for PCP message authentication is dynamically established using EAP authentication where the integrity key associated the PCP SA is derived from EAP MSK (Master Session Key) [RFC3748]. In [I-D.ietf-pcp-authentication], two approaches are identified for establishing a PCP SA. The first approach is separate key management that is based on running PANA (Protocol for carrying Authentication for Network Access) [RFC5191] between the end-points to carry out an EAP authentication process needed for establishing a PCP SA. The second approach is inline key management that is based on running an EAP authentication process between two PCP devices.

In this document, a complete solution for the first approach is described.

1.1. Specification of Requirements

In this document, several words are used to signify the requirements of the specification. These words are often capitalized. The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

2. Establishing a PCP SA

A PaC (PANA Client) on a PCP client node initiates PANA authentication prior to send an authenticated PCP message. The initiation may be requested by the PCP client. We assume that PAA (PANA Authentication Agent) is implemented on each PCP server that support authenticated PCP messages. Therefore, the PCP server's IP address is used as the address of the PAA. The PANA authentication for establishing a PCP SA may be conducted as part of network access authentication or dedicated to the PCP usage.

Upon successful PANA authentication, the message authentication key for PCP message is derived from the EAP MSK as follows:

`PCP_AUTH_KEY = prf+(MSK, "IETF PCP" | SID | KID | PCP_Server_ID)`

where `|` denotes concatenation.

- o The `prf+` function is defined in IKEv2 [RFC5996]. The pseudo-random function to be used for the `prf+` function is negotiated using PRF-Algorithm AVP in the initial PANA-Auth-Request and PANA-Auth-Answer exchange with 'S' (Start) bit set.
- o "IETF PCP" is the ASCII code representation of the non-NULL terminated string (excluding the double quotes around it).
- o SID is a four-octet PANA Session Identifier [RFC5191].
- o KID is the content of the Key-ID AVP [RFC5191] associated with the MSK.
- o PCP_Server_ID is the IP address of the PCP server. The length of PCP_Server_ID is 4 octets for IPv4 address and 16 octets for IPv6 address.

The same integrity algorithm used for the PANA session MUST be used for PCP message authentication.

The PCP_AUTH_KEY and its associated parameters (i.e., the IP addresses of the PCP client and PCP server, Session ID, Key ID, message authentication algorithm and lifetime) are passed from the PAA application to the PCP server application on the same PCP server device, and also passed from the PaC application to the PCP client application on the same PCP client device, using an API. The API can be implementation-specific, and therefore is not specified in this document.

Once a PCP SA is established, any PCP message that does not contain a valid Authentication Tag and a fresh Nonce under the current PCP SA MUST be silently discarded.

The PCP SA MUST be immediately deleted when the corresponding PANA SA is deleted. The PCP SA SHALL remain as long as the corresponding PANA SA exists.

3. Security Considerations

The key provisioning mechanism described in this document provides a cryptographic binding between a PANA session and a PCP SA based on using the PCP server address, and PANA session identifier and key identifier in the PCP_AUTH_KEY derivation function.

4. IANA Considerations

There is no IANA actions required for this document.

5. Acknowledgments

TBD.

6. References

6.1. Normative References

- [RFC3748] Aboba, B., Blunk, L., Vollbrecht, J., Carlson, J., and H. Levkowetz, "Extensible Authentication Protocol (EAP)", RFC 3748, June 2004.
- [RFC5191] Forsberg, D., Ohba, Y., Patil, B., Tschofenig, H., and A. Yegin, "Protocol for Carrying Authentication for Network Access (PANA)", RFC 5191, May 2008.
- [RFC5996] Kaufman, C., Hoffman, P., Nir, Y., and P. Eronen, "Internet Key Exchange Protocol Version 2 (IKEv2)", RFC 5996, September 2010.
- [I-D.ietf-pcp-base]
Wing, D., Cheshire, S., Boucadair, M., Penno, R., and P. Selkirk, "Port Control Protocol (PCP)", draft-ietf-pcp-base-26 (work in progress), June 2012.
- [I-D.ietf-pcp-authentication]
Wasserman, M., Hartman, S., and D. Zhang, "Port Control Protocol (PCP) Authentication Mechanism", draft-ietf-pcp-authentication-00 (work in progress), June 2012.

6.2. Informative References

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

Authors' Addresses

Yoshihiro Ohba
Toshiba Corporate Research and Development Center
1 Komukai-Toshiba-cho
Saiwai-ku, Kawasaki, Kanagawa 212-8582
Japan

Phone: +81 44 549 2127
Email: yoshihiro.ohba@toshiba.co.jp

Yasuyuki Tanaka
Toshiba Corporate Research and Development Center
1 Komukai-Toshiba-cho
Saiwai-ku, Kawasaki, Kanagawa 212-8582
Japan

Phone: +81 44 549 2127
Email: yatch@isl.rdc.toshiba.co.jp

Subir Das
Applied Communication Sciences
1 Telcordia Drive
Piscataway, NJ 08854
USA

Email: sdas@appcomsci.com

Port Control Protocol
Internet-Draft
Intended status: Standards Track
Expires: January 3, 2013

R. Penno
D. Wing
Cisco
P. Selkirk
Internet Systems Consortium
M. Boucadair
France Telecom
July 02, 2012

PCP Support for Nested NAT Environments
draft-penno-pcp-nested-nat-02

Abstract

Nested NATs or multi-layer NATs are already widely deployed. They are characterized by two or more NAT devices in the path of packets from the subscriber to the Internet. Moreover, NAT devices currently deployed are PCP unaware and it is assumed that NAT aware PCP devices will take a long time to be rolled out. Therefore in order to lower the adoption barrier of PCP and make it work for current deployed networks, this document proposes a few mechanisms for PCP-enabled applications to work through nested NATs with varying levels of PCP protocol support.

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 January 3, 2013.

Copyright Notice

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

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

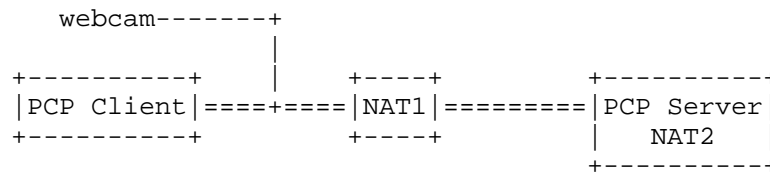
Table of Contents

1. Introduction	3
1.1. Terminology	3
1.2. Problem Statement	3
1.3. Scope	4
2. PCP Nested NAT Methods	4
2.1. PCP and UPnP unaware Intermediate NATs	5
2.2. PCP Server intermediate NAT	7
2.3. UPnP enabled intermediate NAT	8
2.4. PCP Proxy Intermediate NAT	8
2.4.1. PCP Proxy Discovery	9
3. RECEIVED_SOURCE_PORT Option	9
4. SCOPE Option	10
5. IANA Considerations	11
6. Security Considerations	11
7. Acknowledgements	11
8. References	11
8.1. Normative References	11
8.2. Informative References	12
Authors' Addresses	12

1. Introduction

Nested NATs are widely deployed and come in different topology flavors. It could be a home subscriber which has an ISP provided NAT CPE chained with another personal NAT router. It could be an ISP provided CPE chained with a CGN.

An example of the use of the proposed options is illustrated in the following figure where there is a NAT in the path between the PCP Client and the PCP Server.



An example of instructing mappings in the PCP Server is as follows:

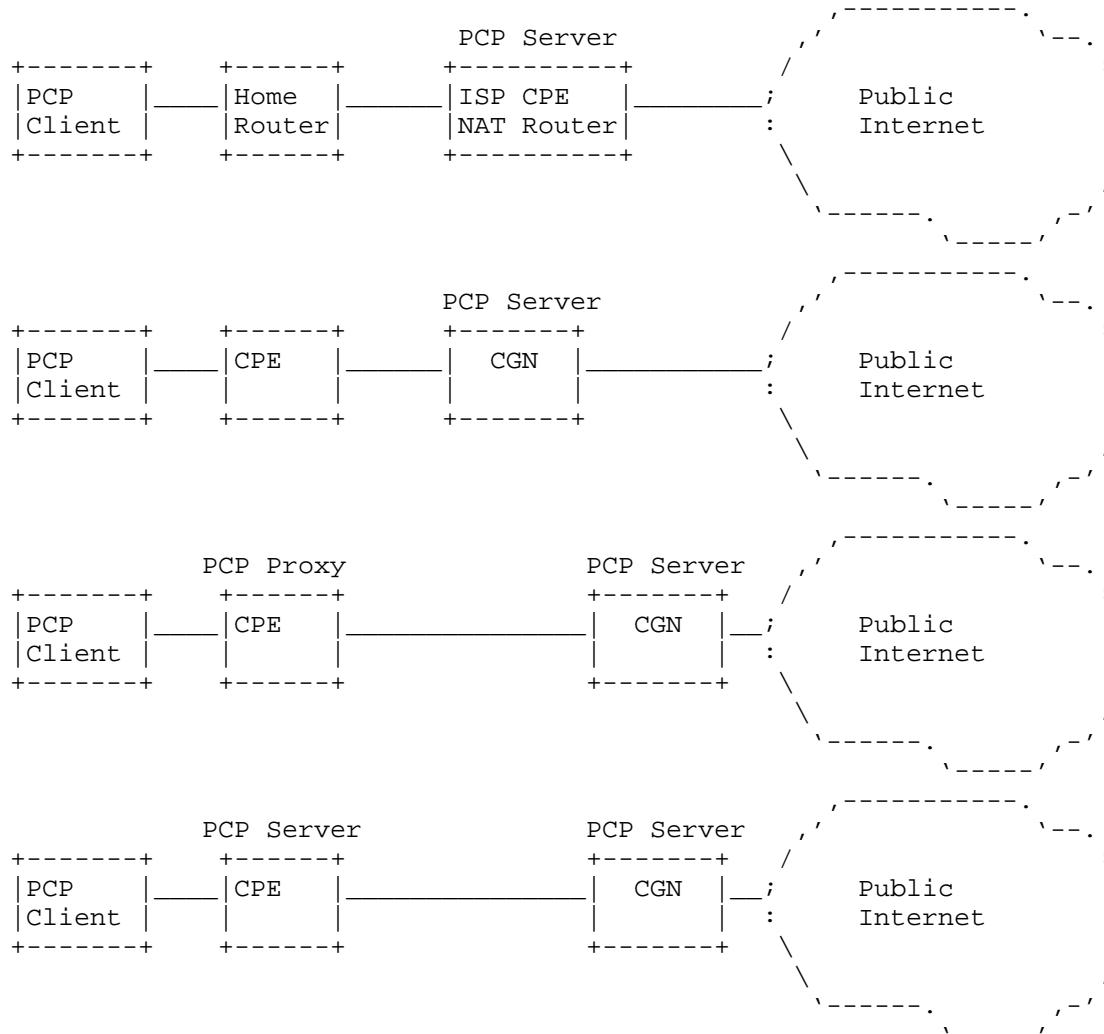
- o NAT1 is detected in the path between the PCP Client and the PCP Server owing to the use of the RECEIVED_PORT Option and returned perceived IP address in PCP response;
- o After learning about that NAT, the PCP Client uses UPnP IGD, NAT-PMP or manual configuration to interact with NAT1 and open the necessary port on NAT1 (e.g., IP address= IPx, port=X);
- o The PCP Client then sends PCP message to the PCP Server, indicating IPx and X as the internal IP address and port. The PCP Server opens pinhole towards IPx and X.

1.1. Terminology

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

1.2. Problem Statement

The current NAT deployed devices will take years to be replaced or upgraded to become PCP aware. Moreover, nested NATs are common and come in a variety of flavors (examples below). Therefore, as applications become PCP enabled, it is important that they can work through nested NAT networks as is, without requiring infrastructure changes. From the point of view of a PCP-enabled application running on an end host, the core problem is common across different nested NAT topologies: how to install PCP mappings in a nested NAT scenario where the different NATs in the path have varying level of PCP protocol support.



1.3. Scope

This proposal considers the discovery of the PCP Server out of scope. Nonetheless, it is a critical piece of PCP deployment in service provider networks.

2. PCP Nested NAT Methods

There are a few methods to make PCP work through nested NATs. They differ mainly based on the level of support that can be expected from

intermediate NATs, which can be:

- o PCP and UPnP unaware or disabled
- o PCP Server
- o UPnP Server
- o PCP Proxy

The next sections discuss each scenario on the basis of protocol support on intermediate NATs.

2.1. PCP and UPnP unaware Intermediate NATs

This method will most likely be used by PCP clients in nested NAT environments while PCP Proxy support is not ubiquitous. It assumes no UPnP or PCP Proxy support on intermediate NATs. This proposal leverages the current behavior of PCP [I-D.ietf-pcp-base] which allows a PCP Client and Server to detect intervening nested NATs. The PCP Server uses the information on the outer IP and PCP headers to detect and install a proper NAT mapping and return the source IP:port from the IP header on the PCP response. It does not assume any change to current deployed NATs.

1. The PCP Client sends the MAP request as it normally would without any changes.
2. As the message goes through one (or more) PCP-unaware NAT, the source IP:port of the IP header will change accordingly
3. The PCP Server compares the PCP Client IP:port in the PCP header with the source IP:port of the IP header
4. If these are different, the server knows that the PCP message went through a PCP-unaware NAT. Therefore it installs a mapping directed to the source IP address found on the IP header and internal port of the PCP header.

s/dport: source/destination port
 s/dIP : source/destination IP
 PCP-C : PCP client
 iport : Internal port
 PCP-U : PCP Unaware NAT
 E-port : External port
 E-IP : External IP

PCP Client

PCP-U NAT

PCP Server

Map request		
Outer sIP:192.68.0.2		
Outer sPort:19268	Map request	
PCP-C Addr:192.168.0.1	Outer sIP:10.0.0.2	
PCP-C port:19268	Outer sPort:10002	
iPort:40000	PCP-C Addr:192.168.0.1	
----->	PCP-C port:19268	
	iPort:40000	
	----->	
		PCP client IP != Outer IP
		Allocate public IP and port
		Mapping:
		(10.0.0.2, 40000) <- (20.0.0.1, 20001)
	Map response	
	Outer dIP:10.0.0.2	
	Outer dport:10002	
	Assigned E-port:20001	
Map response	Assigned E-IP:20.0.0.1	
Outer dIP:192.168.0.2	PCP-C Addr:10.0.0.2	
Outer dport:19268	PCP-C port:10002	
Assigned E-port:20001	<-----	
Assigned E-IP:20.0.0.1		
PCP-C Addr:10.0.0.2		
PCP-C port:10002		
<-----		

- Subscriber installs a port forwarding or DMZ entry on its home CPE (PCP U-NAT) through manual configuration. The entry would be (*, 40000) -> (10.0.0.1, 40000). Alternatively the application could use UPnP for the same purpose.

2.2. PCP Server intermediate NAT

If the intermediate NAT implements a PCP Server (but not a Proxy), a two-step iterative process is needed in order to install PCP PEER mappings for the PCP control message itself followed by another PCP mapping for the data path. If the PCP client relies on nested NAT detection the first step is not needed. It is assumed that before the PCP MAP request to the CGN the client would install the following map on the NAT Home Gateway: (192.168.0.2, 40000) <- (10.0.0.2, 40000). The internal port that the server listens on does not necessarily need to be 40000, it could be different than the internal port used between the CGN and CPE.

The drawback of this technique is that there is no obvious way for the PCP Client to know the PCP Servers downstream. One possibility is for each PCP Server in the path to return the address of the upstream PCP Server to the PCP Client.

PCP Client	PCP Server (CPE)	PCP Server (CGN)
------------	------------------	------------------

<pre> PEER request Outer sIP:192.168.0.2 Outer sPort:19216 PCP-C Addr:192.168.0.2 PCP-C port:19216 iPort:19216 Remote Port:44323 Remote IP: 10.0.0.1 -----> PEER response Outer sIP:192.168.0.1 Outer sPort: 19216 Assigned E-port: 10002 Assigned E-IP: 10.0.0.2 PCP-C Addr:192.168.0.2 PCP-C port:19216 iPort:19216 Remote Port:44323 Remote IP: 10.0.0.1 <----- (192.68.0.2,19216) -> (10.0.0.2,10002) Dest: 10.0.0.1, 44323 Map request Outer sIP:192.168.0.2 Outer sPort:19216 PCP-C Addr:10.0.0.2 PCP-C port:10002 </pre>	<pre> </pre>	<pre> </pre>
---	--------------	--------------

iPort:40000 ----->	Map request Outer sIP:10.0.0.2 Outer sPort:10002 PCP-C Addr:10.0.0.2 PCP-C port: 10002 iPort:40000 ----->	
		(10.0.0.2, 40000) <- (20.0.0.1, 20001)
Map response Outer dIP:192.168.0.2 Outer dport:19216 Assigned E-port: 20001 Assigned E-IP: 20.0.0.1 PCP-C Addr: 10.0.0.2 PCP-C port: 10002 <-----	Map response Outer dIP:10.0.0.2 Outer dport: 10002 Assigned E-port: 20001 Assigned E-IP: 20.0.0.1 PCP-C Addr: 10.0.0.2 PCP-C port: 10002 <-----	

2.3. UPnP enabled intermediate NAT

This scenario is very similar to the PCP Server intermediate NAT, but the CPE implements a UPnP Server instead of PCP Server. The mechanics are the same with the difference that first PEER message to setup the PCP Control messages mapping is substituted by its UPnP equivalent.

2.4. PCP Proxy Intermediate NAT

This method assumed that the intermediate NATs implement a PCP Proxy function. There are two non-exclusive types of proxy functions: interception (ALG) and server-client based. In the interception case the PCP Proxy intercepts PCP messages destined to a PCP Server downstream, modifies IP, UDP and PCP headers, allocates a mapping and send them to the downstream PCP Server. Ideally if the interception PCP Proxy also implements a PCP server it would let the PCP Client know of its existence in a PCP response through an option (TBD) and henceforth the PCP Client would start directing messages to it.

In the server-client scenario the PCP Client sends PCP messages to the proxy which acts as both PCP Server and Client. This proxy in turn will terminate the PCP request and generate a new one acting as

a PCP Client to its own PCP Server. Therefore mappings are installed in all NAT devices in a recursive manner. This is the recommended method since it does not need a special discovery procedure and works with any number of NATs. More information about this method can be found in [I-D.bpw-pcp-proxy].

2.4.1. PCP Proxy Discovery

TBD

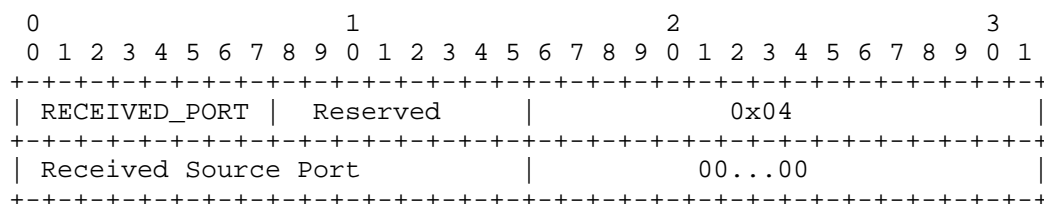
3. RECEIVED_SOURCE_PORT Option

This option (Code TBA, Figure 1) is used by a PCP Server to indicate in a PCP response the source port of PCP messages received from a PCP Client. Together with the IP Address of the PCP Client conveyed in the common PCP header, a PCP Client uses this information to detect whether a NAT is present in the path to reach its PCP Server.

A PCP Client MAY include this option to learn the port number as perceived by the PCP Server. When this option is received by the PCP Server, it uses the source port of the received PCP request to set the Received Port.

This Option:

Option Name: PCP Received Port Option (RECEIVED_PORT)
 Number: TBA (IANA)
 Purpose: Detect the presence of a NAT in the path
 Valid for Opcodes: MAP
 Length: 0x04
 May appear in: both request and response
 Maximum occurrences: 1



Received Source Port: The source port number of the received PCP request as seen by the PCP Server.

Figure 1: Received IP address/port PCP option

4. SCOPE Option

The Scope Option (Code TBA, Figure 2) is used by a PCP Client to indicate to the PCP Server the scope of the flows that will use a given mapping. This object is meant to be used in the context of cascaded PCP Servers/NAT levels. Two values are defined:

Value	Meaning
0x00	Internet
0x01	Internal

When 0x00 value is used, the PCP Proxy MUST propagate the mapping request to its upstream PCP Server. When 0x01 value is used, the mapping is to be instantiated only in the first PCP-controlled device; no mapping is instantiated in the upstream PCP-controlled device.

When no Scope Option is included in a PCP message, this is equivalent to including a Scope Option with a scope value of "Internet".

This Option:
Option Name: PCP Scope Policy Option (SCOPE)
Number: TBA (IANA)
Purpose: Restrict the scope of PCP requests
Valid for Opcodes: MAP
Length: 0x04
May appear in: both request and response
Maximum occurrences: 1

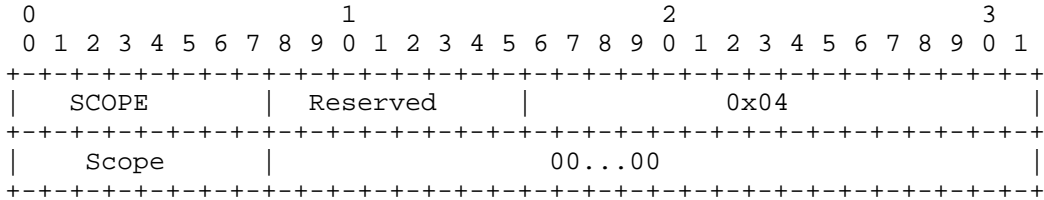


Figure 2: Scope Option

5. IANA Considerations

The following PCP Option Codes are to be allocated:

- RECEIVED_PORT
- SCOPE

6. Security Considerations

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

7. Acknowledgements

TBD

8. References

8.1. Normative References

- [I-D.ietf-pcp-base]
Wing, D., Cheshire, S., Boucadair, M., Penno, R., and P.

Selkirk, "Port Control Protocol (PCP)",
draft-ietf-pcp-base-26 (work in progress), June 2012.

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

8.2. 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.

Authors' Addresses

Reinaldo Penno
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, California 95134
USA

Email: repenno@cisco.com

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

Email: dwing@cisco.com

Paul Selkirk
Internet Systems Consortium
950 Charter Street
Redwood City, California 94063

Phone:
Fax:
Email: pselkirk@isc.org
URI:

Mohamed Boucadair
France Telecom
Rennes, 35000
France

Email: mohamed.boucadair@orange.com

PCP Working Group
Internet-Draft
Intended status: Standards Track
Expires: December 25, 2012

T. Reddy
P. Patil
D. Wing
F. Baker
Cisco
June 23, 2012

PCP Server Discovery in IPv6 Multihoming
draft-reddy-pcp-server-discovery-00

Abstract

A multihomed network may have a PCP server on each router connecting to each upstream network, providing firewall or prefix translation functions to hosts in the network. In these networks, a PCP client needs to discover all of those PCP servers and then send PCP requests to them individually.

This document proposes a multicast mechanism to discover PCP servers.

Status of this Memo

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

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

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

This Internet-Draft will expire on December 25, 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. Notational Conventions	3
3. DISCOVER OpCode	3
3.1. PCP Server joining a multicast group	4
3.2. Generating a DISCOVER Request	4
3.3. Processing a DISCOVER Request	5
3.4. Processing a DISCOVER Response	5
3.5. Discover Option Packet Format	6
4. Operational Considerations	6
5. Security Considerations	7
6. IANA Considerations	7
7. References	7
7.1. Normative References	7
7.2. Informative References	8
Authors' Addresses	8

1. Introduction

Using Port Control Protocol (PCP) [I-D.ietf-pcp-base] a host can create mappings with its NAT or firewall. PCP expects only one PCP server. In a multihomed network, there may be multiple PCP servers and the PCP client is unaware of all designated PCP Servers in the network. For example, there may be a PCP server integrated into every firewall device connecting to each network. Hence there is a need for PCP client to discover all such PCP Servers with specific functionalities so that the PCP client can make appropriate PCP requests to each one of them.

This document proposes a means by which a PCP client can discover all such PCP servers within the network. Each PCP server in the network joins a certain multicast. Using the new DISCOVER OpCode, defined in this document, each PCP client sends a DISCOVER request to that multicast group address. Each PCP server responds with a DISCOVER response. The PCP client then sends regular unicast PCP request messages (e.g., MAP or PEER OpCodes) to each of those discovered PCP servers.

2. Notational Conventions

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

3. DISCOVER OpCode

DISCOVER : Discover PCP servers listening on specific multicast groups.

PCP Servers SHOULD provide a configuration option to allow administrators to disable DISCOVER support if they wish. PCP DISCOVER requests are only designed to discover appropriate PCP servers on the network. The request does not offer functionality defined by other OpCodes described in [I-D.ietf-pcp-base].

The following diagram shows the usage of DISCOVER OpCode, where PCP Server1 and PCP Server2 join the same multicast group (for e.g. ALL-IPv6-FIREWALLS)

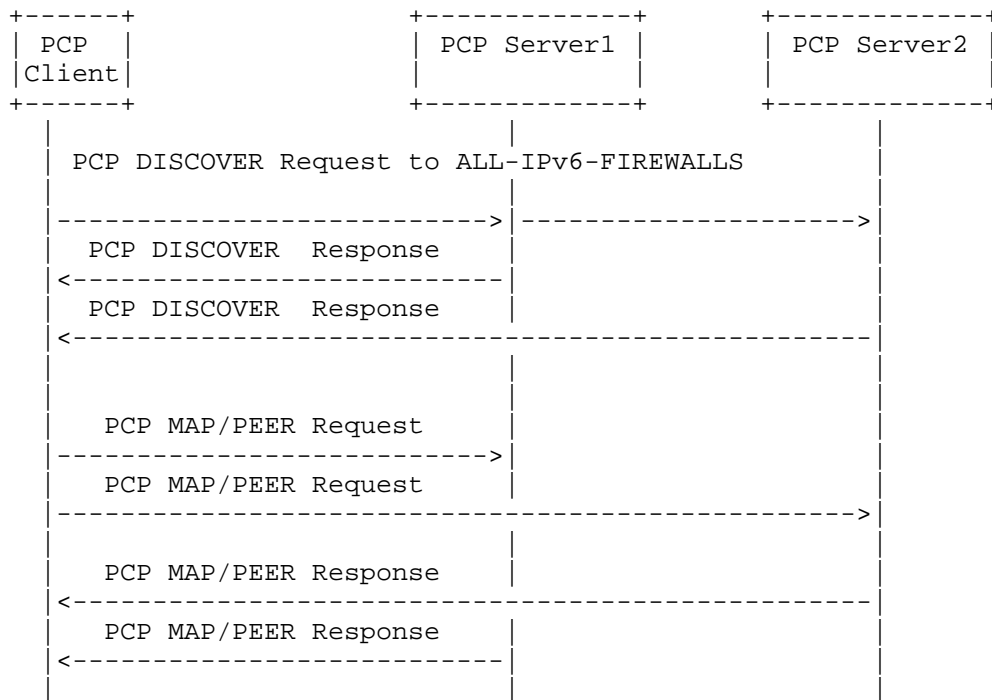


Figure 1: PCP Server Discover

3.1. PCP Server joining a multicast group

Each PCP server in the network joins a certain multicast group based on other functionalities embedded with it. Consider a scenario in which a firewall also implements a Port Control Protocol (PCP) [I-D.ietf-pcp-base] Server, in which case it joins a multicast group ALL-IPv6-FIREWALLS.

A PCP server can join more than one mutlicast groups if it offers multiple functionalities within the same device.

3.2. Generating a DISCOVER Request

To discover the PCP servers listening on each of the assigned multicast addresses of interest to the PCP client, the PCP client sends a DISCOVER request to each of those multicast addresses.

A Discover Nonce is included in the request by the PCP client. The Discover Nonce is randomly chosen by the PCP client, and is used as part of validation of PCP responses.

To accommodate packet loss, the request SHOULD be transmitted several times with a random jitter between them to each of the multicast address. It is RECOMMENDED to transmit the DISCOVER 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.

Periodic PCP DISCOVER requests should be made to determine the updated list of PCP servers in the network. A PCP client can send DISCOVER messages periodically every 600 seconds to each of the multicast addresses.

3.3. Processing a DISCOVER Request

When a PCP server listening on one of the multicast groups as described in Section 3.1 receives a PCP DISCOVER Opcode, after successful parsing and processing, it generates a SUCCESS response with zero Assigned Lifetime. If a PCP DISCOVER Request is received on an unassigned multicast group, it should be ignored.

Each PCP Server sends a separate DISCOVER response with unicast source address signaling to the PCP client that the source IPv6 address of DISCOVER response is the PCP Server address. The Discover Nonce field from the request is copied into the PCP DISCOVER response.

3.4. Processing a DISCOVER Response

A DISCOVER request sent to the multicast group will result in zero, one, or more responses from each of the addresses multicasted in Section 3.1. If the network contains multiple servers then multiple DISCOVER responses will normally be received. After regular PCP response processing, a PCP client should check using the Discover Nonce from the response to match with a previously sent DISCOVER request and hence also determine the capability of the PCP server.

If a DISCOVER request results in one or more DISCOVER response then the client can update its PCP server list with the source addresses of all DISCOVER responses. This list is essentially a list of all PCP Servers in the network. Future specifications that use PCP DISCOVER to discover PCP servers will also define how PCP clients will use the discovered PCP server list.

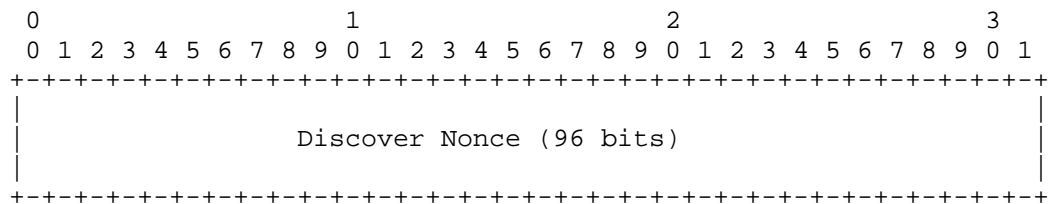
A PCP server may join or leave a network unexpectedly (e.g., device failure, link failure, or link recovery). To accommodate these situations, the PCP client should also periodically send PCP DISCOVER requests to each of the multicast groups to ensure that the client has an updated list of PCP Servers.

3.5. Discover Option Packet Format

The DISCOVER Opcode has a similar layout for both request and response.

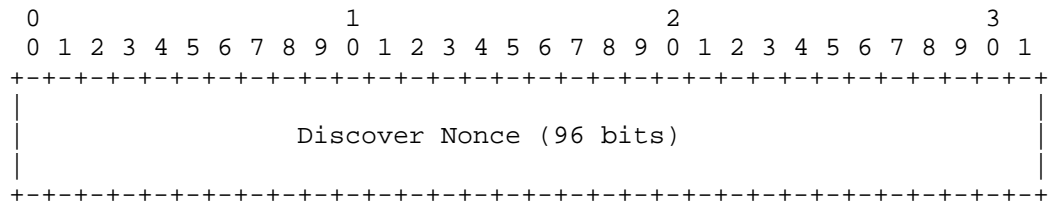
The following diagram shows the format of the Opcode-specific information in a request for the DISCOVER Opcode:

The Requested Lifetime value MUST be set to zero in the PCP common header.



Discover Nonce: Random value chosen by the PCP client.

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



Discover Nonce: Copied from the request.

4. 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 firewalls 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 Firewalls advertising a default route into a specific part of its network.

5. 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 and respond to discovery requests pretending to be PCP servers. Discovery of such rogue systems as PCP servers, in itself, is not a security threat if there is a means for the PCP client to authenticate and authorize the discovered PCP servers.

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

6. IANA Considerations

This note requests of the IANA the assignment of a new PCP Opcode

value	Opcode
----	-----
TBD	DISCOVER

This note also 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-IPv6-FIREWALLS". One address should be assigned for each of the following scopes: Link-Local, Admin-Local, Site-Local, and Organization-Local.

7. References

7.1. Normative References

- [I-D.ietf-pcp-base] Wing, D., Cheshire, S., Boucadair, M., Penno, R., and P. Selkirk, "Port Control Protocol (PCP)", draft-ietf-pcp-base-26 (work in progress), June 2012.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, February 2006.

7.2. Informative References

[v6mult] IANA, "IPv6 Multicast Address Space Registry",
December 2011, <[http://www.iana.org/assignments/
ipv6-multicast-addresses/ipv6-multicast-addresses.xml](http://www.iana.org/assignments/ipv6-multicast-addresses/ipv6-multicast-addresses.xml)>.

Authors' Addresses

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

Email: tiredy@cisco.com

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

Email: praspati@cisco.com

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

Email: dwing@cisco.com

Fred Baker
Cisco Systems, Inc.
Santa Barbara, California 93117
USA

Email: fred@cisco.com

Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: January 17, 2013

Q. Sun
China Telecom
M. Boucadair
X. Deng
France Telecom
C. Zhou
Huawei Technologies
T. Tsou
Huawei Technologies (USA)
July 16, 2012

Lightweight 4over6 Port-set Allocation: Using PCP To Coordinate Between
the CGN and Home Gateway
draft-tsou-pcp-natcoord-07

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 January 17, 2013.

Copyright Notice

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

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

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

Table of Contents

1. Application Scenario	3
2. MAP_PORT_SET Opcode	3
2.1. MAP_PORT_SET Operation Packet Formats	4
2.2. Port-Set Options Formats	6
2.2.1. Port_Range_Option	6
2.2.2. Cryptographically_Random_Port_Range_Option	7
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	11
2.7. Mapping Lifetime and Deletion	11
2.8. PREFER_FAILURE Option for MAP_PORT_SET Opcode	11
3. Security Considerations	11
4. IANA Considerations	11
5. Author List	11
6. Contributor List	12
7. References	12
7.1. Normative References	12
7.2. informative References	13
Authors' Addresses	13

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 in lightweight 4over6 [I-D.cui-softwire-b4-translated-ds-lite], 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 lightweight 4over6 [I-D.cui-softwire-b4-translated-ds-lite] in port-set allocation process.

MAP_PORT_SET: Create an explicit dynamic mapping between an

Internet's IP Address and an External Address + Port set

The format of a port-set can either be contiguous or non-contiguous including a cryptographical 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.

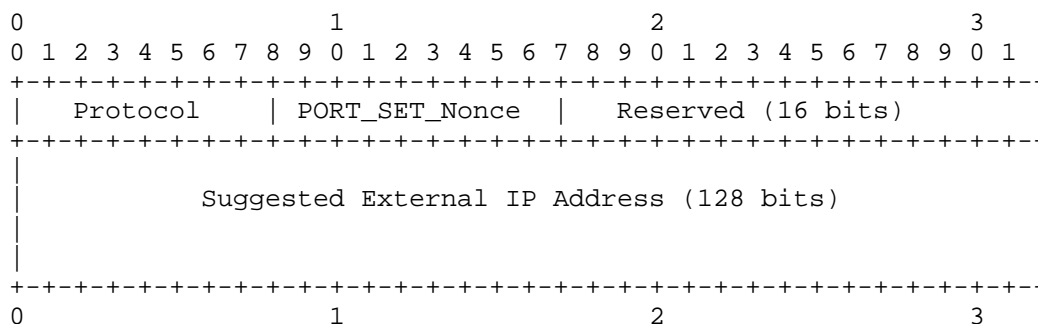


Figure 1: MAP_PORT_SET Opcode format of Request

These fields are described below:

- o Protocol: the default value is zero (to indicate all transport protocols).
- o PORT_SET_NONCE: Incremental or Random Value chosen by the PCP Client, which SHOULD be different for individual PCP requests.

But the same value MUST be kept in one request re-transmission.

- o Reserved bits: 16 bits MUST be set to 0.
- o 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:

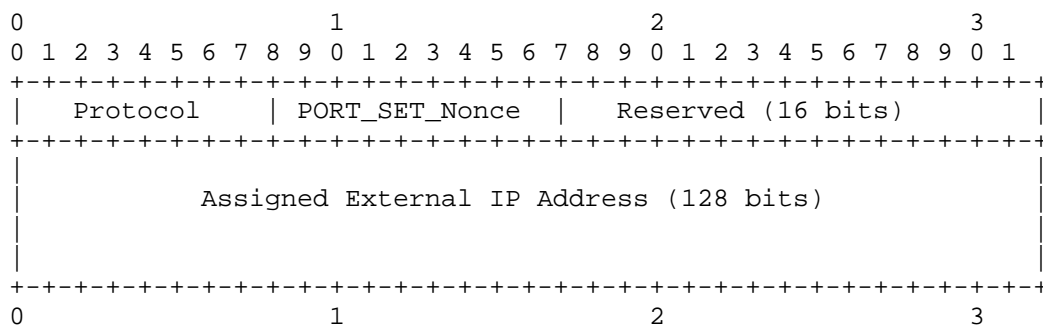


Figure 2: MAP_PORT_SET Opcode format of Response

These fields are described below:

- o Protocol: MUST be copied from the request.
- o PORT_SET_Nonce: MUST be copied from the request.
- o Reserved bits: 16 bits MUST be set to 0.
- o Assigned External IP Address (128 bits): This field conveys the assigned external IPv4 (encoded using IPv4-mapped IPv6 address) or IPv6 address for the mapping. On an error response, the Assigned External IP Address is copied from the request.
- o 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 PORT_RANGE 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

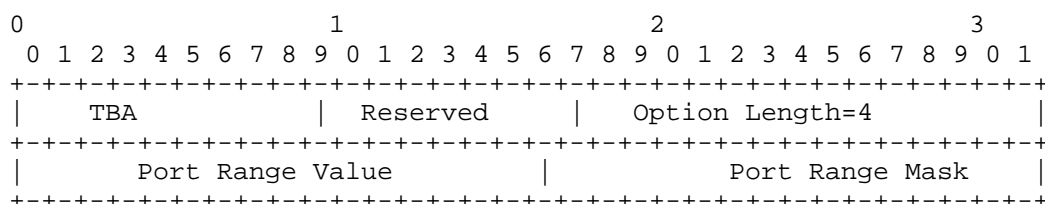


Figure 3: Port_Range_Option

- o Port Range Value (PRV): The PRV indicates the value of the significant bits of the Port Mask. By default, no PRM value is assigned. It can also convey Suggested Port Range Value if the client has a hint on it. In MAP_PORT_SET response, it is an Assigned Port Range Value.
- o 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. It can also convey Suggested Port Range Mask if the client has a hint on it. In MAP_PORT_SET response, it is an Assigned Port Range Mask.

This option:

- o name: Port range option
- o number: TBA
- o 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.

- o is valid for OpCodes:MAP_PORT_SET.
- o length:4 octets
- o may appear in:request and response
- o maximum occurrences:1

2.2.2. Cryptographically_Random_Port_Range_Option

The cryptographically random Port Range PCP Option is formatted as below.

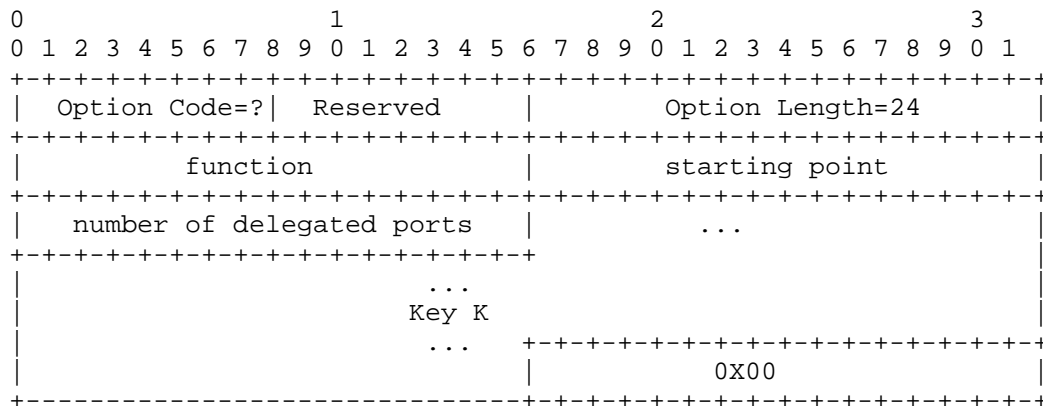


Figure 4: Cryptographically_Random_Port_Range_Option

- o 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:

- o name: Cryptographically Random Port Range Option
- o number: TBA
- o 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.

- o is valid for OpCodes:MAP_PORT_SET.
- o length: 24 octets.
- o may appear in:request and response
- o 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.

If a 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 PORT_SET_Nonce should be copied from the previous MAP_PORT_SET request.

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 renewal can be regarded as a new PCP request with a different PORT_SET_Nonce. The MAP_PORT_SET request MUST 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.

The PCP client SHOULD renew the mapping before its expiry time, otherwise the port-set binding record will be removed by the PCP server.

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.

It is totally up to the server to determine the port-set quota for each subscriber. A PCP server SHOULD maintain MAX_USER_QUOTA and MAX_REQUEST_QUOTA. MAX_USER_QUOTA is to indicate the maximum number of ports a subscriber may get in total, and MAX_REQUEST_QUOTA is to indicate the maximum number of ports in each request. The specific mechanism to configure the quotas is out of scope.

The error codes in MAP_PORT_SET Response mainly have the following possibilities:

- o If the PCP server or PCP-controlled device does not support MAP_PORT_SET Opcode, the error UNSUPP_OPCODE MUST be returned.
- o 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.
- o 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 requested lifetime is zero, it indicates a request to delete an existing mapping.

The PCP server needs to remember N PORT_SET_Nonces, in which N SHOULD not be larger than $\text{floor}(\text{MAX_USER_QUOTA}/\text{MAX_REQUEST_QUOTA})$. In order to simplify the implementation, it is recommended that N is equal to ONE so that only one MAP_PORT_SET assignment request is permitted for each subscriber. This policy SHOULD be configurable.

It is possible that a mapping might already exist for a requested Internal address (derived from client's IP address). If so, the PCP server MUST take the following actions:

If the suggested External address and port-set in request packet matches the mapping record (including the Internal address, assigned External address, and the port-set), and the existing mapping is dynamic (created by a previous MAP_PORT_SET), the PCP server MUST update the lifetime of the existing mapping and return the existing External Address and Port in response.

If the suggested External address and port-set in request packet does not match the mapping record for the client, the PCP server SHOULD check whether the PORT_SET_Nonce in the request has a corresponding mapping. If so, it means that this mapping record is created by previous MAP_PORT_SET and request/response might be

discarded for some reason in transmission. Then 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. The lifetime of the existing dynamic mapping MUST be updated.

If there is no mapping record in PCP server for the particular PORT_SET_Nonce of MAP_PORT_SET request, it means that the client requires for another delegated set of ports using a new MAP_PORT_SET request. In this case, the PCP server SHOULD check whether the amount of current allocated ports for the client is less than the MAX_USER_QUOTA, and SHOULD assign a new mapping if it does not reach the MAX_USER_QUOTA and there is no PREFER_FAILURE Option in packet. It is highly suggested that the same external IP address should be assigned for the same subscriber.

If no mapping exists for the requested Internal address (derived from client's IP address), and the PCP server is able to create a mapping using the suggested External Address and Port-set, it Should do so. This is beneficial for re-establishing state lost in the PCP server. If the PCP server cannot assign the Suggested External Address and Port-set but can assign some other External Address and Port-set (and the request did not contain the PREFER_FAILURE Option) the PCP server MUST do so and return newly assigned External Address and Port-set in response.

If the MAP request contains the PREFER_FAILURE Option, but the Suggested External Address and Port is not available, the PCP server MUST return CANNOT_PROVIDE_EXTERNAL.

If the PCP server supports both MAP and MAP_PORT_SET Opcode, the server SHOULD check whether the assigned external address is exactly the same with the one for MAP_PORT_SET, and the external port for MAP is within the range of the port-set for MAP_PORT_SET. Otherwise, the PCP server MUST return NO_RESOURCES.

[Discussion: Should we support MAP_PORT_SET and MAP co-existence scenario? Normally, the PCP server for MAP_PORT_SET will not run NAT. And so, there is no NAT binding in PCP.]

[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?]

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. Author List

The following are extended authors who contributed to the effort:

Yunqing Chen

China Telecom

Room 502, No.118, Xizhimennei Street

Beijing 100035

P.R.China

Chongfeng Xie

China Telecom

Room 502, No.118, Xizhimennei Street

Beijing 100035

P.R.China

Yong Cui

Tsinghua University

Beijing 100084

P.R.China

Phone: +86-10-62603059

Email: yong@csnet1.cs.tsinghua.edu.cn

Qi Sun

Tsinghua University

Beijing 100084

P.R.China

Phone: +86-10-62785822

Email: sunqibupt@gmail.com

6. Contributor List

Gabor Bajko

Nokia

Email: gabor.bajko@nokia.com

7. References

7.1. 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,
<<http://datatracker.ietf.org/doc/rfc6431/>>.

7.2. informative References

[I-D.cui-softwire-b4-translated-ds-lite]
Cui, Y., Sun, Q., Boucadair, M., Tsou, T., and Y. Lee,
"Lightweight 4over6: An Extension to DS-Lite
Architecture", Feb 2012.

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

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

