Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: October 27, 2012

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

April 25, 2012

Some Extensions to Port Control Protocol (PCP)
draft-boucadair-pcp-extensions-03

Abstract

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

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

Status of this Memo

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

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

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

This Internet-Draft will expire on October 27, 2012.

Copyright Notice

Table of Contents

1. Introduction

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

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

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

2. DESCRIPTION

This option (Code TBA, Figure 1) MAY be included in a PCP MAP request to include a description associated with a requested mapping. This option is optional to be supported by PCP Servers and PCP Clients. The maximum length SHOULD be a configurable option in the PCP Server. If a PCP Client includes a Description PCP option with a length exceeding the maximum length supported by the PCP Server, only the portion of the Description field fitting that maximum length is stored by the PCP Server.

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

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

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

This Option:

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

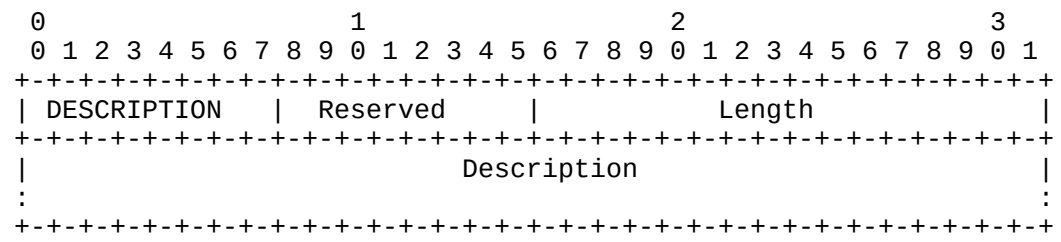


Figure 1: Description Option

3. DSCP_POLICY

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

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

This option is mandatory to process.

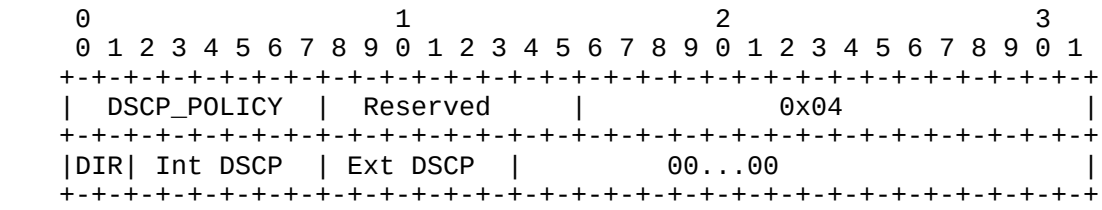
This option (Code TBA, Figure 2) allows to:

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

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

This Option:

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



DIR : Indicates the direction:
0 Inbound
1 Outbound
2 Both
Int DSCP: Indicates the DSCP value in the customer-faced interface. 0x3F is used to indicate ANY value.
Ext DSCP: Indicates the DSCP value in the Internet-faced interface. 0x3F is used to indicate ANY value.

Figure 2: DSCP Marking option

4. CAPABILITY

The CAPABILITY option (Code: TBA, Figure 3) is used by a PCP Server to indicate to a requesting PCP Client the capabilities it supports with regards to port forwarding operations. Several Capability options MAY be conveyed in the same PCP response message if several functions are co-located in the same PCP-controlled device (e.g., NAT44 and NAT64, NAT44 and ports set assignment capability, etc.).

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

This Option:

Option Name: PCP Capabilities Option (CAPABILITY)
Number: TBA (IANA)
Purpose: Retrieve the capabilities of a PCP-controlled device
Valid for Opcodes: can be returned in a error message
Length: 0x01
May appear in: both request and response
Maximum occurrences: None

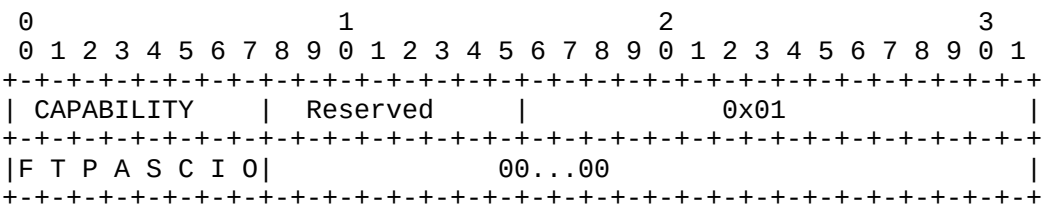


Figure 3: Capability option

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

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

The value of the F, T, P, A, S, C, I and O bits are as follows:

Position	Name	Meaning
1	From (F)	0=from IPv4, 1=from IPv6
2	To (T)	0=to IPv4, 1=to IPv6
3	Port-Xlate (P)	1=translated, 0=not translated
4	Addr-Xlate (A)	1=translated, 0=not translated
5	Port-Set (S)	1=enabled, 0=not supported
6	Packet-Control (C)	1=enabled, 0=not supported
7	Direction-Out (I)	1=enabled, 0=disabled
8	Direction-In (O)	1=enabled, 0=disabled

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

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

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

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

A NAT44 would be characterized as follows:

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

5. REPORT

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

This option can be used for troubleshooting or diagnose purposes.

This Option:

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

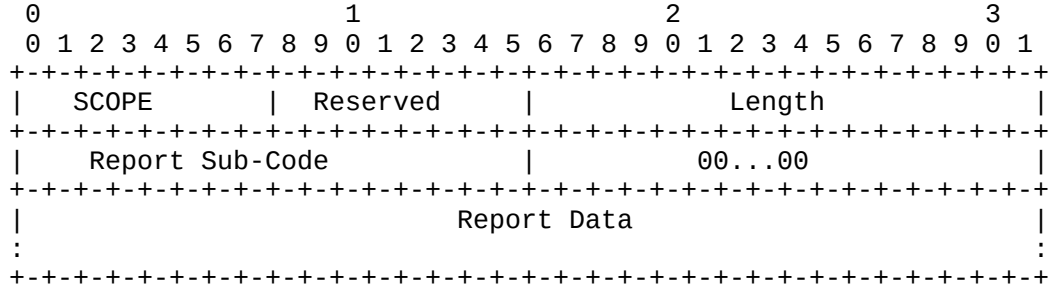


Figure 4: Report Option

The following Report Sub-Code values are defined:

Position	Meaning
0x00	Time since last reboot/boot
0x01	Count of transmitted PCP messages to the PCP Server since last boot
0x02	Count of retransmitted PCP messages to the PCP Server since last boot
0x03	Count of received PCP Error messages from the PCP Server

6. CLIENT_IDENTIFIER

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

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

This Option:

Option Name: PCP Client Identifier Option (CLIENT_ID)
Number: TBA (IANA); mandatory to process option
Purpose: Associate an identifier with the mappings
Valid for Opcodes: MAP
Length: Variable
May appear in: both request and response
Maximum occurrences: 1

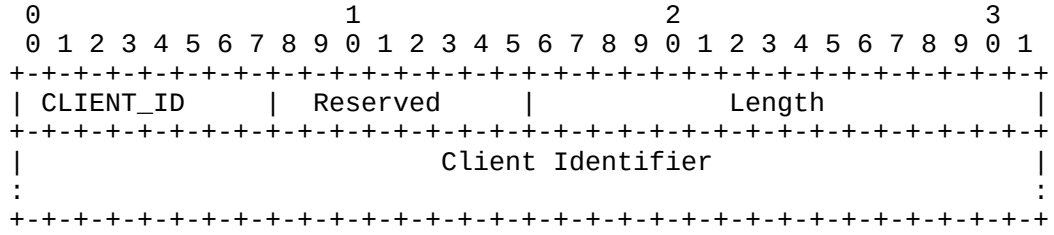


Figure 5: CLIENT_ID PCP Option

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

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

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

7. Security Considerations

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

8. IANA Considerations

The following PCP Option Codes are to be allocated:

DESCRIPTION

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

CAPABILITY

REPORT

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

9. Normative References

[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.

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

[RFC4086] Eastlake, D., Schiller, J., and S. Crocker, "Randomness Requirements for Security", BCP 106, RFC 4086, June 2005.

[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.

Authors' Addresses

Mohamed Boucadair
France Telecom
Rennes, 35000
France

Email: mohamed.boucadair@orange.com

Reinaldo Penno
Cisco
USA

Email: repenno@cisco.com

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

Email: dwing@cisco.com

PCP Working Group
Internet-Draft
Intended status: Informational
Expires: November 17, 2013

M. Boucadair
France Telecom
R. Penno
Cisco

May 16, 2013

Analysis of Port Control Protocol (PCP) Failure Scenarios
draft-boucadair-pcp-failure-06

Abstract

This document identifies and analyzes several PCP failure scenarios. Identifying these failure scenarios is useful to assess the efficiency of the protocol and also to decide whether new PCP extensions are needed.

Status of This Memo

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

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

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

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

Copyright Notice

Table of Contents

1. Introduction

This document discusses several failure scenarios that may occur when deploying PCP [RFC6887].

2. PCP Client Failure Scenarios

2.1. Change of the IP Address of The PCP Server

When a new IP address is used to reach its PCP Server, the PCP Client must re-create all of its explicit dynamic mappings using the newly discovered IP address.

The PCP Client must undertake the same process as per refreshing an existing explicit dynamic mapping (see [RFC6887]); the only difference is the PCP requests are sent to a distinct IP address. No specific behavior is required from the PCP Server for handling these requests.

> Proposed Action: No particular extension is required to be added to the base specification to mitigate this failure scenario.

2.2. Application Crash

When a fatal error is encountered by an application relying on PCP to open explicit dynamic mappings on an upstream device, and upon the restart of that application, the PCP Client should issue appropriate requests to refresh the explicit dynamic mappings of that application (e.g., clear old mappings and install new ones using the new port number used by the application).

If the same port number is used but a distinct Mapping Nonce is generated, the request will be rejected with a NOT_AUTHORIZED error with the Lifetime of the error indicating duration of that existing mapping (see Section 2.7 of [I-D.boucadair-pcp-flow-examples]).

> Proposed Action: A solution to recover the Mapping Nonce used when instantiating the mapping may be envisaged; this solution may not be viable if PCP authentication is not in use. Mapping Nonce recovery in the simple PCP threat model (especially when Mapping Check validation is enabled) induces the same security threated as those discussed in [RFC6887].

If a distinct port number is used by the application to bound its service (i.e., a new internal port number is to be signaled in PCP), the PCP Server may honor the refresh requests if the per-subscriber quota is not exceeded. A distinct external port number would be assigned by the PCP Server due to the presence of "stale" explicit dynamic mapping(s) associated with the "old" port number.

> Proposed Action: To avoid this inconvenience induced by stale explicit dynamic mappings, the PCP Client may clear the "old" mappings before issuing the refresh requests; but this would require the PCP Client to store the information about the "old" port number. This can be easy to solve if the PCP Client is embedded in the application. In some scenarios, this is not so easy because the PCP Client may handle PCP requests on behalf of several applications and no means to identify the requesting application may be supported. Means to identify the application may be envisaged.

[RFC6887] does not allow a PCP Client to issue a request to delete all the explicit dynamic mappings associated with an internal IP address. If a PCP Client is allowed to clear all mappings bound to the same IP address, this would have negative impact on other applications and PCP Client(s) which may use the same internal IP address to instruct their explicit dynamic mappings in the PCP Server.

2.3. PCP Client Crash

The PCP Client may encounter a fatal error leading to its restart. In such case, the internal IP address and port numbers used by requesting applications are not impacted. Therefore, the explicit dynamic mappings as maintained by the PCP Server are accurate and there is no need to refresh them.

On the PCP Client side, a new UDP port should be assigned to issue PCP requests. As a consequence, if outstanding requests have been sent to the PCP Server, the responses are likely to be lost.

If the PCP Client stores its explicit dynamic mappings in a persistent memory, there is no need to retrieve the list of active mappings from the PCP Server.

Proposed Action: If several PCP Clients are co-located on the same host, related PCP mapping tables should be uniquely distinguished (e.g., a PCP Client does not delete explicit dynamic mappings instructed by another PCP Client).

If the PCP Client does not store the explicit dynamic mappings and new Mapping Nonces are assigned, the PCP Server will reject to refresh these mappings.

Proposed Action: This issue can be solved if the PCP Client uses GET OpCode (Appendix B) to recover the mapping nonces used when instantiating the mappings if PCP authentication is used or Mapping Nonce validation check is disabled.

2.4. Change of the Internal IP Address

When a new IP address is assigned to a host embedding a PCP Client, the PCP Client must install on the PCP Server all the explicit dynamic mappings it manages, using the new assigned IP address as the internal IP address. The hinted external port number won’t be assigned by the PCP Server since a "stale" mapping is already instantiated by the PCP Server (but it is associated with a distinct internal IP address).

For a host configured with several addresses, the PCP Client must maintain a record about the target IP address it used when issuing its PCP requests. If no record is maintained and upon a change of the IP address or de-activation of an interface, the PCP-instructed explicit dynamic mappings are broken and inbound communications will fail to be delivered.

Depending on the configured policies, the PCP Server may honor all or part of the requests received from the PCP Client. Upon receipt of the response from the PCP Server, the PCP Client must update its local PCP state with the new assigned port numbers and external IP address.

> Proposed Action: Because of the possible negative impact if the quota is exceed due to the presence of stale mappings (see the example in Section 2.14 of [I-D.boucadair-pcp-flow-examples]), a procedure to clear stale mappings may have some benefits.

A PCP Client may be used to manage explicit dynamic mappings on behalf of a third party (i.e., the PCP Client and the third party are not co-located on the same host). If a new internal IP address is assigned to that third party (e.g., webcam), the PCP Client should be instructed to delete the old mapping(s) and create new one(s) using the new assigned internal IP address. When the PCP Client is co-located with the DHCP server (e.g., PCP Proxy [I-D.ietf-pcp-proxy], IWF in the CP router [I-D.ietf-pcp-upnp-igd-interworking]), the state can be updated using the state of the local DHCP server. Otherwise, it is safe to recommend the use of static internal IP addresses if PCP is used to configure third-party explicit dynamic mappings.

> Proposed Action: No particular extension is required to be added to the base specification to mitigate this failure scenario.

2.5. Change of the CPE WAN IP Address

The change of the IP address of the WAN interface of the CPE would have an impact on the accuracy of the explicit dynamic mappings instantiated in the PCP Server:

- For the DS-Lite case [RFC6333]: if a new IPv6 address is used by the B4 element when encapsulating IPv4 packets in IPv6 ones, the explicit dynamic mappings should be refreshed: If the PCP Client is embedded in the B4, the refresh operation is triggered by the change of the B4 IPv6 address. This would be more complicated when the PCP Client is located in a device behind the B4. If a PCP Proxy is embedded in the CPE, the proxy can use ANNOUNCE OpCode towards internal IPv4 hosts behind the DS-Lite CPE.

- For the NAT64 case [RFC6146], any change of the assigned IPv6 prefix delegated to the CPE will be detected by the PCP Client (because this leads to the allocation of a new IPv6 address). The PCP Client has to undertake the operation described in Section 2.4.

- For the NAT444 case, similar problems are encountered because the PCP Client has no reasonable way to detect the CPE's WAN address changed.

  Proposed Action: Means to help detecting the CPE's WAN address change would help in mitigating this failure scenario.

2.6. UPnP IGD/PCP IWF

In the event an UPnP IGD/PCP IWF [I-D.ietf-pcp-upnp-igd-interworking] fails to renew a mapping, there is no mechanism to inform the UPnP Control Point about this failure.

Proposed Action: This issue can not be solved.

On the reboot of the IWF, if no mapping table is maintained in a permanent storage, "stale" mappings will be maintained by the PCP Server and per-user quota will be consumed. This is even exacerbated if new mapping nonces are assigned by the IWF.

Proposed Action: This issue can be soften by synchronizing the mapping table owing to the invocation of the GET OpCode defined in Appendix B. This procedure is supported only if Mapping Nonce validation checks are disabled.

3. Restart or Failure of the PCP Server

This section covers failure scenarios encountered by the PCP Server.

3.1. Basic Rule

In any situation the PCP Server loses all or part of its PCP state, the Epoch value must be reset when replying to received requests. Doing so would allow PCP Client to audit its explicit dynamic mapping table.

If the state is not lost, the PCP Server must not reset the Epoch value returned to requesting PCP Clients.

Proposed Action: No action is required to update the base PCP specification for this failure scenario.

3.2. Clear PCP Mappings

When a command line or a configuration change is enforced to clear all or a subset of PCP explicit dynamic mappings maintained by the PCP Server, the PCP Server must reset its Epoch to zero value.

In order to avoid all PCP Clients to update their explicit dynamic mappings, the PCP Server should reset the Epoch to zero value only for impacted users.

> Proposed Action: No action is required to update the base PCP specification for this failure scenario.

3.3. State Redundancy is Enabled

When state redundancy is enabled, the state is not lost during failure events. Failures are therefore transparent to requesting PCP Clients. When a backup device takes over, Epoch must not be reset to zero.

> Proposed Action: No action is required to update the base PCP specification for this failure scenario.

3.4. Cold-Standby without State Redundancy

In this section we assume that a redundancy mechanisms is configured between a primary PCP-controlled device and a backup one but without activating any state synchronization for the PCP-instructed explicit dynamic mappings between the backup and the primary devices.

If the primary PCP-controlled device fails and the backup one takes over, the PCP Server must reset the Epoch to zero value. Doing so would allow PCP Clients to detect the loss of states in the PCP Server and proceed to state synchronization.

> Proposed Action: No action is required to update the base PCP specification for this failure scenario.

3.5. Anycast Redundancy Mode

When an anycast-based mode is deployed (i.e., the same IP address is used to reach several PCP Servers) for redundancy reasons, the change of the PCP Server which handles the requests of a given PCP Client won’t be detected by that PCP Client.

Tweaking the Epoch (Section 8.5 of [RFC6887]) may help to detect the loss of state and therefore to re-create missing explicit dynamic mappings.

Proposed Action: No action is required to update the base PCP specification for this failure scenario.

4. Security Considerations

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

5. IANA Considerations

No action is required from IANA.

6. Acknowledgements

Francis Dupont contributed text to this document. Many thanks to him.

7. References

7.1. Normative References

[I-D.ietf-pcp-proxy]
Boucadair, M., Penno, R., and D. Wing, "Port Control Protocol (PCP) Proxy Function", draft-ietf-pcp-proxy-02 (work in progress), February 2013.

[I-D.ietf-pcp-upnp-igd-interworking]
Boucadair, M., 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-10 (work in progress), April 2013.

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

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

7.2. Informative References

[I-D.boucadair-pcp-flow-examples]
Boucadair, M., "PCP Flow Examples", draft-boucadair-pcp-flow-examples-00 (work in progress), February 2013.

[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.

Appendix A. PCP State Synchronization: Overview

The following sketches the state synchronization logic:

- One element (i.e., PCP Client/host/application, PCP Server, PCP Proxy, PCP IWF) of the chain is REQUIRED to use stable storage
- If the PCP Client (resp., the PCP Server) crashes and restarts it just have to synchronize with the PCP Server (resp., the PCP Client);
- If both crash then one has to use stable storage and we fall back in the previous case as soon as we know which one (the Epoch value gives this information);
- PCP Server -> PCP Client not-disruptive synchronization requires a GET/NEXT mechanism to retrieve the state from the PCP Server; without this mechanism the only way to put the PCP Server in a known state is for the PCP Client to send a delete all request, a clearly disruptive operation.
- PCP Client -> PCP Server synchronization is done by a re-create or refresh of the state. The PCP Client MAY retrieve the PCP Server state in order to prevent stale explicit dynamic mappings.

Appendix B. GET/NEXT Operation

This section defines a new PCP OpCode called GET and its associated Option NEXT.

These PCP Opcode and Option are used by the PCP Client to retrieve an explicit mapping or to walk through the explicit dynamic mapping table maintained by the PCP Server for this subscriber and retrieves a list of explicit dynamic mapping entries it instantiated.

GET can also be used by a NoC to retrieve the list of mappings for a given subscriber.

B.1. OpCode Format

The GET OpCode payload contains a Filter used for explicit dynamic mapping matching: only the explicit dynamic mappings of the subscriber which match the Filter in a request are considered so could be returned in response.

Implementation Note: Some existing implementations use 98 (0x62) codepoint for GET OpCode, 131 for AMBIGUOUS error code, and 131 (0x83) for NEXT Option.

The layout of GET OpCode is shown in Figure 1.

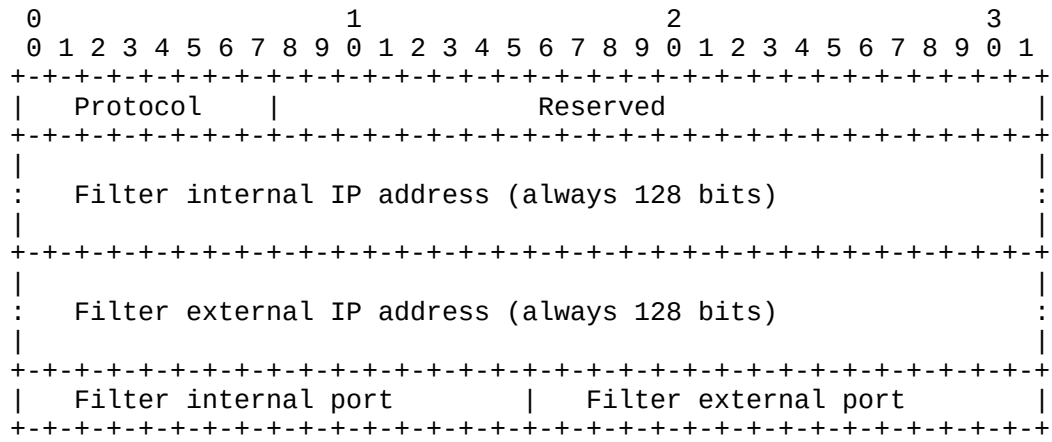


Figure 1: GET: OpCode format

For all fields, the value 0 in a request means wildcard filter/any value matches. Of course this has to be sound: no defined port with protocol set to any.

These fields are described below:

Protocol: Same than for MAP [RFC6887].

Reserved: MUST be sent as 0 and MUST be ignored when received.

Filter internal IP address: Conveys the internal IP address (including an unspecified IPv4IPv6 address). The encoding of this field follows Section 5 of [RFC6887].

Filter external IP address: Conveys the external IP address (including an unspecified IPv4IPv6 address). The encoding of this field follows Section 5 of [RFC6887].

Filter internal port: The internal port (including 0).

Filter external port: The external port (including 0).

Responses include a bit-to-bit copy of the OpCode found in the request.

B.2. OpCode-Specific Result Code

This OpCode defines two new specific Result Code

TBD: NONEXIST_MAP, e.g., no explicit dynamic mapping matching the Filter was found.

TBD: AMBIGUOUS. This code is returned when the PCP Server is not able to decide which mapping to return. Existing implementations use 131 as codepoint.

B.3. Ordering and Equality

The PCP server is required to implement an order between matching explicit dynamic mappings. The only property of this order is to be stable: it doesn't change (*) between two GET requests with the same Filter.

(*) "change" means two mappings are not gratuitously swapped: expiration, renewal or creation are authorized to change the order but they are at least expected by the PCP client.

[Ed. Note: We have two proposals for the order: lexicographical order and lifetime order. Both work, this should be left to the implementor.]

Equality is defined by:

- same protocol and;
- same internal address and;
- same external address and;
- same internal port and;
- same external port.

B.4. NEXT Option

Formal definition:

Name: NEXT

Number: at most one in requests, any in responses.

Purpose: carries a Locator in requests, matching explicit dynamic mappings greater than the Locator in responses.

Is valid for OpCodes: GET OpCode.

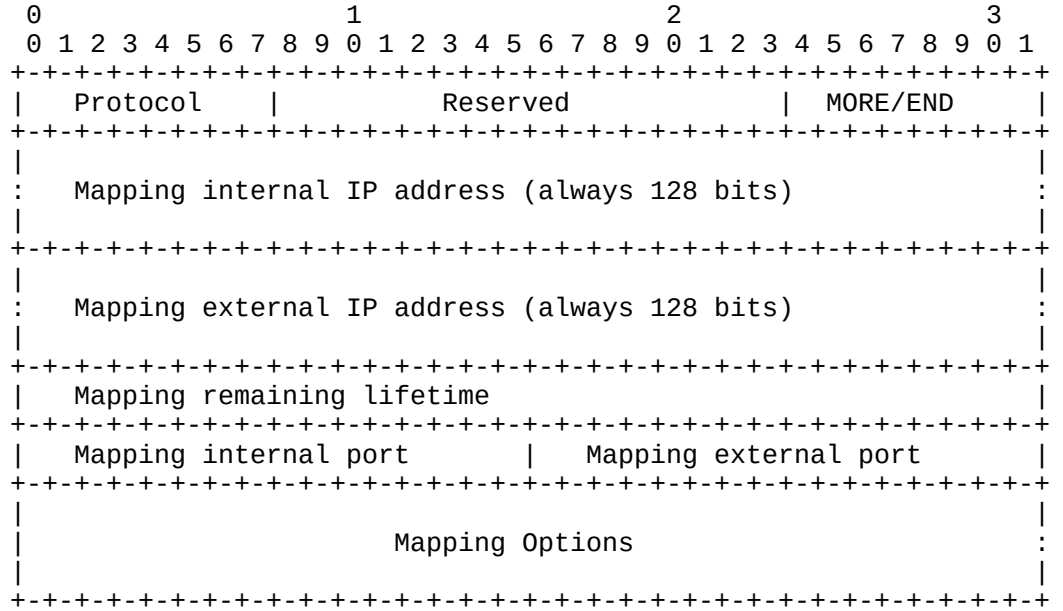
Length: variable, the minimum is 11.

May appear in: both requests and responses.

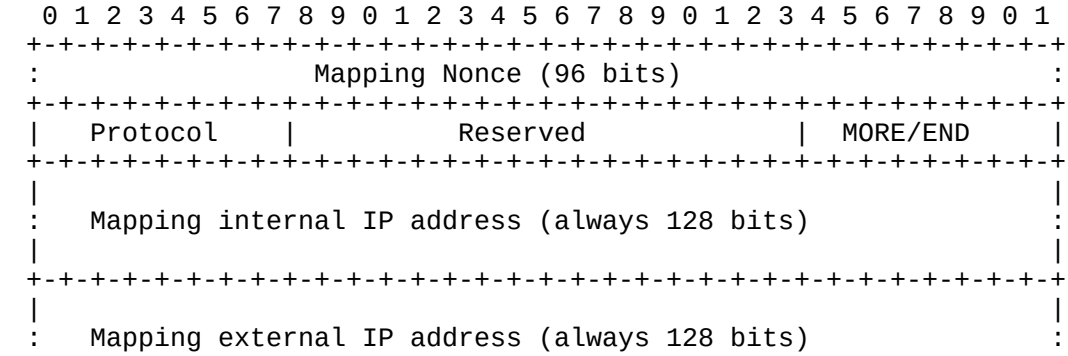
Maximum occurrences: one for requests, bounded by maximum message size for PCP responses [RFC6887].

The layout of the NEXT Option is shown in Figure 2.

Version=1



Version=2



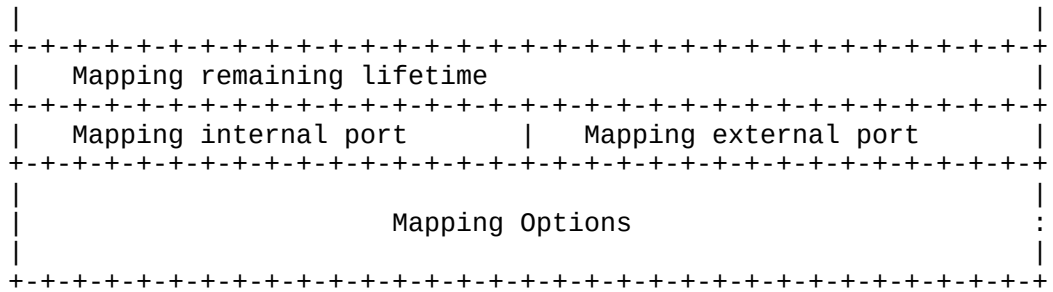


Figure 2: NEXT: Option format

In requests the NEXT Option carries a Locator: a position in the list of explicit dynamic mappings which match the Filter. The following two useful forms of Locators are considered:

- o the "Undefined" form where the Protocol, Addresses, Ports fields are set to zero.
- o the "Defined" form where none of the Protocol, Addresses and Ports is set to zero.

The new fields in a Locator (a.k.a., the NEXT Option in a GET request) are described below:

MORE/END: The value 0 denotes "MORE" and means the response MAY include multiple NEXT Options; a value other than 0 (1 is RECOMMENDED) denotes "END" and means the response SHALL include at most one NEXT Option.

Mapping remaining lifetime: MUST be sent as 0 and MUST be ignored when received.

Mapping Options: The Option Codes of the PCP Client wants to get in the response (e.g., THIRD_PARTY). The format is the same than for the UNPROCESSED Option (see rev 17 of[RFC6887]).

In responses the NEXT Options carry the returned explicit dynamic mappings, one per NEXT Option. The fields are described below:

Protocol: The protocol of the returned mapping.

MORE/END: The value 0 when there are explicit dynamic mapping matching the Filter and greater than this returned mapping; a value other than 0 (1 is RECOMMENDED) when the return mapping is the greatest explicit dynamic mapping matching the Filter.

Mapping internal IP address: the internal address of the returned mapping. The encoding of this field follows Section 5 of [RFC6887].

Mapping external IP address: the external address of the returned mapping. The encoding of this field follows Section 5 of [RFC6887].

Mapping remaining lifetime: The remaining lifetime in seconds of the returned mapping.

Mapping internal port: the internal port of the returned mapping.

Mapping external port: the external port of the returned mapping.

Mapping Options: An embedded list of option values. Each corresponding Option Code MUST be present in the request NEXT Option, each option MUST be related to the returned mapping or not related to any mapping.

B.5. GET/NEXT PCP Client Theory of Operation

GET requests without a NEXT Option have low usage but with a full wildcard Filter they ask the PCP Server to know if it has at least one explicit dynamic mapping for this subscriber.

GET requests with an END NEXT Option are "pure" GET: they asks for the status and/or the remaining lifetime or options of a specific explicit dynamic mapping. It is recommended to use an Undefined Locator and to use the Filter to identify the mapping.

GET requests with a MORE NEXT Option are for the whole explicit dynamic mapping table retrieval from the PCP Server. The initial request contains an Undefined Locator, other requests a Defined Locator filled by a copy of the last returned mapping with the Lifetime and Option fields reseted to the original values. An END NEXT Option marks the end of the retrieval.

B.6. GET/NEXT PCP Server Theory of Operation

The PCP Server behavior is described below:

- on the reception of a valid GET request the ordered list of explicit dynamic mapping of the subscriber matching the given Filter is (conceptually) built.

- if the list is empty a NONEXIST_MAP error response is returned. It includes no NEXT Option.

- o the list is scanned to find the Locator using the Equality defined in Appendix B.3. If it is found the mappings less than the Locator are removed from the list, so the result is a list which begins by the mapping equals to the Locator followed by greater mappings.

- o if the NEXT Option in the request is an END one, the first mapping of the list is returned in an only NEXT option, marked END if the list contains only this mapping, marked MORE otherwise.

- o if the NEXT option in the request is a MORE one, as many as can fit mappings are returned in order in the response, marked as MORE but if the whole list can be returned the last is marked END.

"Returned" means to include required options when they are defined for a mapping: if the mapping M has 3 REMOTE_PEER_FILTERs and the REMOTE_PEER_FILTER code was in the request NEXT, the NEXT carrying M will get the 3 REMOTE_PEER_FILTER options embedded.

B.7. Flow Examples

As an illustration example, let’s consider the following explicit dynamic mapping table is maintained by the PCP Server:

Pro	Internal IP Address	Internal Port	External IP Address	External Port	Remaining Lifetime
UDP	198.51.100.1	25655	192.0.2.1	15659	1659
TCP	198.51.100.2	12354	192.0.2.1	32654	3600
TCP	198.51.100.2	8596	192.0.2.1	25659	6000
UDP	198.51.100.1	19856	192.0.2.1	42654	7200
TCP	198.51.100.1	15775	192.0.2.1	32652	9000

Table 1: Excerpt of a mapping table

As shown in Table 1, the PCP Server sorts the explicit dynamic mapping table using the internal IP address and the remaining lifetime.

Figure 3 illustrates the exchange that occurs when a PCP Client tries to retrieve the information related to a non-existing explicit dynamic mapping.

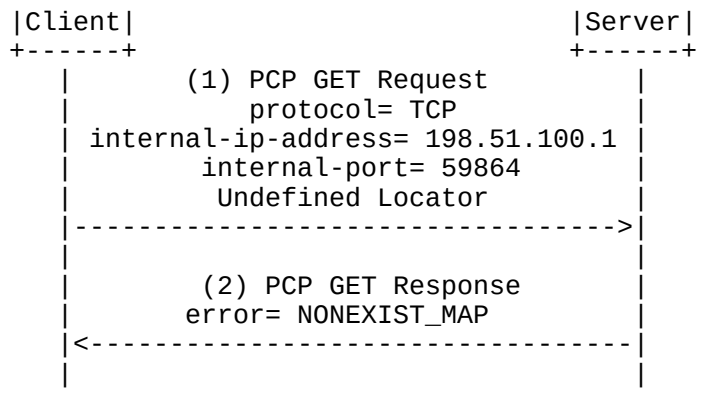
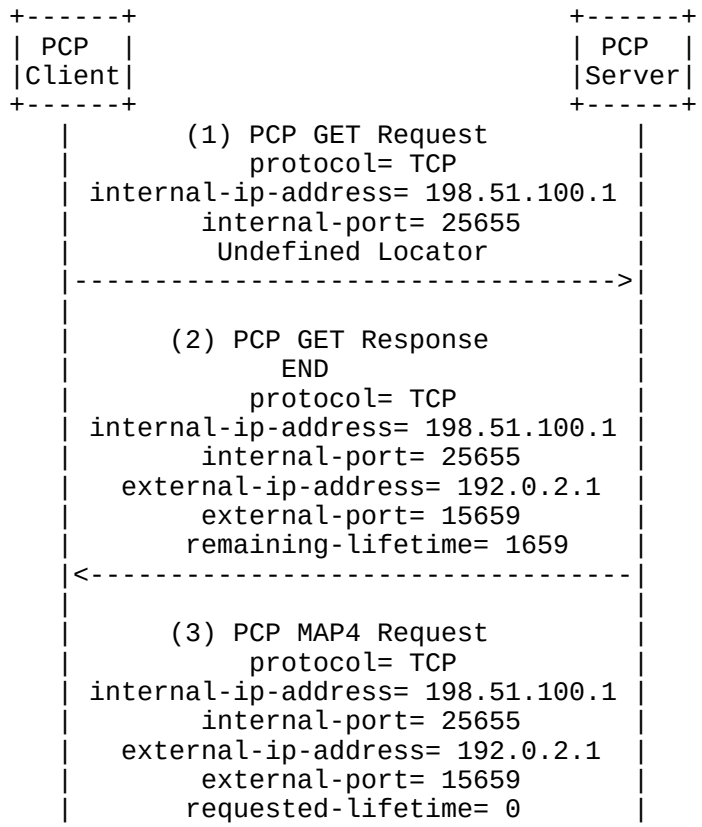


Figure 3: Example of a failed GET operation

Figure 4 shows an example of a PCP Client which retrieves successfully an existing mapping from the PCP Server.



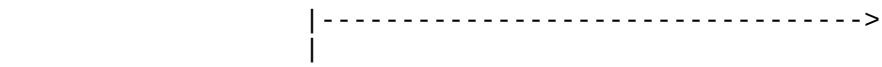


Figure 4: Example of a successful GET operation

In reference to Figure 5, the PCP Server returns the explicit dynamic mappings having the internal address equal to 192.0.2.1 ordered by increasing remaining lifetime.

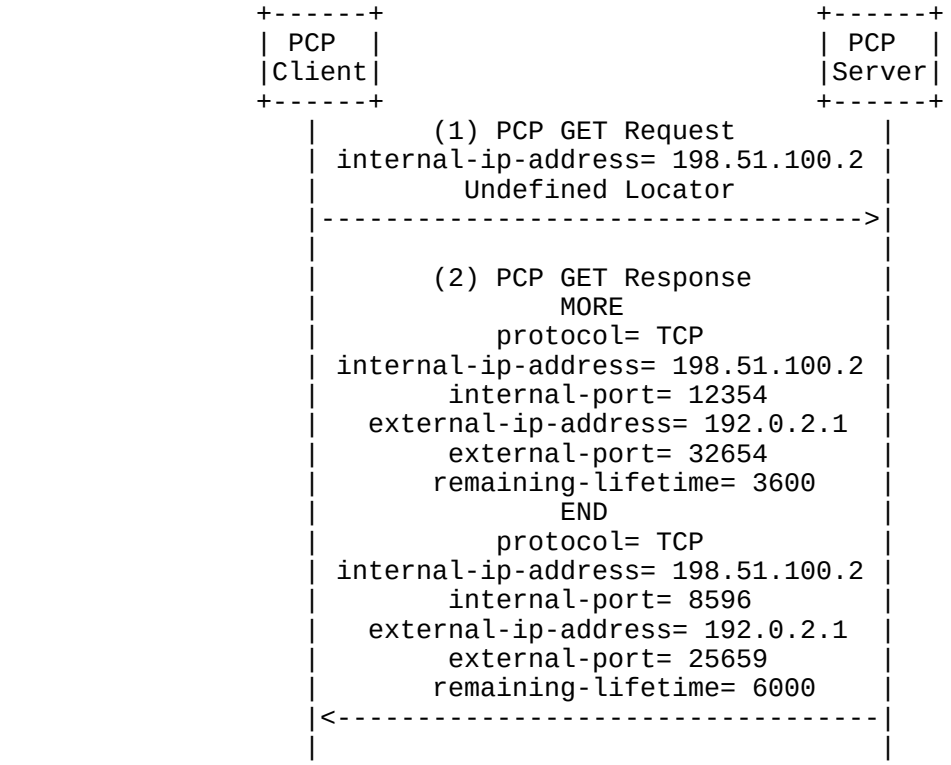
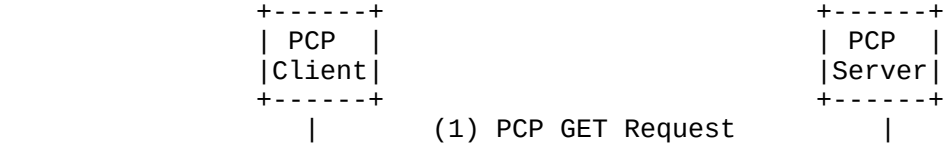


Figure 5: Flow example of GET/NEXT

In reference to Figure 6, the PCP Server returns the explicit dynamic mappings having the internal address equal to 192.0.2.2 ordered by increasing remaining lifetime. In this example, the same internal port is used for TCP and UDP.



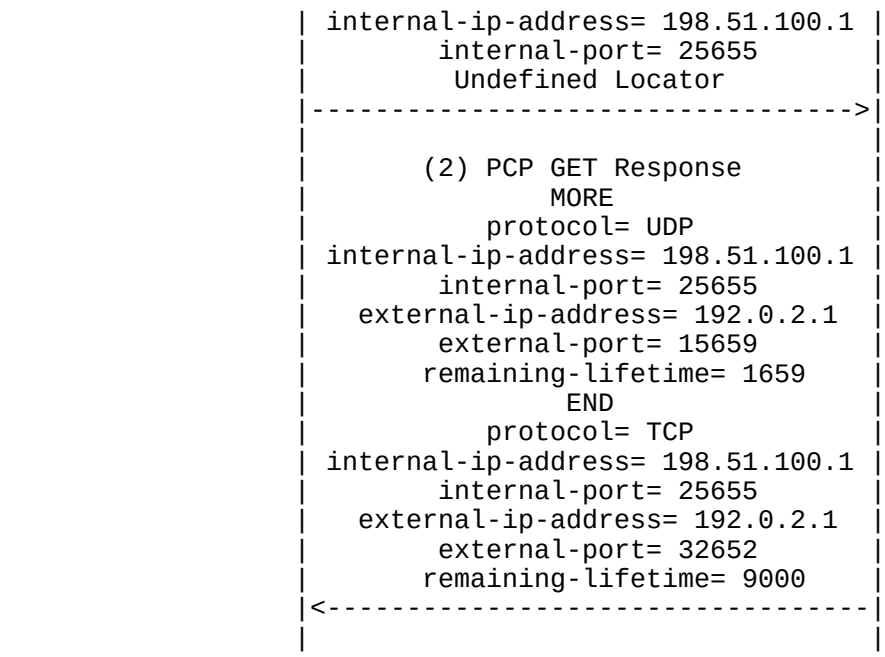


Figure 6: Flow example of GET/NEXT: same internal port number

Authors' Addresses

Mohamed Boucadair
France Telecom
Rennes 35000
France

Email: mohamed.boucadair@orange.com

Reinaldo Penno
Cisco
USA

Email: repenno@cisco.com

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

M. Boucadair
France Telecom
S. Sivakumar
Cisco

October 15, 2012

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

Abstract

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

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

Status of this Memo

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

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

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

This Internet-Draft will expire on April 18, 2013.

Copyright Notice

Table of Contents

1. Introduction

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

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

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

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

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

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

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

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

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

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

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

3. Definition of the Port Reservation Option

3.1. Requirements

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

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

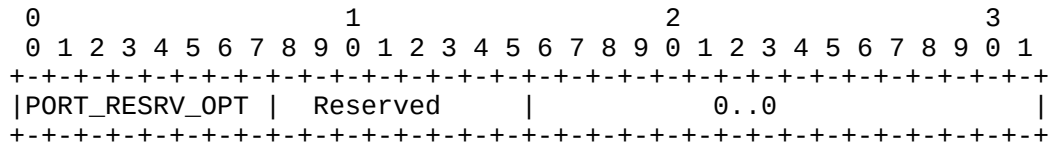
2. Preserve port contiguity as discussed in Section 4.2.3 of [RFC4787] (i.e., RTCP = RTP+1).

3.2. Rationale

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

3.3. PCP Port Reservation Option

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



This Option:

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

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

4. Client Behaviour

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

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

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

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

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

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

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

5. Server Behaviour

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

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

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

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

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

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

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

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

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

6. Illustration Examples

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

6.1. Port Reservation Option Not Supported by The PCP Server

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

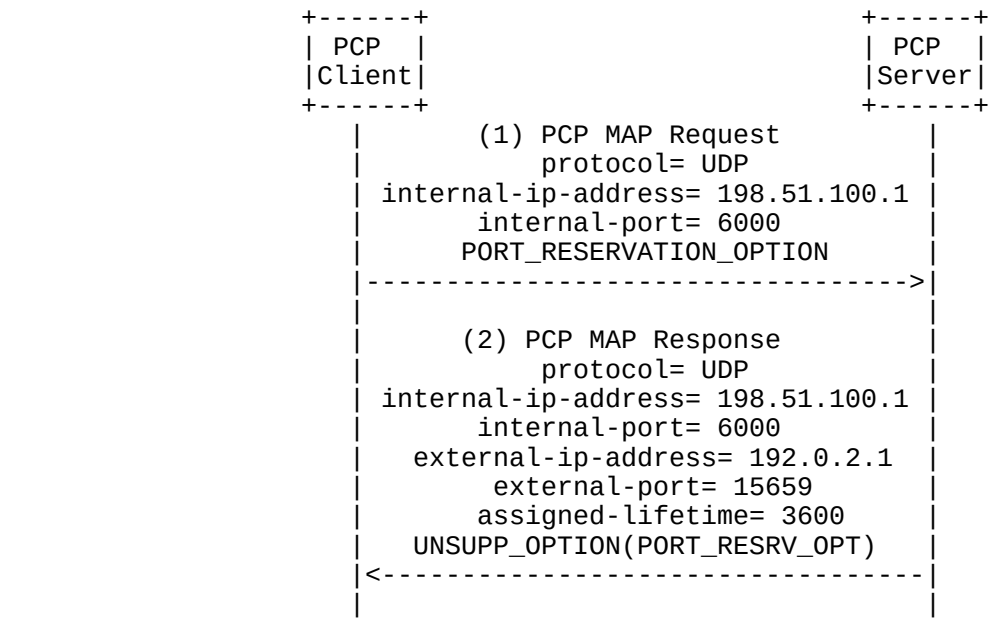


Figure 2: Flow Example of a PCP Server which does not support the Port Reservation Option

6.2. Port Reservation Option Is Supported by The PCP Server

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

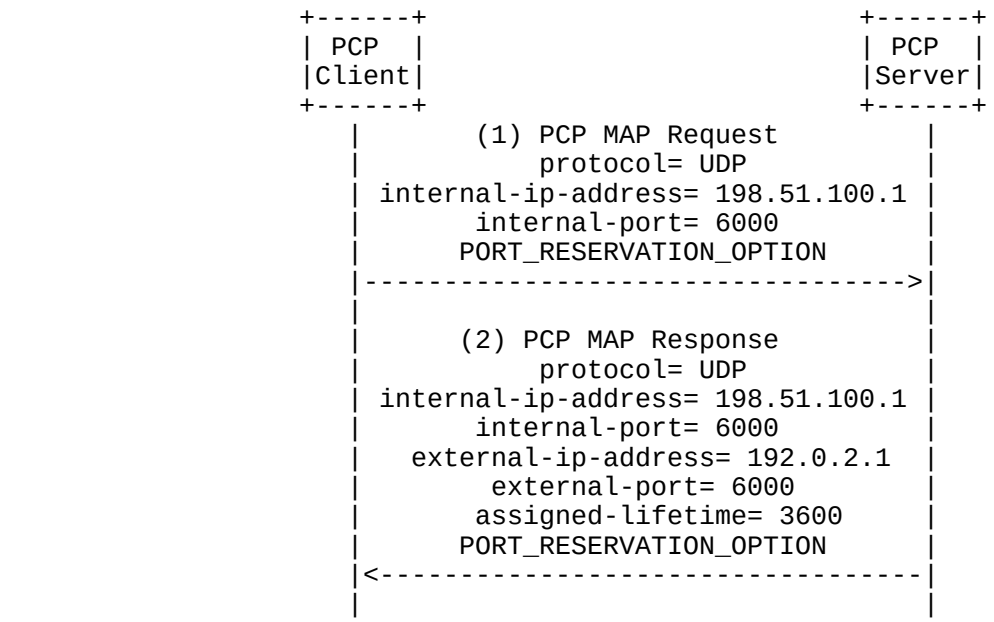


Figure 3: Flow Example of a PCP Server supporting the option and honoring the hinted external port

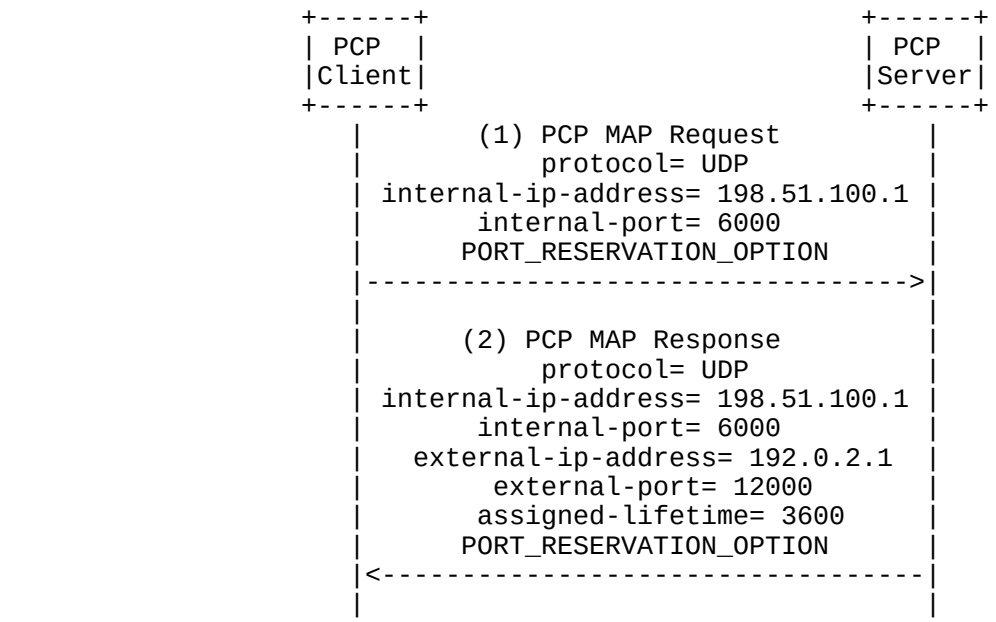


Figure 4: Flow Example of a PCP Server supporting the option but not honoring the hinted external port

6.3. Delete the Mappings

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

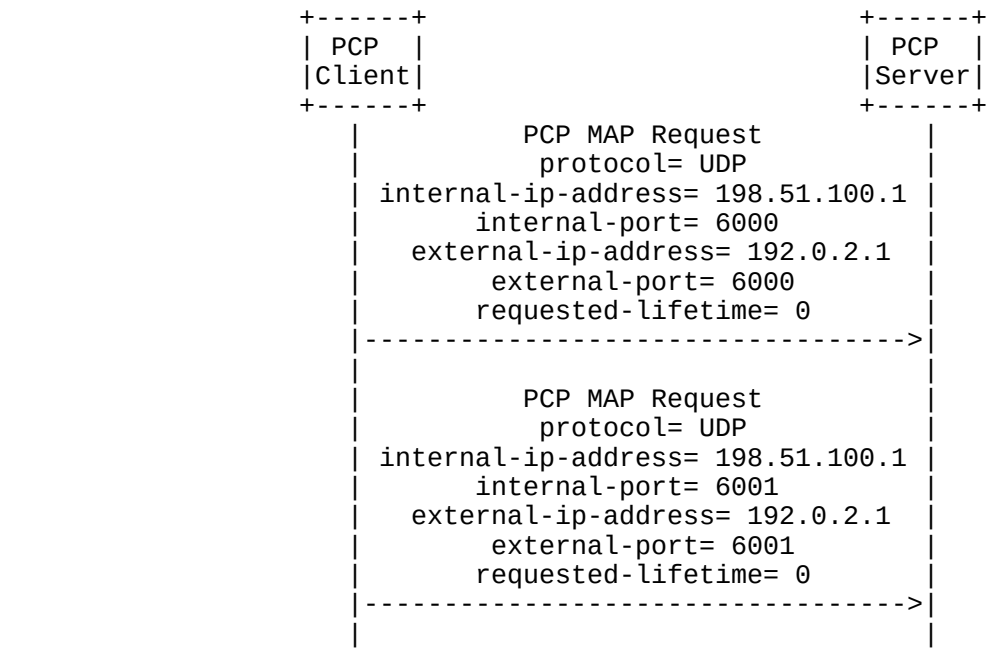


Figure 5: Flow example to delete the mappings

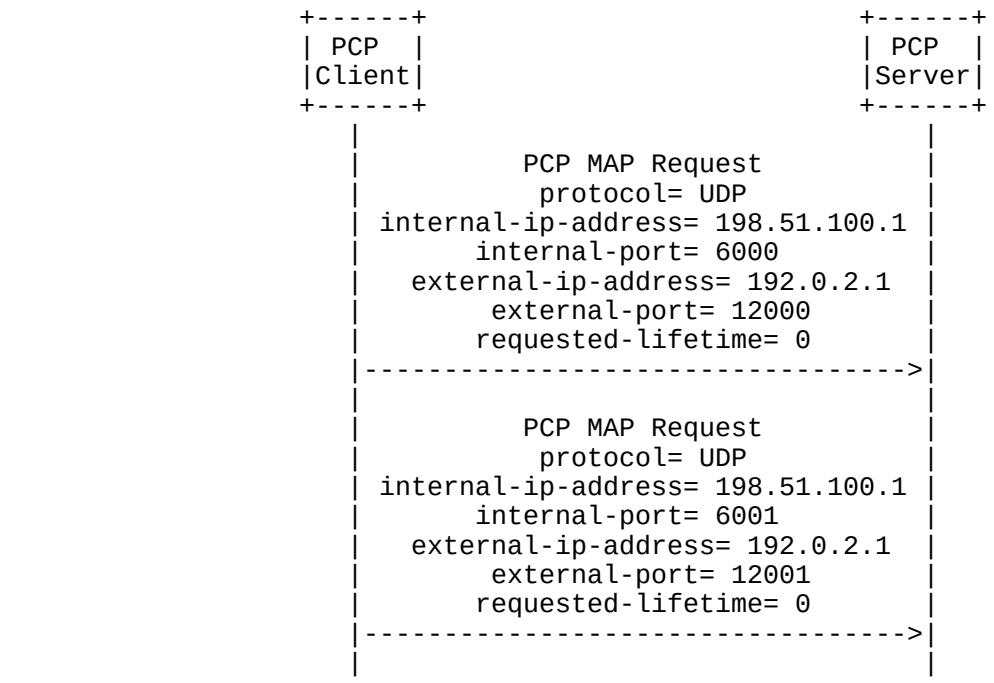


Figure 6: Flow example to delete the mappings (2)

7. IANA Considerations

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

Option Name	Value
PORT_RESERVATION_OPTION	TBA

8. Security Considerations

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

9. Acknowledgments

Many thanks to S. Perrault for his comments.

10. References

10.1. Normative References

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

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

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

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

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

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

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

10.2. Informative References

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

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

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

Authors' Addresses

Mohamed Boucadair
France Telecom
Rennes, 35000
France

Email: mohamed.boucadair@orange.com

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

Email: ssenthil@cisco.com

PCP Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 8, 2011

M. Boucadair
France Telecom
R. Penno
Juniper Networks
D. Wing
Cisco
R. Dupont
Internet Systems Consortium

March 7, 2011

Port Control Protocol (PCP) NAT-PMP Interworking Function
draft-bpw-pcp-nat-pmp-interworking-00

Abstract

This document specifies the behavior of a PCP NAT Port Mapping Protocol (NAT-PMP) Interworking element, for instance embedded in Customer Premise routers.

Status of this Memo

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

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

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

This Internet-Draft will expire on September 8, 2011.

Copyright Notice

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

The NAT Port Mapping Protocol (NAT-PMP [I-D.cheshire-nat-pmp]) provides LAN based NAT control features which are a subset of the new Port Control Protocol (PCP [I-D.ietf-pcp-base]).

This document is about an Interworking Function (IWF) between NAT-PMP clients on internal hosts and a PCP server running on a ISP Carrier-Grade NAT.

Two kinds of IWFs are described:
- Link IWF which serves only clients attached to a LAN
- Home IWF which serves directly or indirectly through Link IWFs all the clients of the Home domain

The Home IWF can be integrated with a UPnP IGD IWF [I-D.bpw-pcp-upnp-igd-interworking] and/or a PCP Proxy [I-D.bpw-pcp-proxy]. Because NAT-PMP does not work through routers, an IWF is REQUIRED to serve any LAN where a NAT-PMP client is attached. A Home IWF is REQUIRED per Home domain where a NAT-PMP client is to be served.

Note the NAT-PMP IWF architecture is closed to the PCP Proxy one so a knowledge of [I-D.bpw-pcp-proxy] is assumed.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

2. TODO

To be filled (imports from UPnP IGD IWF / PCP Proxy)

3. Link IWF

A Link IWF is used to cross routers, i.e., it allows a NAT-PMP client attached to a link where the Home IWF is not connected to get the service.

The Link IWF keeps:
- the IP address of the Home IWF
- a service socket per link where it offers the service
- the source address and port of pending requests
- the operation code of pending requests

Pending requests are expired after a reasonable timeout, e.g., 30 seconds.

NAT-PMP port requests and responses are mapped to PCP MAP4 requests and responses. A THIRD_PARTY option is used to carry the client address.

public address requests and responses are not mapped to PCP messages but are sent to and received from the Home IWF.

4. Result code mapping

PCP result codes and error conditions are mapped to NAT-PMP result codes following this table:

a bad version in NAT-PMP request is mapped to code 1 "Unsupported Version"

a bad opcode in NAT-PMP request is mapped to code 5 "Unsupported Opcode"

to have no external address and similar conditions are mapped to code 3 "Network Failure"

NO_RESOURCES and USER_EX_QUOTA are mapped to code 4 "Out of resources"

NOT_AUTHORIZED is mapped to code 2 "Not Authorized/Refused"

SUCCESS is mapped to code 0 "Success"

[I-D.woodyatt-spnatpmp-appl]

5. Home IWF

At the exception of public address request handling, a Home IWF works as a Smart PCP Proxy. In particular the Epoch handling is a REQUIRED service.

When the Epoch value is reset, a multicast public address announce SHOULD be sent on served links with a multicast capability.

A Home IWF MUST deal with public address request and response internally, i.e., it gets the Epoch value and the external address from its internal state.

The request/response caching and retransmission services SHOULD be supported as the IWF adapts retransmission scheduling between protocols.

6. multicast announces

To be filled.

7. IANA Considerations

This document makes no request of IANA.

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

8. Security Considerations

To be filled.

9. Acknowledgments

To be filled.

10. References

10.1. Normative References

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

[I-D.ietf-pcp-base]
Wing, D., Cheshire, S., Boucadair, M., Penno, R., and F. Dupont, "Port Control Protocol (PCP)", draft-ietf-pcp-base-06 (work in progress), February 2011.

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

10.2. 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-00 (work in progress), March 2011.

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

[I-D.woodyatt-spnatpmp-appl]
Woodyatt, J., "Applicability of NAT-PMP with Service Provider Deployments of Network Address Translation", draft-woodyatt-spnatpmp-appl-01 (work in progress), November 2008.

Authors' Addresses

Mohamed Boucadair
France Telecom
Rennes 35000
France

Email: mohamed.boucadair@orange-ftgroup.com

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

Email: rpenno@juniper.net

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

Email: dwing@cisco.com

Francis Dupont
Internet Systems Consortium

Email: fdupont@isc.org

PCP Working Group
Internet-Draft
Intended status: Standards Track
Expires: March 15, 2012

M. Boucadair
France Telecom
R. Penno
Juniper Networks
D. Wing
Cisco
R. Dupont
Internet Systems Consortium

September 12, 2011

Port Control Protocol (PCP) Proxy Function
draft-bpw-pcp-proxy-02

Abstract

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

Status of this Memo

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

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

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

This Internet-Draft will expire on March 15, 2012.

Copyright Notice

Table of Contents

1. Introduction

PCP [I-D.ietf-pcp-base] discusses the implementation of NAT control features that rely upon Carrier Grade NAT (CGN) devices such as DS-Lite AFTR [RFC6333].

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

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

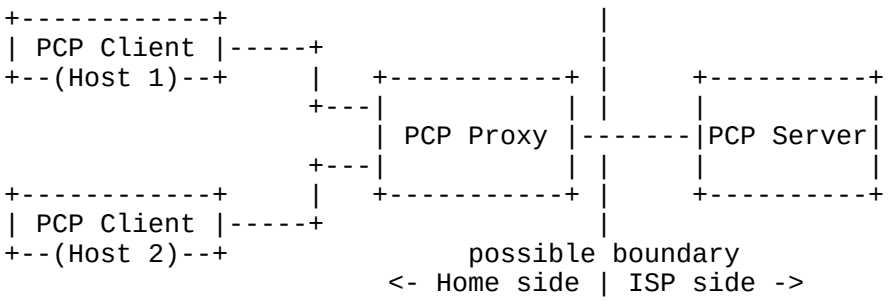


Figure 1: Reference Architecture

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

2. PCP Server Discovery and Provisioning

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

The address of the PCP Proxy is provisioned to local PCP Clients as their default PCP Server: If the PCP DHCP option is supported by an

internal PCP Client, it will retrieve the PCP Server IP address to use from its local DHCP server (usually embedded on the CP router); otherwise internal PCP Clients will assume their default router being the PCP Server.

3. PCP Proxy as a PCP Server

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

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

4. Control of the Firewall

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

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

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

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

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

5. Embedded NAT in the CP Router

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

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

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

- a PCP Proxy co-located with a NAT SHOULD reject by an UNSUPP_OPCODE error response a received request with an unknown OpCode;

- a PCP Proxy co-located with a NAT SHOULD reject by an UNSUPP_OPTION error response a received request with a mandatory-to-process unknown Option;

- a PCP Proxy co-located with a NAT SHOULD remove any optional-to-process unknown Options from received requests before forwarding them.

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

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

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

6. Simple PCP Proxy

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

The detailed behavior at the reception of a PCP request on an internal interface is as follows:

- check if the source IP address and the PCP target address are the same.
- apply security controls, including with the result of the previous item.
- if the request is rejected, build a synthetic error response and send it back to the PCP client.
- if the request is accepted, adjust it (e.g., adding a THIRD_PARTY Option, updating the internal address and port to their translated values as specified in Section 5 and forward it on a fresh UDP socket connected to the PCP server.
- Wait for the response during a reasonable delay.
- when the response is received from the PCP server, adjust it back (e.g., removing the THIRD_PARTY Option added previously, updating the internal address and port to their initial values as specified in Section 5), forward it to the source PCP client and close the socket to the PCP server.

  [Ed. Note: is there extra validation useful? The response comes from the PCP server and the PCP client will validated it anyway.]

- on a hard error on the UDP socket, build a synthetic ICMP error and send it to the source PCP client.

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

For each pending request, the proxy MUST maintain in a data record:

- the request payload
- the interface where the request was received

- the source IP address of the request
- the source UDP port of the request
- the UDP socket connected to the PCP server
- an expire timeout

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

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

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

7. Smart Proxy

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

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

7.1. Multiple PCP Servers

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

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

[Ed. Note: we propose to require a partition of clients, clients on the same host or sharing a target address SHOULD be in the same subset, i.e., the same PCP server and the same Epoch.]

[Ed. Note: the Proxy can get per PCP server capabilities, for instance from the error responses.]

7.2. Epoch Handling

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

The Epoch MUST be reset when explicit dynamic mappings are lost, i.e.:

- at startup if the PCP proxy can’t recover the state.

  [Ed. Note: as it is very optional to manage state in the Proxy it should be the default.]

- when the WAN address is changed or any similar events which show any previous state is no longer valid.

- when the Epoch value in a PCP response is too small (cf. Epoch value validation rules in [I-D.ietf-pcp-base]).

- when the External Address has changed

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

7.3. Request/Response Caching

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

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

7.4. Retransmission Handling

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

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

7.5. Full State

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

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

8. IANA Considerations

This document makes no request of IANA.

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

9. Security Considerations

The security controls are applied on PCP requests and are about:

- authorized target addresses, in particular in case of a third party.
- authorized internal and external ports (note the external port is in general assigned by the CGN PCP server).

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

When a PCP Proxy is at the boundary of two trust domains (named

"internal" and "external" sides), it MUST provide at least these two security controls:

- split horizon anti-spoofing: requests from the external side and responses from the internal side MUST be dropped.
- a policy about requests on the behalf of a third party MUST be enforced.

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

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

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

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

10. References

10.1. Normative References

[I-D.bpw-pcp-dhcp] Boucadair, M., Penno, R., and D. Wing, "DHCP and DHCPv6 Options for the Port Control Protocol (PCP)", draft-bpw-pcp-dhcp-04 (work in progress), April 2011.

[I-D.ietf-pcp-base] Wing, D., Cheshire, S., Boucadair, M., Penno, R., and P. Selkirk, "Port Control Protocol (PCP)", draft-ietf-pcp-base-13 (work in progress), July 2011.

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

10.2. Informative References

[I-D.boucadair-pcp-failure]
Boucadair, M., Dupont, F., and R. Penno, "Port Control Protocol (PCP) Failure Scenarios", draft-boucadair-pcp-failure-01 (work in progress), March 2011.

[I-D.bpw-pcp-nat-pmp-interworking]
Boucadair, M., Penno, R., Wing, D., and F. Dupont, "Port Control Protocol (PCP) NAT-PMP Interworking Function", draft-bpw-pcp-nat-pmp-interworking-00 (work in progress), March 2011.

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

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

Authors' Addresses

Mohamed Boucadair
France Telecom
Rennes 35000
France

Email: mohamed.boucadair@orange-ftgroup.com

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

Email: rpenno@juniper.net

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

Email: dwing@cisco.com

Francis Dupont
Internet Systems Consortium

Email: fdupont@isc.org

PCP Working Group
Internet-Draft
Intended status: Standards Track
Expires: August 11, 2011

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

February 07, 2011

Universal Plug and Play (UPnP) Internet Gateway Device (IGD)-Port Control Protocol (PCP) Interworking Function
draft-bpw-pcp-upnp-igd-interworking-02

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 August 11, 2011.

Copyright Notice

Table of Contents

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 [I-D.ietf-softwire-dual-stack-lite] or NAT64 [I-D.ietf-behave-v6v4-xlate-stateful]. 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 (an example is illustrated in Figure 1).

Two configurations are considered:

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

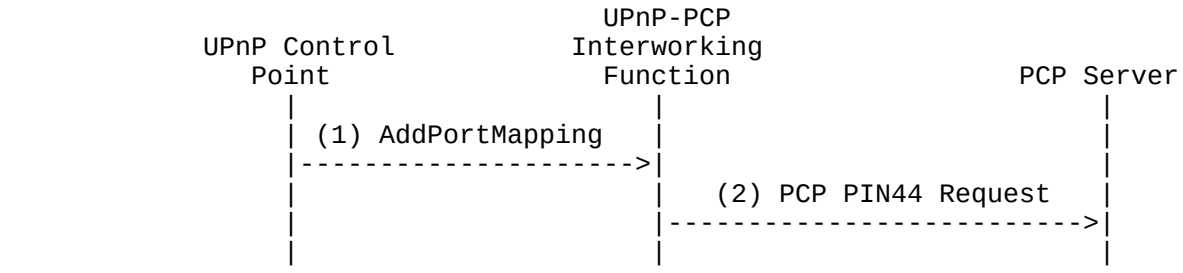


Figure 1: Flow Example

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:

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

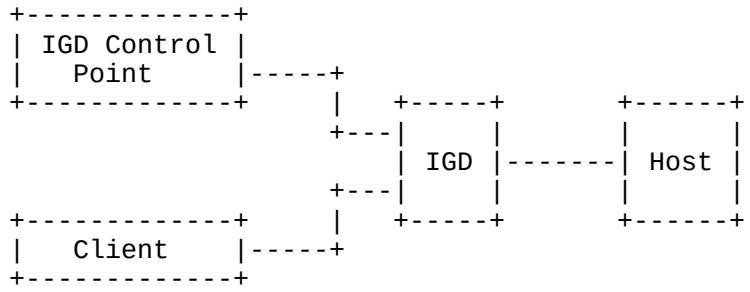


Figure 2: UPnP IGD Model

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.

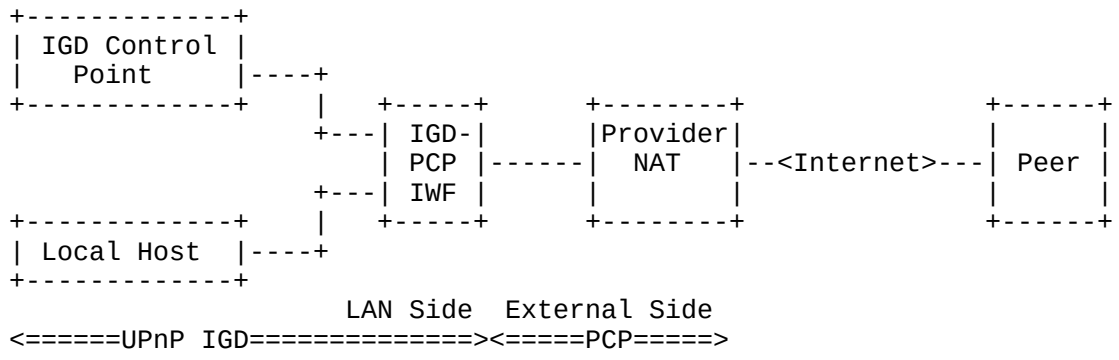


Figure 3: UPnP IGD-PCP Interworking Model

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).

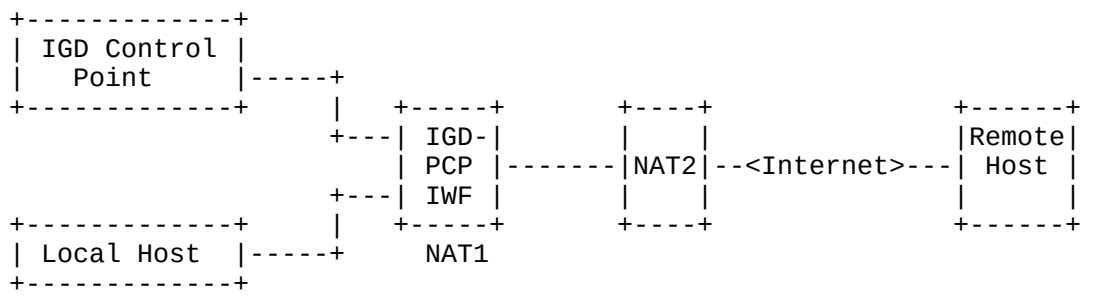


Figure 4: Cascaded NAT scenario

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 successfull communication was established with a PCP Server, or with a state maintained in a stable storage the number of seconds since the initialization of the current state).

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

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

ExternaIPAddress: External IP Address
Read-only variable with the value from the last PCP response or the empty string if none was received yet.

PortMappingNumberOfEntries: Not applicable
Managed locally by the UPnP IGD-PCP Interworking Function.

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

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

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

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

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

PortMappingProtocol: Transport Protocol
Mapped to PCP protocol field in PINxx messages. Note both IGD and PCP only support TCP and UDP.

InternalClient: Internal IP Address
InternalClient can be an IP address or a domain name. Only an IP address scheme is supported in PCP. If a domain name is used Point, it must be resolved to an IP address by the Interworking Function when relying the message to the PCP Server.

PortMappingDescription: Not applicable
Not supported in base PCP. When present in UPnP IGD messages, this parameter SHOULD NOT be propagated in the corresponding PCP messages. If the local PCP Client support a PCP Option to convey the description, this option MAY be used.

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

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

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

4.2. IGD-PCP: Methods

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

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

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

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

ForceTermination: Notapplicable
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: Not applicable
Under normal conditions, the IGD-PCP Interworking Function maintains an updated list of active mapping as instantiated in the PCP Server. The IGD-PCP Interworking Function locally handles this request and provides back the port mapping entry based on the ExternalPort, the PortMappingProtocol, and the RemoteHost. See Section 5.8 for more information.

AddPortMapping: PIN44
We recommend the use of AddAnyPortMapping() instead of AddPortMapping(). Refer to Section 5.7.2.

AddAnyPortMapping (for IGD:2 only): PIN44
No issue is encountered to proxy this request to the PCP Server. Refer to Section 5.7.1 for more details

DeletePortMapping: PIN44 with a requested lifetime set to 0
Refer to Section 5.9.

DeletePortMappingRange (for IGD:2 only): PIN44 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 yet a method for retrieving the external IP address. Issuing PIN44 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

Section 4.3 lists PCP errors codes and the corresponding UPnP IGD ones. Error codes specific to IGD:2 are tagged accordingly.

3 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.

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

5 AMBIGUOUS: IGD:1 718 "ConflictInMappingEntry" / IGD:2 729 "ConflictWithOtherMechanisms"

[[Note: Currently not defined in base PCP.]]

128 UNSUPP_VERSION: 501 "ActionFailed"
Should not happen.

129 UNSUPP_OPCODE: 501 "ActionFailed"
Should not happen.

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

131 MALFORMED_OPTION: 501 "ActionFailed"
Should not happen.

132 UNSPECIFIED_ERROR: 501 "ActionFailed"

150 UNSUPP_PROTOCOL: 501 "ActionFailed"
Should not happen.

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

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

153 CANNOT_HONOR_EXTERNAL_PORT: 718 "ConflictInMappingEntry"

154 UNABLE_TO_DELETE_ALL: Not applicable
Should not happen as all mapped delete operations are for individual mappings.

155 CANNOT_FORWARD_PORT_ZERO: Not applicable
Should not happen: stateless NATs are not supported.

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.bpw-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.

Once the PCP Sever is reachable, the IGD-PCP Interworking Function MUST synchronize its state as specified in Section 5.10.

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.

[[Note: Firewall support is no longer specified in base PCP]]

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. If not, upon reset or reboot, the IGD-PCP Interworking Function SHOULD

synchronise its states as specified in Section 5.10.

Upon receipt of a PCP PIN44 Response from the PCP Server, the IGD-PCP Interworking Function MUST retrieve the enclosed mapping and MUST store it in the local mapping table. The local mapping table is an image of the mapping table as maintained by the PCP Server for a given subscriber.

5.5. Interworking Function Without NAT in the CP Router

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

5.6. NAT Embedded in the CP Router

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

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

- The internal IP address and/or port number as initially positioned by the UPnP Control Point and stored in the CP router NAT are used to update the corresponding fields in received PCP responses.
- The external IP and port number are not altered by the IGD-PCP Interworking Function.
- The NAT mapping entry in the first NAT is updated with the result of PCP request.

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

5.7. Creating a Mapping

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

AddAnyPortMapping() is the RECOMMENDED method.

5.7.1. AddAnyPortMapping()

When an UPnP Control Point issues a AddAnyPortMapping(), this request is received by the UPnP Server. The request is then relayed to the IGD-PCP Interworking Function which generates a PCP PIN44 Request (see Section 4.1 for mapping between WANIPConnection and PCP parameters). Upon receipt of PCP PIN44 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.

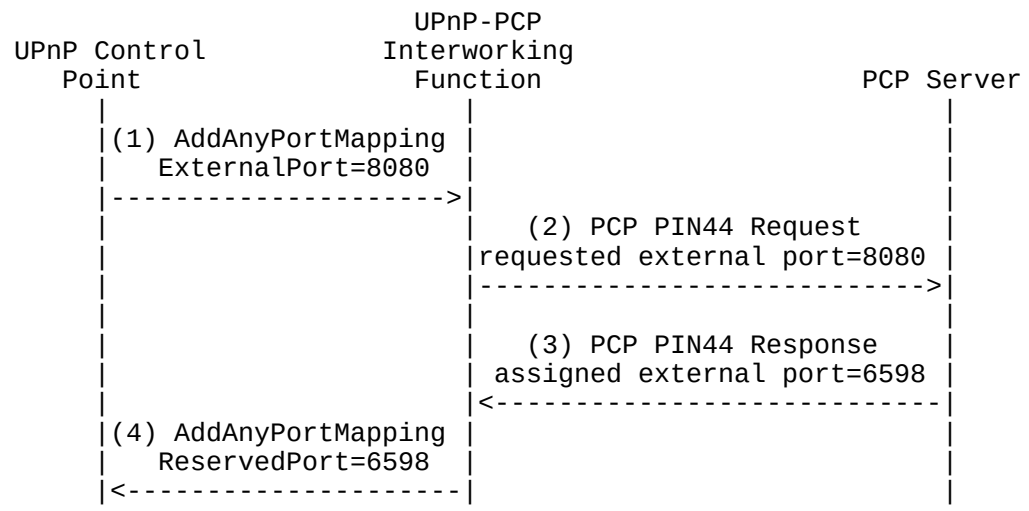


Figure 5: Flow example when AddAnyPortMapping() is used

5.7.2. AddPortMapping()

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

> [[Stateless NAT and stateless-like NAT operations are no clearly defined in base PCP.]]

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 Figure 6).

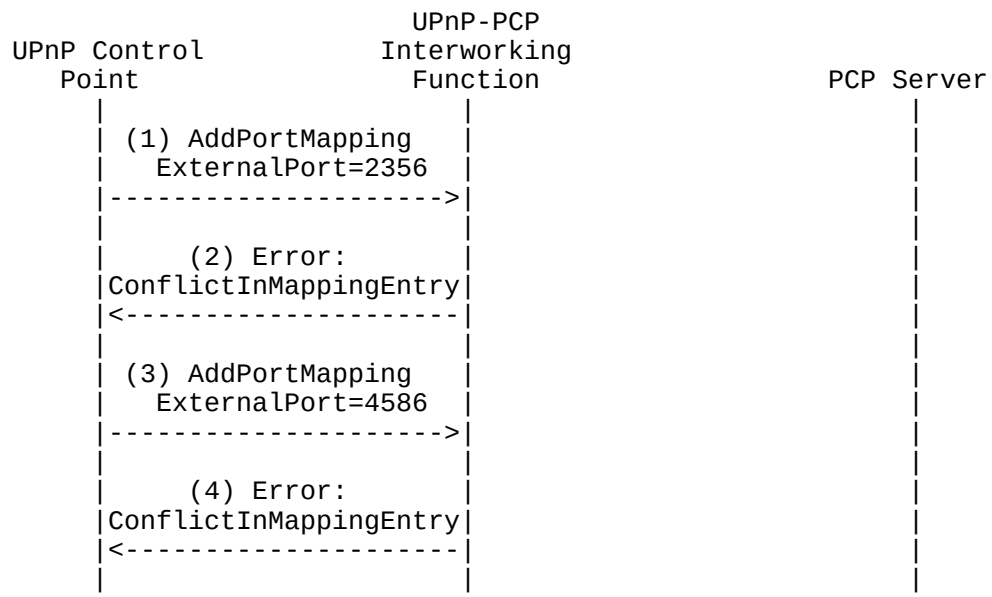


Figure 6: IWF Local Behaviour

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 generates a PCP PIN44 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 inserts a HONOR_EXTERNAL_PORT 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_HONOR_EXTERNAL_PORT as the error cause. The IGD-PCP IWF relays a negative message to the UPnP Control Point indicating ConflictInMappingEntry as error code. The UPnP Control Point re-issues 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.

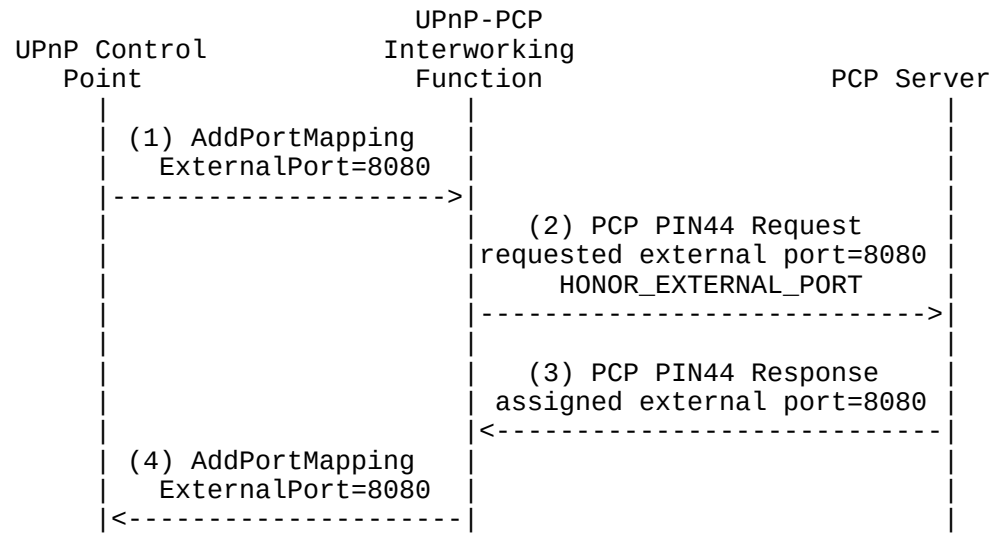


Figure 7: Flow Example (Positive Answer)

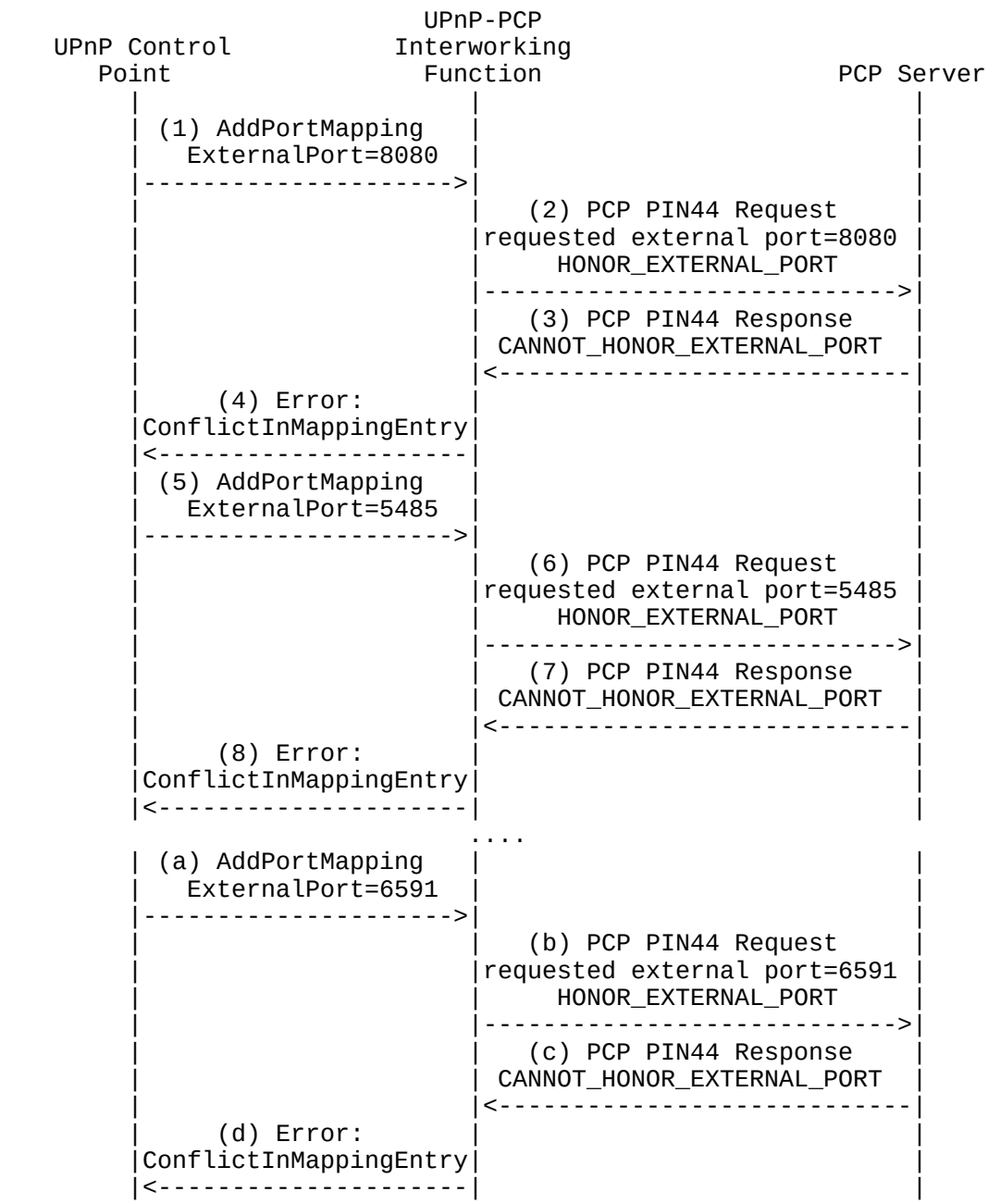


Figure 8: Flow Example (Negative Answer)

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().

These methods MUST NOT be proxied to the PCP Server since a local mapping is maintained by the IGD-PCP Interworking Function.

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.

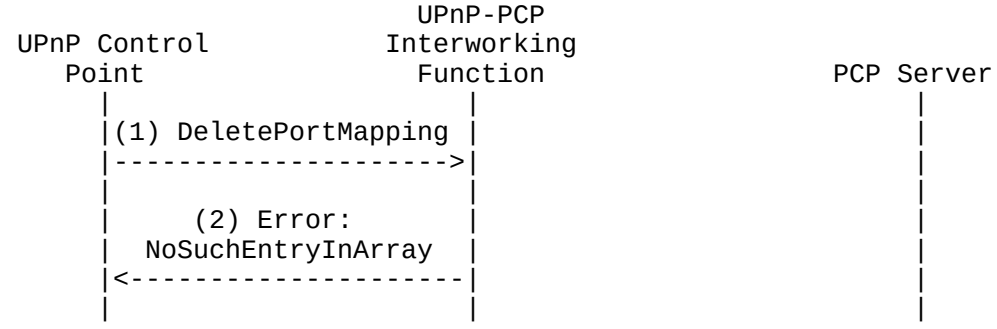


Figure 9: Local Delete (IGD-PCP IWF)

If a mapping matches in the local table, a PCP PIN44 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 PIN44 delete request is received by the PCP Server, it proceeds to removing the corresponding entry. A PCP PIN44 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 PIN44 delete requests corresponding to these mappings (See the example shown in Figure 11).

[[Discussion note: The IWF can 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.]]

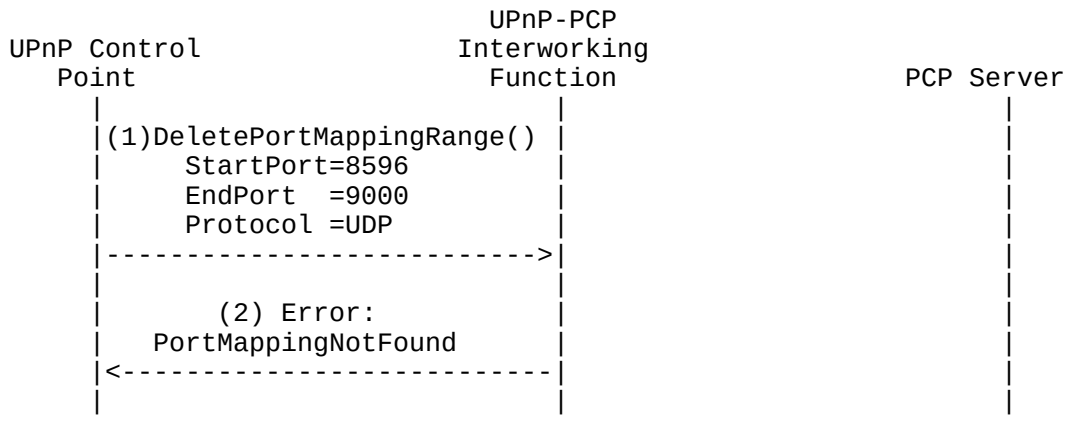


Figure 10: Flow example when an error encountered when processing DeletePortMappingRange()

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 PIN44 requests to delete these mappings.

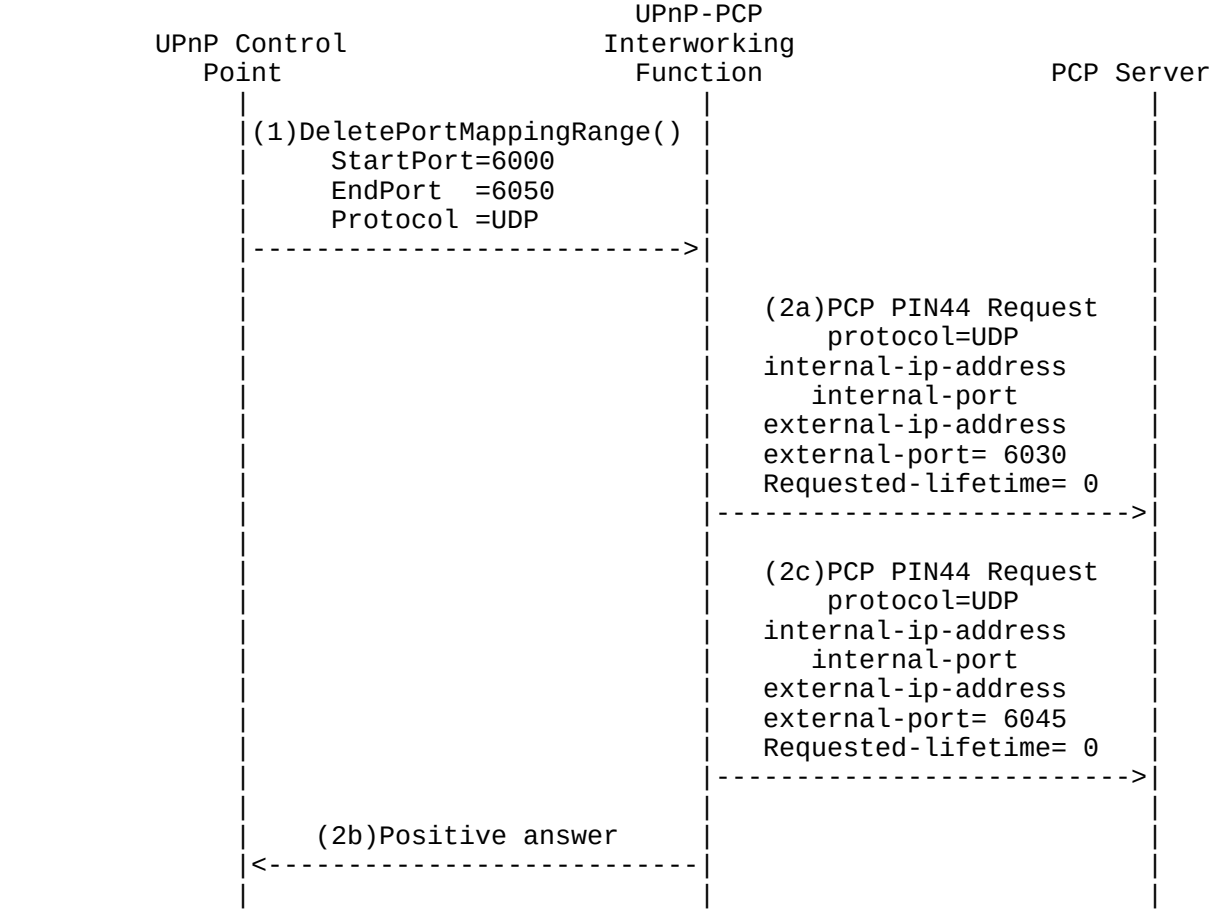


Figure 11: Example of DeletePortMappingRange()

5.10. Mapping Synchronisation

[[Note: This section needs further discussion among authors]]

Under normal conditions, since a valid copy of the mapping table is stored locally in the CP router, the IGD-PCP Interworking Function SHOULD NOT issue any subsequent PCP request to handle a request received from an UPnP Control Point to list active mappings. Nevertheless, in case of loss of synchronisation (e.g., reboot,

system crashes, power outage, etc.), the IGD-PCP Interworking Function SHOULD generate a get method to retrieve all active mappings in the PCP Server and update its local mapping table without waiting for an explicit request from a UPnP Control Point. Doing so, the IGD-PCP Interworking Function maintains an updated mapping table.

In case of massive reboot of CP routers (e.g., avalanche restart phenomenon), PCP request bursts SHOULD be avoided. For this aim, we recommend the use of a given timer denoted as PCP_SERVICE_WAIT. This timer can be pre-configured in the CP router or to be provisioned using a dedicated means such as DHCP. Upon reboot of the CP router, PCP messages SHOULD NOT be sent immediately. A random value is selected between 0 and PCP_SERVICE_WAIT. This value is referred to as RAND(PCP_SERVICE_WAIT). Upon the expiration of RAND(PCP_SERVICE_WAIT), the CP router SHOULD proceed to its synchronisation operations (i.e., retrieve all active mappings which have been instructed by internal UPnP Control Point(s)).

[[Note: per-subscriber quota may be exhausted due to unlimited lifetime and stale mappings in IGD due to reboots, etc.]]

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 and C. Jacquenet for their review and comments.

9. References

9.1. Normative References

[I-D.ietf-pcp-base]
Wing, D., "Port Control Protocol (PCP)", draft-ietf-pcp-base-03 (work in progress), January 2011.

[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.bpw-pcp-dhcp]
Boucadair, M., Penno, R., and D. Wing, "DHCP and DHCPv6 Options for Port Control Protocol (PCP)", draft-bpw-pcp-dhcp-02 (work in progress), January 2011.

[I-D.ietf-behave-v6v4-xlate-stateful]
Bagnulo, M., Matthews, P., and I. Beijnum, "Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers", draft-ietf-behave-v6v4-xlate-stateful-12 (work in progress), July 2010.

[I-D.ietf-softwire-dual-stack-lite]
Durand, A., Droms, R., Woodyatt, J., and Y. Lee, "Dual-Stack Lite Broadband Deployments Following IPv4 Exhaustion", draft-ietf-softwire-dual-stack-lite-06 (work in progress), August 2010.

[IGD2] UPnP Forum, "WANIPConnection:2 Service (http://upnp.org/specs/gw/UPnP-gw-WANIPConnection-v2-Service.pdf)", September 2010.

[Sec_DCP] UPnP Forum, "Device Protection:1", November 2009.

Authors' Addresses

Mohamed Boucadair
France Telecom
Rennes, 35000
France

Email: mohamed.boucadair@orange-ftgroup.com

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

Email: rpenno@juniper.net

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

Email: dwing@cisco.com

Francis Dupont
Internet Systems Consortium

Email: fdupont@isc.org

Internet Engineering Task Force
Internet-Draft
Intended status: Informational
Expires: May 3, 2012

G. Chen
H. Deng
J. Zhang
China Mobile

October 31, 2011

Link-Local Multicast Name Resolution (LLMNR) Deployment Consideration for PCP Server Discovery
draft-chen-pcp-discovery-llmnr-01

Abstract

The memo has recommended to deploy Link-Local Multicast Name Resolution (LLMNR) in PCP scenarioes. Depending on that, it could not only avoid adherence to DNS during PCP server name resolving, but also company with PCP FQDN DHCP options extention to accomplish PCP server discovery. In order to fit LLMNR into PCP network, particular LLMNR deployment guide and relevant configurations are considerated along with PCP elements installation.

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 3, 2012.

Copyright Notice

Table of Contents

1. Introduction

Pinhole Control Protocol (PCP)[PCP] has proposed amechanism to control how incoming packets are forwarded by upstream NAT devices. Therein, PCP server discovery is trageted to be resolved. It recommended two approaches. One is DHCP based solution. Another is to assign a fixed IPv4 and a fixed IPv6 to locate PCP server. To meet demands, PCP DHCP Options is trying to extend the DHCP as one of the mechanisms to discover a PCP Server. For flexibility reasons, this document defines both IP address and FQDN Options. Service provider could chooce either of them for their convenience. This memo takes Link-Local Multicast Name Resolution (LLMNR) into account as condidate solution to discovery PCP server. Alternatively, it could be complemented DHCP FQDN option to achieve the goal of discovering PCP server.

In the draft-wing-softwire-port-control-protocol-02, several mechanisms for discovering the PCP Server were analyzed. NAPTR is as candidate to be noted as where there would be a hurdle for updating the zone configuration. Especially, any change of the engineering policies occurred, like introducing new CGN device, load-based dimensioning. Such flexibility issues make adherence to DNS is not encouraged. And, more flexibility means are expected to be promoted. LLMNR[RFC4795] could be taken as an advantageous part to improve such flexibility.

The document is structured as follows. Section 2 analyzes the applicability of LLMNR in PCP circumstance. Section 3 elaborates the overall LLMNR deployment consideration and relevant LLMNR configurations, including LLMNR usage, namespace configuration and conflict resolution. Section 5 is further securities consideration.

2. LLMNR Feasibility Analysis

In order to allow PCP applications to learn their external IP address, PCP server should be able to interact with controlled NAT located on the packet egress path. This element might be the same as the device embedding the controlled NAT. Thereby, the position of PCP server might locate on the local link with PCP client. This deployment features would perfectly match region of LLMNR operation.In addition, LLMNR is desirable to consider expand multi-cast scope beyond link-local depending on the specific deployment experiences and service demands. Such scalabilities could allow LLMNR fit into PCP deployment scope.

Propagation of LLMNR packets is considereds sufficient to enable PCP discovery, where there are the PCP functionalites integrated with

current CP (Customer Premises) router model offered to customers. In that case, more fine-grained changes of the engineering policies could be handled by LLMNR better than DNS, since dynamic updates in DNS aren't supported widely.

In draft-bpw-pcp-dhcp-00, DHCPv4/v6 would consider carrying FQDN option to locate PCP server, there is obviously a need to consider a sort of name resolving service to support. LLMNR could avoid adhenrence to DNS considering several flexibilties issues.

LLMNR could increase the searching effiency of PCP server. The response for PCP discovery request could be transmitted to the originator locally, otherwire it might have to flow through the distance network to reach authoritative DNS server.

3. LLMNR Deployment Consideration

LLMNR could fit into the PCP network with minimum modifications on PCP element. LLMNR would install on the PC, laptop and co-locate with PCP client. Currently, LLMNR has already implemented in Windows operation system. It could accelerate the depolyement process. This section will go into relevant details.

3.1. LLMNR Deployment Framework

LLMNR is recommended as sencondary name resolution mechanism to be utilized in some particular situation. It is not targeting to substitute the existing DNS system. Therefore, both LLMNR and DNS would be placed in PCP network. It is up to Service Providers to determine which way to resolve PCP server naming depending on the specific network situations.

In the case of adapting LLMNR, DNS server do not need to restore PCP server related RR. The PCP server discovery will be performed in local network, where there are PCP client and server located.

3.2. LLMNR Usage Model

LLMNR usage should be customized to make it more suitable for PCP situation. The following statements should be taken into account once LLMNR has been deployed in PCP network.

- LLMNR operation scope. RFC4795 originally recommended to send LLMNR queries to 224.0.0.252 over IPv4 and FF02:0:0:0:0:0:1:3 over IPv6. However, it encouraged to expand operation scope based on accumulated experiences. for the PCP case, it needs to figure out the appropriate multicast scope to satisfy PCP accomodation. for

the scenarios where multicast is not enabled, LLMNR could use unicast queries and responses to perform PCP server discovery. for example, it could use sender queries for a PTR RR of a fully formed IP address within the "in-addr.arpa" or "ip6.arpa" zones.

- o LLMNR enabler. LLMNR is enable on a per-interface basis. It could be configured manually or automatically. Along with PCP DHCP options extention, it might consider adding LLMNR enable option to assist DHCP FQDN manner.

- o LLMNR PCP namespace configuration. LLMNR responders may self-allocate a name within the single-label namespace to represent PCP server name. Since single-label names allow to not be unique, Service Provider could self-define the particular name for indicating their PCP server.

- o Conflict detection. LLMNR defined thorough conflict defense procedures to prevent LLMNR response from confusion. These processes should take place in PCP scenarioes. The further consideration need to be identified for resolving specific PCP problems.

3.3. DHCP Consideration

LLMNR is a peer-to-peer name resolution protocol. There was LLMNR Enable DHCP extention work described in LLMNR Enable [LLMNREnable], can be used to explicitly enable or disable use of LLMNR on an interface. The further consideration need to be identified for PCP specific cases.

3.4. Namespace Configuratin Consideration

LLMNR sender SHOULD send LLMNR queries only for single-label names. A specific PCP namespace configuration need to be further identified.

4. IANA Considerations

This memo includes no request to IANA.

5. Security Considerations

The memo would follow LLMNR security consideration. The further consideration need to be identified for PCP specific cases.

6. Normative References

[LLMNREnable] Guttman, E., "DHCP LLMNR Enable Option", April 2002.

[PCP] Wing, D., "Pinhole Control Protocol (PCP)", draft-ietf-pcp-base-16.txt (work in progress), October 2011.

[RFC4795] Aboba, B., Thaler, D., and L. Esibov, "Link-local Multicast Name Resolution (LLMNR)", RFC 4795, January 2007.

Authors' Addresses

Gang Chen
China Mobile
53A,Xibianmennei Ave.,
Xuanwu District,
Beijing 100053
China

Email: chengang@chinamobile.com

Hui Deng
China Mobile
53A,Xibianmennei Ave.
Beijing 100053
P.R.China

Phone: +86-13910750201
Email: denghui02@gmail.com

Junbin Zhang
China Mobile
Jinyuan Building
Jiangxi
P.R.China

Phone:
Email: zhangjunbin@jx.chinamobile.com

PCP working group
Internet-Draft
Intended status: Standards Track
Expires: April 29, 2012

S. Cheshire
Apple

October 27, 2011

PCP Rapid Recovery
draft-cheshire-pcp-recovery-02

Abstract

Port Control Protocol (PCP) Rapid Recovery 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.

Status of this Memo

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

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

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

This Internet-Draft will expire on April 29, 2012.

Copyright Notice

1. 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].

2. Introduction

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

PCP Rapid Recovery 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.

3. PCP Restart Announcement

When a PCP server device [PCP] that implements PCP Rapid Recovery 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 multicasting the UDP packet shown below on all multicast-capable interfaces on which it accepts PCP requests. A PCP server device which accepts PCP requests over IPv4 sends the Restart Announcement to the IPv4 multicast address 224.0.0.1:5350. A PCP server device which accepts PCP requests over IPv6 sends the Restart Announcement to the IPv6 multicast address [FF02::1]:5350. A PCP server device which accepts PCP requests over both IPv4 and IPv6 sends a pair of Restart Announcements, one to each multicast address. 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.

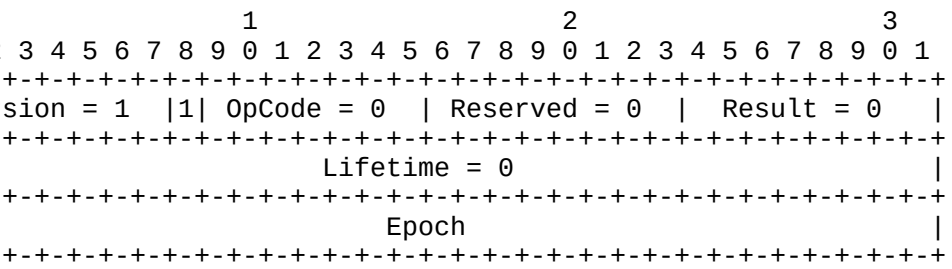


Figure 1: PCP Restart Announcement Packet

A PCP client that implements PCP Rapid Recovery MUST listen to receive these PCP Restart Announcements on all multicast-capable interfaces on which it sends PCP requests. A PCP client device which sends PCP requests using IPv4 MUST listen for packets sent to the IPv4 multicast address 224.0.0.1:5350. A PCP client device which sends PCP requests using IPv6 MUST listen for packets sent to the IPv6 multicast address [FF02::1]:5350. A PCP client device which sends PCP requests using both IPv4 and IPv6 MUST listen 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 PCP Restart Announcement a PCP client MUST (as it does with all received PCP response packets) inspect the Announcement's source IP address, and if the Epoch value is outside the expected range for that server [PCP], then for all PCP mappings it made at that address the client should issue new PCP requests to to recreate any lost mapping state. The use of the Suggested External IP Address and Suggested External Port fields in the client's renewal request 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.

The PCP Rapid Recovery capability enables users to, for example, connect to remote machines using ssh, and then reboot the NAT gateway (or even replace it with completely new hardware) without losing their established ssh connections.

Use of PCP Rapid Recovery is a performance optimization. Without it,

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.

4. PCP Mapping Update

If a PCP server device has not forgotten its mapping state, but for some other reason has determined that some or all of its mappings have become unusable (e.g. when a home gateway is assigned a different external IPv4 address by the upstream DHCP server) then the PCP server device MAY chose to remedy this situation by automatically repairing its mappings and notifying its clients.

For PCP MAP 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 responses to inform the PCP client of the new External IP Address and External Port. Such MAP responses are identical to the MAP responses normally returned in response to client MAP requests, except they may be viewed as a long-delayed response to an earlier MAP request, containing newly updated External IP Address and External Port values.

To accommodate packet loss, the PCP server device MAY transmit such packets up to ten times (with an appropriate Epoch time value in each to reflect the passage of time between transmissions) provided that the interval between the first two notifications is at least 250ms, and the interval between subsequent notification at least doubles.

Upon receipt of such long-delayed MAP responses, a PCP client MUST to use the information in them to update its DNS records, or other address and port information recorded with some kind of application-specific rendezvous server. Existing TCP connections will be lost, but promptly updating the DNS or rendezvous server with the new data will allow new connections to be made.

For PCP PEER mappings there is no general way to recover them (the remote host doesn’t know the new External IP Address and External Port) so existing connections will be lost. Accordingly, a PCP server device is not required to take any specific action for PEER mappings. It MAY delete all PEER mappings immediately (and let application-layer timeouts detect the failure) or it MAY choose to retain them for some time in case another change in the external environment (e.g. a lost DHCP-assigned external address is re-assigned after a few seconds) results in the mappings becoming usable again.

5. Security Considerations

Forged PCP Restart Announcements could be used to cause high load on a PCP server.

Forged MAP responses could be used to mislead a PCP client about what External IP Address and External Port is has been allocated.

6. IANA Considerations

IANA is requested to record that UDP port 5350 is now formally reallocated from "NAT-PMP Restart Announcement" to "PCP Restart Announcement".

7. Normative References

[PCP] Wing, D., "Port Control Protocol (PCP)", draft-ietf-pcp-base-07 (work in progress), March 2011.

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

Author's Address

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

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

PCP Working Group
Internet-Draft
Intended status: Standards Track
Expires: May 2, 2012

F. Dupont
Internet Systems Consortium
T. Tsou
Huawei Technologies
J. Qin
ZTE Corporation
October 30, 2011

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

Abstract

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

Status of this Memo

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

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

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

This Internet-Draft will expire on May 2, 2012.

described in the Simplified BSD License.

1. Introduction

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

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

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

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

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

2. Plain Mode

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

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

In the plain mode the PCP discovery ([I-D.ietf-pcp-base] section 7.1 "General PCP Client: Generating a Request") is changed into:

1. if a PCP server is configured (e.g., in a configuration file or via DHCPv6), that single configuration source is used as the list of PCP Server(s), else;
2. use the IPv6 address of the AFTR.

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

replaced by the AFTR.

3. IANA Considerations

This document makes no request of IANA.

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

4. Security Considerations

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

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

5. Acknowledgments

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

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

6. References

6.1. Normative References

[I-D.ietf-pcp-base] Wing, D., Cheshire, S., Boucadair, M., Penno, R., and P. Selkirk, "Port Control Protocol (PCP)", draft-ietf-pcp-base-16 (work in progress), October 2011.

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

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

6.2. Informative References

[I-D.bpw-pcp-proxy]
Boucadair, M., Penno, R., Wing, D., and F. Dupont, "Port Control Protocol (PCP) Proxy Function", draft-bpw-pcp-proxy-02 (work in progress), September 2011.

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

Appendix A. Encapsulation Mode

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

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

Appendix B. Justification

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

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

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

Authors' Addresses

Francis Dupont
Internet Systems Consortium

Email: fdupont@isc.org

Tina Tsou
Huawei Technologies
2330 Central Expressway
Santa Clara
USA

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

Jacni Qin
ZTE Corporation
Shanghai
P.R.China

Email: jacni@jacni.com

PCP Working Group
Internet-Draft
Intended status: Standards Track
Expires: October 25, 2013

F. Dupont
Internet Systems Consortium
T. Tsou
Huawei Technologies
J. Qin
ZTE Corporation
M. Wasserman
Painless Security
D. Zhang
Huawei

April 23, 2013

The Port Control Protocol in Dual-Stack Lite environments
draft-dupont-pcp-dslite-05

Abstract

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

Status of this Memo

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

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

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

This Internet-Draft will expire on October 25, 2013.

Copyright Notice

1. Introduction

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

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

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

Two different modes of operations have been proposed: the plain and the encapsulation modes. This document recommends use of the plain mode.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

2. Plain Mode

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

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

In the plain mode the PCP discovery ([I-D.ietf-pcp-base] section 7.1 "General PCP Client: Generating a Request") is changed into:

1. if a PCP server is configured (e.g., in a configuration file or via DHCPv6), that single configuration source is used as the list of PCP Server(s), else;
2. use the IPv6 address of the AFTR.

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

3. IANA Considerations

This document makes no request of IANA.

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

4. Security Considerations

A DS-Lite PCP deployment could be secure under the Simple Threat Model described in the Security Considerations of the base PCP specification, even though the B4 device makes PCP mapping requests on behalf of internal clients using the THIRD_PARTY option.

To meet the requirements of the Simple Threat Model the DS-Lite PCP server MUST be configured to only allow the B4 device to make THIRD_PARTY requests, and only on behalf of other Internal Hosts sharing the same DS-Lite IPv6 tunnel. The B4 must ensure that the internal IP address in the THIRD_PARTY option corresponds to the IP source address in the IP Header of the PCP request (or proxied UPnP request) that triggered the THIRD_PARTY request. The B4 device MUST guard against spoofed packets being injected into the IPv6 tunnel using the B4 device's IPv4 source address, so the DS-Lite PCP Server can trust that packets received over the DS-Lite IPv6 tunnel with the B4 device's source IPv4 address do in fact originate from the B4 device. The B4 device is in a position to enforce this requirement, because it is the DS-Lite IPv6 tunnel endpoint.

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

If the conditions described above cannot be ensured, a PCP Authentication mechanism must be implemented to meet the requirements of the Advanced Security Model, as discussed in the PCP specification.

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

On the other hand, the encapsulation mode Appendix A defaults to being fully transparent for the B4: PCP requests are blindly encapsulated as any other IPv4 packets to the Internet. This makes it more difficult to apply policy to PCP requests, and will generally require implementation of a PCP authentication protocol to meet the Security Considerations of the base PCP specification.

5. Acknowledgments

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

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

Sam Hartman for his help with the Security Considerations text.

6. References

6.1. Normative References

[I-D.ietf-pcp-base]
Wing, D., Cheshire, S., Boucadair, M., Penno, R., and P. Selkirk, "Port Control Protocol (PCP)", draft-ietf-pcp-base-29 (work in progress), November 2012.

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

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

6.2. Informative References

[I-D.ietf-pcp-proxy]
Boucadair, M., Penno, R., and D. Wing, "Port Control Protocol (PCP) Proxy Function", draft-ietf-pcp-proxy-02 (work in progress), February 2013.

[I-D.ietf-pcp-upnp-igd-interworking]
Boucadair, M., 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-08 (work in progress), April 2013.

Appendix A. Encapsulation Mode

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

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

Appendix B. Justification

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

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

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

Authors' Addresses

Francis Dupont
Internet Systems Consortium

Email: fdupont@isc.org

Tina Tsou
Huawei Technologies
2330 Central Expressway
Santa Clara
USA

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

Jacni Qin
ZTE Corporation
Shanghai
P.R.China

Email: jacni@jacni.com

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

Dacheng Zhang
Huawei
Beijing,
China

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

PCP working group
Internet-Draft
Intended status: Standards Track
Expires: May 3, 2012

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

October 31, 2011

Port Control Protocol (PCP)
draft-ietf-pcp-base-17

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 3, 2012.

Copyright Notice

Table of Contents

1. Introduction

The Port Control Protocol (PCP) provides a mechanism to control how incoming packets are forwarded by upstream devices such as NAT64, 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 CPE routers, and all of the currently-known transition scenarios towards IPv6-only CPE routers. PCP allows hosts to operate servers for a long time (e.g., a webcam) or a short time (e.g., while playing a game or on a phone call) when behind a NAT device, including when behind a CGN operated by their Internet service provider or an IPv6 firewall integrated in their CPE router.

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

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

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

Many NATs and firewalls include application layer gateways (ALGs) to create mappings for applications that establish additional streams or accept incoming connections. ALGs incorporated into NATs may also modify the application payload. Industry experience has shown that these ALGs are detrimental to protocol evolution. PCP allows an application to create its own mappings in NATs and firewalls,

reducing the incentive to deploy ALGs in NATs and firewalls.

2. Scope

2.1. Deployment Scenarios

PCP can be used in various deployment scenarios, including:

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

2.2. Supported Protocols

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

2.3. Single-homed Customer Premises Network

PCP assumes a single-homed IP address model. That is, for a given IP address of a host, only one default route exists to reach the Internet from that source IP address. This is important because after a PCP mapping is created and an inbound packet (e.g., TCP SYN) arrives at the host, the outbound response (e.g., TCP SYNACK) has to go through the same path so it is seen by the firewall or rewritten by the NAT. This restriction exists because otherwise there would need to be a PCP-enabled NAT for every egress (because the host could not reliably determine which egress path packets would take) and the

client would need to be able to reliably make the same internal/ external mapping in every NAT gateway, which in general is not possible (because the other NATs might have the necessary port mapped to another host).

3. Terminology

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

Internal Host:
: A host served by a NAT gateway, or protected by a firewall. This is the host that receives the incoming traffic resulting from a PCP MAP request, or the host that initiated an implicit dynamic 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.

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

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

Endpoint-Dependent Mapping (EDM): A term applied to NAT operation where an implicit mapping created by outgoing traffic (e.g., TCP SYN) from a single Internal Address and Port to different Remote Peers and Ports may be assigned different External Ports, and a subsequent PCP MAP request for that Internal Address and Port may be assigned yet another different External Port. This term encompasses both Address-Dependent Mapping and Address and Port-Dependent Mapping [RFC4787].

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

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

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

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

Mapping Types:
There are three different ways to create mappings: implicit dynamic mappings, explicit dynamic mappings, and static mappings.

* 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 state, but they can have that effect when they pass through a NAT gateway.

* Explicit dynamic mappings are created as a result of explicit PCP requests. Like a DHCP address lease, explicit dynamic mappings have finite lifetime, and, as with a DHCP address lease, if a client wants a mapping to persist the client must prove that it is still present by periodically renewing the mapping to prevent it from expiring. If a PCP client goes

away, then any mappings it created will be automatically cleaned up when they expire.

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

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

Static mappings and explicit MAP mappings 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).

Static mappings differ from 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.

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 (just as several independent web browsers 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 IGD, [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 implements the server side of the PCP protocol, via which PCP clients request and manage explicit mappings. See also Section 4.

Interworking Function:
A functional element responsible for translating or proxying another protocol to PCP. For example interworking UPnP IGD [IGDv1] with PCP.

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

4. Relationship between PCP Server and its NAT/firewall

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

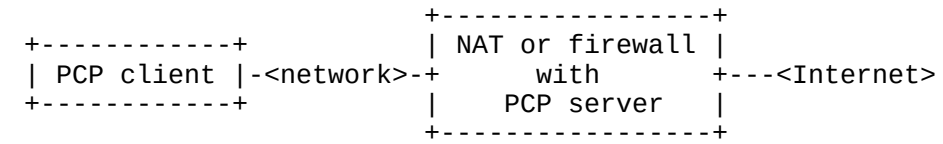


Figure 1: PCP-Enabled NAT or Firewall

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 device supports PCP and wants to receive traffic, and does not possess knowledge of such filtering, it SHOULD use PCP to create the necessary mappings to receive the desired traffic.

5. Note on Fixed-Size Addresses

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

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

When the address field holds an IPv4 address, IPv4-mapped IPv6 addresses [RFC4291] are used (::ffff:0:0/96). This has the first 80 bits set to zero and the next 16 set to one, while its last 32 bits are filled with the IPv4 address. This is unambiguously distinguishable from a native IPv6 address, because IPv4-mapped IPv6 address [RFC4291] are not used as either the source or destination address of actual IPv6 packets.

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 is expressed by filling the fixed-size 128-bit IP address field with all zeroes (::).

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

6. Common Request and Response Header Format

All PCP messages contain a request (or response) header containing an Opcode, any relevant Opcode-specific information, and zero or more Options. The packet layout for the common header, and operation of the PCP client and PCP server, are described in the following sections. The information in this section applies to all Opcodes. Behavior of the Opcodes defined in this document is described in Section 9 and Section 10.

6.1. Request Header

All requests have the following format:

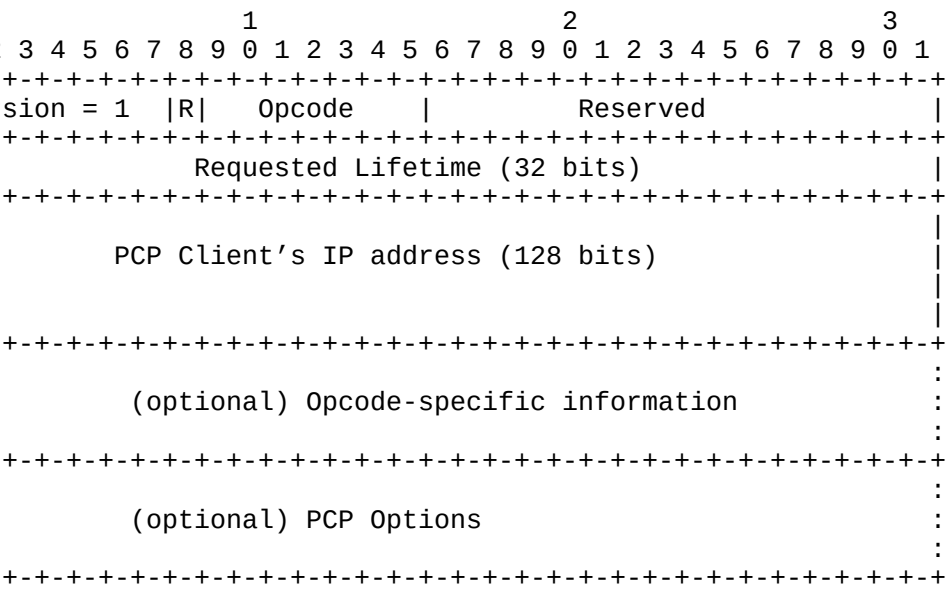


Figure 2: Common Request Packet Format

These fields are described below:

Version: This document specifies protocol version 1. This value MUST be 1 when sending, and MUST be 1 when receiving. This field is used for version negotiation as described in Section 7.7.

R: Indicates Request (0) or Response (1). All Requests MUST use 0.

Opcode: A seven-bit value specifying the operation to be performed. Opcodes are defined in Section 9 and Section 10.

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

Requested Lifetime: An unsigned 32-bit integer, in seconds, ranging from 0 to 4,294,967,295 seconds. This is used by the MAP and PEER Opcodes defined in this document for their requested lifetime. Future Opcodes which don't need this field MUST set the field to zero on transmission and ignore it on reception.

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.

6.2. Response Header

All responses have the following format:

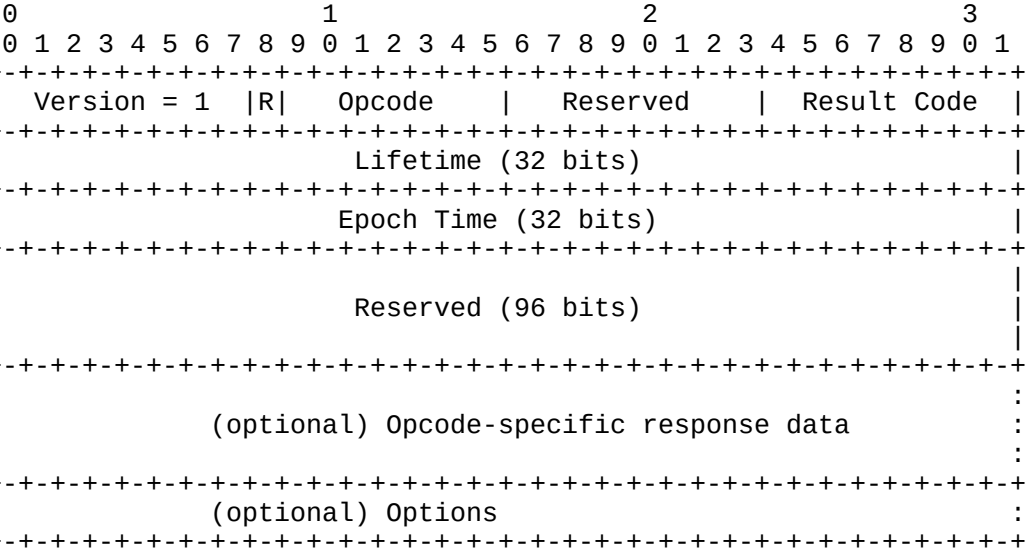


Figure 3: Common Response Packet Format

These fields are described below:

Version: Responses MUST use version 1. This is set by the server.

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

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

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

Result Code: The result code for this response. See Section 6.4 for values. This is set by the server.

Lifetime: An unsigned 32-bit integer, in seconds, ranging from 0 to 4,294,967,295 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 currently-defined PCP Opcodes -- MAP and PEER -- this indicates the lifetime for this mapping. If future Opcodes are defined that do not have a lifetime associated with them, then in success responses for those Opcodes the Lifetime MUST be set to zero on transmission and MUST be ignored on reception. This is set by the server.

Epoch Time: The server's Epoch time value. See Section 7.5 for discussion. This value is set by the server, in both success and error responses.

Reserved: 96 reserved bits, MUST be sent as 0, MUST be ignored when received. This is set by the server. This padding exists so that for unrecognized requests, the server can blindly copy an entire request message into a response message and have that response make some kind of reasonable sense to the recipient.

6.3. 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:

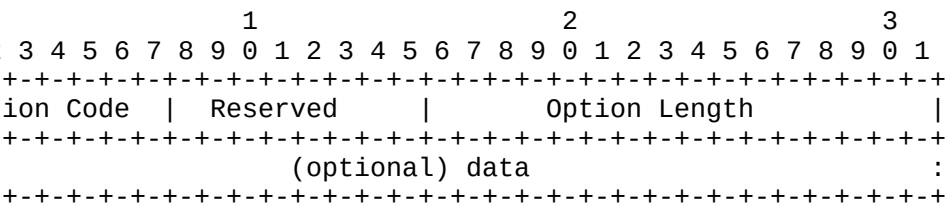


Figure 4: Options Header

The description of the fields is as follows:

Option Code: 8 bits. Its most significant bit and 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.

The handling of an Option by the PCP client and PCP server MUST be specified in an appropriate document, which MUST state whether the PCP Option can appear in a request and/or response, whether it can appear more than once, and indicate what sort of Option data it conveys. 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 with an error including UNPROCESSED Options; this can be achieved by sending messages that don’t exceed a length of 1020-number_of_options, rounded *down* to a multiple of four. (E.g. A PCP request containing one option MUST NOT exceed 1016 octets.)

If, while processing an Option, an error is encountered that causes a PCP error response to be generated, the PCP request MUST cause no state change in the PCP server or the PCP-controlled device (i.e., it rolls back any changes it might have made while processing the request). The response MUST encode the Options in the same order as received in the request. Additional Options included in the response

(if any) MUST be included at the end. 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 this occurs in a response, the PCP client processes the first occurrence and MAY log an error. If the PCP server encounters an invalid option (e.g., option length extends beyond the length of the PCP Opcode itself), the error MALFORMED_OPTION SHOULD be returned (rather than MALFORMED_REQUEST), as that helps the client better understand how the packet was malformed. The UNPROCESSED option MUST NOT appear in a request; if it does, it causes a MALFORMED_REQUEST error. If a PCP response would have exceeded the maximum PCP message size, the PCP server MAY respond with MALFORMED_REQUEST.

The most significant bit in the Option Code indicates if its processing is optional or mandatory. If the most significant bit is set, handling this Option is optional, and a PCP server MAY process or ignore this Option, entirely at its discretion. If the most significant bit is clear, handling this Option is mandatory, and a PCP server MUST process this Option or return an error code if it cannot. If the PCP server does not implement this Option, or cannot perform the function indicated by this Option (e.g., due to a parsing error with the Option), it MUST generate an error response with code UNSUPP_OPTION or MALFORMED_OPTION (as appropriate) and MUST include the UNPROCESSED Option in the response (see Section 7.8.1).

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

Different options are valid for different Opcodes. For example, the UNPROCESSED option is valid for all Opcodes, but only in response messages. The THIRD_PARTY Option is valid for both MAP and PEER Opcodes. The PREFER_FAILURE option is valid only for the MAP Opcode (for the PEER Opcode, its semantics are implied). The FILTER option is valid only for the MAP Opcode (for the PEER Opcode it would have no meaning).

Option definitions MUST include 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>
```

6.4. Result Codes

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

0 SUCCESS: Success.

1 UNSUPP_VERSION: Unsupported protocol version.

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.

4 UNSUPP_OPCODE: Unsupported Opcode.

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

6 MALFORMED_OPTION: Malformed Option (e.g., appears too many times, invalid length).

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

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

9 UNSUPP_PROTOCOL: Unsupported Protocol. This is a long lifetime error.

10 USER_EX_QUOTA: Mapping would exceed user's port quota. This is a short lifetime error.

11 CANNOT_PROVIDE_EXTERNAL: the suggested external port and/or external address cannot be provided. This error MUST only be returned for PEER requests, for MAP requests that included the PREFER_FAILURE Option (because otherwise a new external port could have been assigned), or MAP requests for the SCTP protocol. See Section 11.2 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.

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 11.3 for processing information. This is a long lifetime error.

7. General PCP Operation

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

PCP is idempotent, meaning that if the PCP client sends the same request multiple times (or the PCP client sends the request once and it is duplicated by the network), and the PCP server processes those requests multiple times, the result is the same as if the PCP server had processed only one of those duplicate requests.

7.1. 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 9, and procedures specific to the PEER Opcode are described in Section 10.

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).

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

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

To assist with detecting an on-path NAT, the PCP client MUST include the source IP address of the PCP message in the PCP request. This is typically its own IP address; see Section 12.4 for how this can be coded.

When attempting to contact a PCP server, the PCP client initializes a timer to 2 seconds. The PCP client sends a PCP message to the first server in its list of PCP servers. If no response is received before the timer expires, the timer is doubled (to 4 seconds) and the request is re-transmitted. If no response is received before the timer expires, the timer is doubled again (to 8 seconds) and the request is re-transmitted again, and so on, up to a maximum retransmission interval of fifteen minutes, at which point the PCP request is re-transmitted once every fifteen minutes until it is sucessfully answered.

Once a PCP client has successfully received a response from a PCP server on that interface, it sends subsequent PCP requests to that same server, with a retransmission timer of 2 seconds. If, after 2 seconds, a response is not received from that PCP server, the same

back-off algorithm described above is performed.

7.2. General PCP Server: Processing a Request

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

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

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

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

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

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

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

- Copying the entire request packet, or 1024 octets, whichever is less, and zero-padding the response to a multiple of 4 octets if necessary

- Setting the R bit
- Setting the Result Code
- Setting the Lifetime, Epoch Time and Reserved fields
- Possibly updating other fields in the response if appropriate
- Possibly adding an UNPROCESSED option at the end of the response

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

- Building a response packet, with the R bit set and Result Code zero
- Setting the Lifetime, Epoch Time and Reserved fields
- Possibly setting opcode-specific response data if appropriate
- Adding any processed options to the response message
- Possibly adding an UNPROCESSED option at the end of the response if there were any (non-mandatory) options that were not understood or otherwise not handled for any other reason

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 7.7 are followed.

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

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

If the length of the message exceeds 1024 octets, is not a multiple of 4 octets, or is too short for the opcode in question, it is invalid and a MALFORMED_REQUEST response is generated.

The PCP server compares the source IP address (from the received IP header) with the field PCP Client IP Adddress. 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.

7.3. 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 an outstanding PCP request. It validates that the Opcode matches an outstanding PCP request. Responses shorter than 24 octets, longer than 1024 octets, or not a multiple of 4 octets are invalid and ignored, likely causing the request to be re-transmitted. The response is further matched by comparing fields in the response Opcode-specific data to fields in the request Opcode-specific data, as described by the processing for that Opcode. After these matches are successful, the PCP client checks the Epoch Time field to determine if it needs to restore its state to the PCP server (see Section 7.5).

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.

If the result code is 0 (SUCCESS), the PCP client knows the request was successful.

If the result code is not 0, the request failed. If the result code is UNSUPP_VERSION, processing continues as described in Section 7.7. If the result code is NO_RESOURCES, PCP client SHOULD NOT send *any* further requests to that PCP server for the indicated error lifetime. For other error result codes, the PCP client SHOULD NOT resend the same request for the indicated error lifetime. If the PCP server indicates an error lifetime in excess of 30 minutes, the PCP client MAY choose to set its retry timer to 30 minutes.

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.

7.4. Multi-Interface Issues

Hosts which desire a PCP mapping might be multi-interfaced (i.e., own several logical/physical interfaces). Indeed, a host can be configured with several IPv4 addresses (e.g., WiFi and Ethernet) or dual-stacked. These IP addresses may have distinct reachability scopes (e.g., if IPv6 they might have global reachability scope as

for Global Unicast Address (GUA, [RFC3587]) or limited scope as for Unique Local Address (ULA) [RFC4193]).

IPv6 addresses with global reachability (e.g., GUA) SHOULD be used as the source address when generating a PCP request. IPv6 addresses without global reachability (e.g., ULA [RFC4193]), SHOULD NOT be used as the source interface when generating a PCP request. If IPv6 privacy addresses [RFC4941] are used for PCP mappings, a new PCP request will need to be issued whenever the IPv6 privacy address is changed. This PCP request SHOULD be sent from the IPv6 privacy address itself. It is RECOMMENDED that mappings to the previous privacy address be deleted.

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.

As mentioned in Section 2.3, only single-homed CP routers are in scope. Therefore, there is no viable scenario where a host located behind a CP router is assigned two Global Unicast Addresses belonging to different global IPv6 prefixes.

7.5. Epoch

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

If the PCP server resets or loses the state of its explicit dynamic Mappings (that is, those mappings created by PCP requests), due to reboot, power failure, or any other reason, it MUST reset its Epoch time to its initial starting value (usually zero) to provide this hint to PCP clients. After resetting its Epoch time, the PCP server resumes incrementing the Epoch time value by one every second. Similarly, if the public IP address(es) of the NAT (controlled by the PCP server) changes, the Epoch time MUST be reset. A PCP server MAY maintain one Epoch time value for all PCP clients, or MAY maintain distinct Epoch time values (per PCP client, per interface, or based on other criteria); this choice is implementation-dependent.

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

- If this is the first PCP response the client has received from this PCP server, the Epoch time value is treated as necessarily valid, otherwise
  - If the current PCP server Epoch time value (current_server_time) is less than the previously received PCP server Epoch time value (previous_server_time) then the client treats the Epoch time value as obviously invalid (time should not go backwards), else
    - The client computes the difference between its current local time value (current_client_time) and the time the previous PCP response was received from this PCP server (previous_client_time):
      client_delta = current_client_time - previous_client_time;
    - The client computes the difference between the current PCP server Epoch time value (current_server_time) and the
      previously received Epoch time value (previous_server_time):
      server_delta = current_server_time - previous_server_time;
    - If client_delta+2 < server_delta - server_delta/256 or server_delta+2 < client_delta - client_delta/256 then the client treats the Epoch time value as invalid, else the client treats the Epoch time value as valid
- The client records the current time values for use in its next comparison:
  previous_client_time = current_client_time
  previous_server_time = current_server_time

If the PCP client determined that the Epoch time value it received was invalid then it concludes that the PCP server may have lost state, and promptly renews all its active port mapping leases as described in Section 12.3.1.

Note: The "+2" in the calculations above is to accomodate quantization errors in client and server clocks (up to one second quantization error each in server and client time intervals).

Note: The "/256" in the calculations above is to accomodate inaccurate clocks in low-cost devices. This value allows for a difference of up to 0.4% in clock rate between PCP client and server to be treated as benign by the client.

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

7.6. Rapid Recovery

PCP Rapid Recovery 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.

7.6.1. PCP Restart Announcement

When a PCP server device that implements PCP Rapid Recovery 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 multicasting the UDP packet shown below on all multicast-capable interfaces on which it accepts PCP requests. A PCP server device which accepts PCP requests over IPv4 sends the Restart Announcement to the IPv4 multicast address 224.0.0.1:5350. A PCP server device which accepts PCP requests over IPv6 sends the Restart Announcement to the IPv6 multicast address [FF02::1]:5350. A PCP server device which accepts PCP requests over both IPv4 and IPv6 sends a pair of Restart Announcements, one to each multicast address. 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.

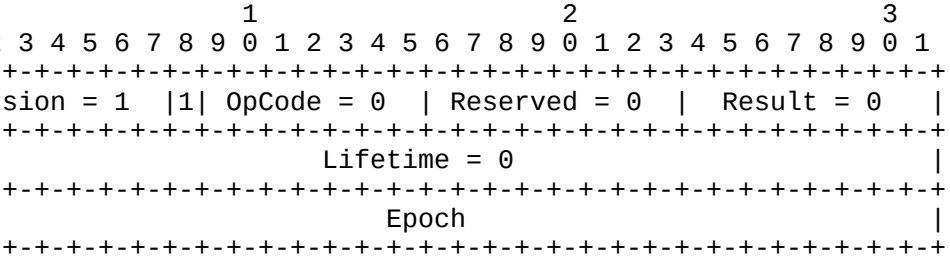


Figure 5: PCP Restart Announcement Packet

A PCP client that implements PCP Rapid Recovery MUST listen to receive these PCP Restart Announcements on all multicast-capable interfaces on which it sends PCP requests. A PCP client device which sends PCP requests using IPv4 MUST listen for packets sent to the IPv4 multicast address 224.0.0.1:5350. A PCP client device which sends PCP requests using IPv6 MUST listen for packets sent to the

IPv6 multicast address [FF02::1]:5350. A PCP client device which sends PCP requests using both IPv4 and IPv6 MUST listen 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 PCP Restart Announcement a PCP client MUST (as it does with all received PCP response packets) inspect the Announcement's source IP address, and if the Epoch value is outside the expected range for that server, then for all PCP mappings it made at that address the client should issue new PCP requests to to recreate any lost mapping state. The use of the Suggested External IP Address and Suggested External Port fields in the client's renewal request 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.

The PCP Rapid Recovery capability enables users to, for example, connect to remote machines using ssh, and then reboot the NAT gateway (or even replace it with completely new hardware) without losing their established ssh connections.

Use of PCP Rapid Recovery is a performance optimization. Without it, 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.

7.6.2. PCP Mapping Update

If a PCP server device has not forgotten its mapping state, but for some other reason has determined that some or all of its mappings have become unusable (e.g. when a home gateway is assigned a different external IPv4 address by the upstream DHCP server) then the PCP server device MAY chose to remedy this situation by automatically repairing its mappings and notifying its clients.

For PCP MAP 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 responses to inform the PCP client of the new External IP Address and External Port. Such MAP responses are identical to the MAP responses normally returned in response to client MAP requests, except they may be viewed as a long-delayed response to an earlier MAP request,

containing newly updated External IP Address and External Port values.

To accommodate packet loss, the PCP server device MAY transmit such packets up to ten times (with an appropriate Epoch time value in each to reflect the passage of time between transmissions) provided that the interval between the first two notifications is at least 250ms, and the interval between subsequent notification at least doubles.

Upon receipt of such long-delayed MAP responses, a PCP client MUST to use the information in them to update its DNS records, or other address and port information recorded with some kind of application-specific rendezvous server. Existing TCP connections will be lost, but promptly updating the DNS or rendezvous server with the new data will allow new connections to be made.

For PCP PEER mappings there is no general way to recover them (the remote host doesn't know the new External IP Address and External Port) so existing connections will be lost. Accordingly, a PCP server device is not required to take any specific action for PEER mappings. It MAY delete all PEER mappings immediately (and let application-layer timeouts detect the failure) or it MAY choose to retain them for some time in case another change in the external environment (e.g. a lost DHCP-assigned external address is re-assigned after a few seconds) results in the mappings becoming usable again.

7.7. Version Negotiation

A PCP client sends its requests using PCP version number 1. Should later updates to this document specify different message formats with a version number greater than 1 it is expected that PCP servers will still support version 1 in addition to the newer version(s). However, in the event that a server returns a response with result code UNSUPP_VERSION, the client MAY log an error message to inform the user that it is too old to work with this server.

Should later updates to this document specify different message formats with a version number greater than 1, and backwards compatibility is desired, this first octet can be used for forward and backward compatibility.

If future PCP versions greater than 1 are specified, version negotiation proceeds as follows:

1. If a client or server supports more than one version it SHOULD support a contiguous range of versions -- i.e., a lowest version and a highest version and all versions in between.

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

3. If the server supports that version it responds normally.

4. 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).

5. 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).

6. If the client receives an UNSUPP_VERSION result containing a version it does not support then 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.

7.8. General PCP Option

The following Option can appear in certain PCP responses, without regard to the Opcode.

7.8.1. UNPROCESSED Option

If the PCP server cannot process any Option, whether mandatory or optional, for whatever reason, it includes the UNPROCESSED Option in the response, shown in Figure 6. This helps with debugging interactions between the PCP client and PCP server. This Option MUST NOT appear more than once in a PCP response. The unprocessed Options are each listed at most once. If a certain Option appeared more than once in the PCP request, that Option value MAY appear once or as many times as it occurred in the request. The order of the Options in the PCP request has no relationship with the order of the Option values in this UNPROCESSED Option. This Option MUST NOT appear in a response unless the associated request contained at least one Option which the server was unable to process.

The UNPROCESSED Option is formatted as follows:

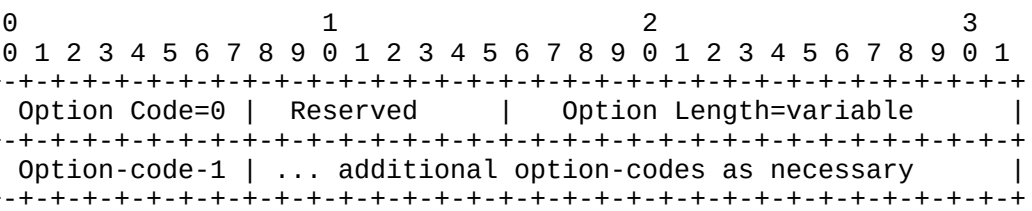


Figure 6: UNPROCESSED option

Option Name: UNPROCESSED
Number: 0
Purpose: indicates which PCP Options in the request were not processed by the PCP server
Valid for Opcodes: all
Length: 1 octet or more
May appear in: responses, and only if the result code is non-zero.
Maximum occurrences: 1

8. Introduction to MAP and PEER Opcodes

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

- a host operating a server and wanting an incoming connection (Section 8.1);
- a host operating a client and server on the same port (Section 8.2);
- a host operating a client and wanting to optimize the application keepalive traffic (Section 8.3);
- and a host operating a client and wanting to restore lost state in its NAT (Section 8.4).

These are discussed in the following sections.

When operating a server (Section 8.1 and Section 8.2) 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., a PCP server on its IPv4 interface or its IPv6 interface), 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 interface (if the PCP server supports NAT44 or IPv4 firewall) or one MAP request from its IPv6 interface (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.

It is REQUIRED that the PCP-controlled device assign the same external IP address to PCP-created explicit dynamic mappings and to implicit dynamic mappings for a given Internal Address. In the absence of a PCP option indicating otherwise, it is REQUIRED that all PCP-created explicit dynamic mappings be assigned the same external IP address. It is RECOMMENDED that static mappings for that Internal Address (e.g., those created by a command-line interface on the PCP server or PCP-controlled device) also be assigned to the same IP address. Once an Internal Address has no implicit dynamic mappings and no explicit dynamic mappings in the PCP-controlled device, a subsequent PCP request 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 hosts to its public IPv4 address pool.

The following table summarizes how various common PCP deployments use IPv6 and IPv4 addresses. The 'internal' address is implicitly the same as the source IP address of the PCP request, except when the THIRD_PARTY option is used. The 'external' address is the Suggested External Address field of the MAP or PEER request, and is address family is usually the same as the 'internal' address family, except when technologies like NAT64 are used. The 'remote peer' address is the Remote Peer IP Address of the PEER request or the FILTER option of the MAP request, and is always the same address family as the 'internal' address, even when NAT64 is used. In NAT64, the IPv6 PCP client is not necessarily aware of the NAT64 or aware of the actual IPv4 address of the remote peer, so it expresses the IPv6 address from its perspective as shown in the table.

	internal	external	remote peer
IPv4 firewall	IPv4	IPv4	IPv4
IPv6 firewall	IPv6	IPv6	IPv6
NAT44	IPv4	IPv4	IPv4
NAT64	IPv6	IPv4	IPv6
NPTv6	IPv6	IPv6	IPv6

Figure 7: Address Families with MAP and PEER

Note that PCP requests containing the MAP or PEER Opcodes cannot delete or shorten the lifetime of an existing implicit mapping for the indicated internal address and port. Conceptually implicit and explicit mappings are different "layers" in the NAT forwarding state database.

8.1. For Operating a Server

A host operating a server (e.g., a web server) listens for traffic on a port, but the server never initiates traffic from that port. For this to work across a NAT or a firewall, the host needs to (a) create a mapping from a public IP address and port to itself as described in Section 9 and (b) publish that public IP address and port via some sort of rendezvous server (e.g., DNS, a SIP message, a proprietary protocol). Publishing the public IP address and port is out of scope of this specification. To accomplish (a), the host follows the procedures described in this section.

As normal, the application needs to begin listening on a port. Then, the application constructs a PCP message with the MAP Opcode, with the external address set to the appropriate all-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 cases, apart from the
      * Suggested External Address and Port which may differ between
      * (1), (2), and (3).
      */
     if (time_to_send_pcp_request())
         pcp_send_map_request(internal_sockaddr.sin_port,
             internal_sockaddr.sin_addr,
             &external_sockaddr, /* will be zero the first time */
             requested_lifetime, &assigned_lifetime);

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

     if (received_incoming_connection_or_packet())
         process_it(s);

     if (other_work_to_do())
         do_it();

     /* ... */

     block_until_we_need_to_do_something_else();
     }
```

Figure 8: Pseudo-code for using PCP to operate a server

8.2. For Operating a Symmetric Client/Server

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

An application that is using the same port for outgoing connections as well as incoming connections MUST first signal its operation of a server using the PCP MAP Opcode, as described in Section 9, 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 TCP SYNs sent from the specified Internal Address 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 12.1.

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

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

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

 while (1)
     {
     /* Note: the "time_to_send_pcp_request()" check below includes:
      * 1. Sending the first request
      * 2. Retransmitting requests due to packet loss
      * 3. Resending a request due to impending lease expiration
      * 4. Resending a request due to server state loss
      * The PCP packet sent is identical in all cases, apart from the
      * Suggested External Address and Port which may differ between
      * (1), (2), and (3).
      */
     if (time_to_send_pcp_request())
         pcp_send_map_request(internal_sockaddr.sin_port,
             internal_sockaddr.sin_addr,
             &external_sockaddr, /* will be zero the first time */
             requested_lifetime, &assigned_lifetime);

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

     if (received_incoming_connection_or_packet())
         process_it(s);

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

     if (data_to_send())
         send_it(s);

     if (other_work_to_do())
         do_it();

     /* ... */

     block_until_we_need_to_do_something_else();
     }
```

Figure 9: Pseudo-code for using PCP to operate a symmetric client/ server

8.3. For Reducing NAT 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 10.

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
      * The PCP packet sent is identical in all cases, apart from the
      * Suggested External Address and Port which may differ between
      * (1), (2), and (3).
      */
     if (time_to_send_pcp_request())
         pcp_send_peer_request(internal_sockaddr.sin_port,
             internal_sockaddr.sin_addr,
             &external_sockaddr, /* will be zero the first time */
             remote_peer, requested_lifetime, &assigned_lifetime);

     if (data_to_send())
         send_it(s);

     if (other_work_to_do())
         do_it();

     /* ... */

     block_until_we_need_to_do_something_else();
     }
```

Figure 10: Pseudo-code using PCP with a dynamic socket

8.4. 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 only creates a new implicit dynamic mapping 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.

9. MAP Opcode

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

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

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

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

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

mapping's Internal Address and Port.

On Operating Systems that allow multiple listening clients to bind to the same Internal Port, clients MUST ensure that they have exclusive use of that Internal Port (e.g., by binding the port using INADDR_ANY, or using SO_EXCLUSIVEADDRUSE or similar) before sending their MAP request, to ensure that no other clients on the same machine are also listening on the same Internal Port.

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

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

9.1. MAP Operation Packet Formats

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

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

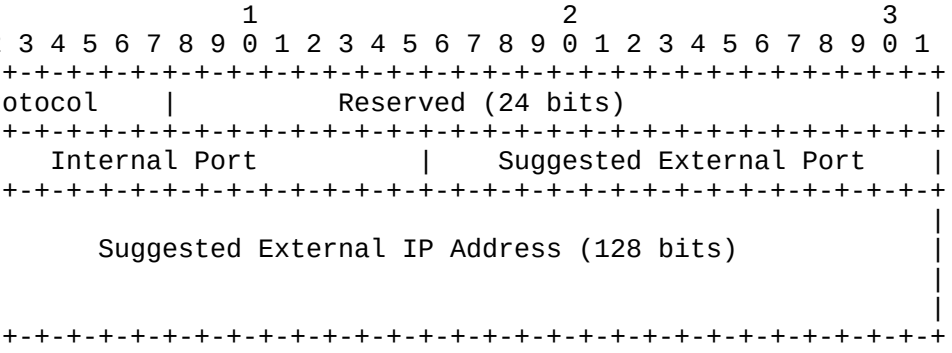


Figure 11: MAP Opcode Request Packet Format

These fields are described below:

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

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

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

Internal Port: Internal port for the mapping. The value 0 indicates "all ports", and is legal when the lifetime is zero (a delete request), if the Protocol does not use 16-bit port numbers, or the Protocol is 0 (meaning 'all protocols')

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

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

The 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:

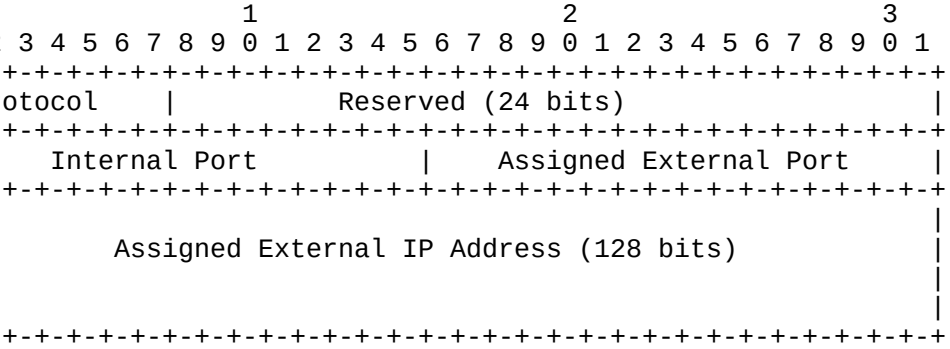


Figure 12: MAP Opcode Response Packet Format

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. The PCP client SHOULD impose an upper limit on this returned Assigned Lifetime value, and 24 hours is RECOMMENDED. This means if the PCP server returns an absurdly long Assigned Lifetime (e.g., 5 years), the PCP client will behave as if it received a more sane value (e.g., 24 hours).

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.

9.2. Generating a MAP Request

This section and Section 9.5 describe the operation of a PCP client when sending requests with the MAP Opcode.

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

If the Protocol does not use 16-bit port numbers (e.g., RSVP), the port number MUST be 0. This will cause all traffic matching that protocol to be mapped.

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

9.2.1. 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 as the suggested external IP address and port, so that if the NAT gateway 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 9.5). To reduce the risk of inadvertent synchronization of renewal requests, a random jitter component should be included. It is RECOMMENDED that PCP clients send a single renewal request packet at a time chosen with uniform random distribution in the range 1/2 to 5/8 of expiration time. If no SUCCESS response is received, then the next renewal request should be sent 3/4 to 3/4 + 1/16 to expiration, and then another 7/8 to 7/8 + 1/32 to expiration, and so on, subject to the constraint that renewal requests MUST NOT be sent less than four seconds apart (a PCP client MUST NOT send a flood of ever-closer-together requests in the last few seconds before a mapping expires).

9.3. Processing a MAP Request

This section and Section 9.5 describe the operation of a PCP server when processing a request with the MAP Opcode. Processing SHOULD be performed in the order of the following paragraphs.

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

If the Requested Lifetime is non-zero, it indicates a request to create a mapping or extend the lifetime of an existing mapping. If the PCP server or PCP-controlled device does not support the Protocol, it MUST generate an UNSUPP_PROTOCOL error. If the requested Lifetime is non-zero, the Internal Port is zero, and the Protocol is non-zero, it indicates a request to map all incoming traffic for that entire Protocol. If this request cannot be fulfilled in its entirety, the error NO_RESOURCES MUST be returned. If the requested Lifetime is non-zero, the Internal Port is zero, and the Protocol is zero, it indicates a request to map all incoming traffic for all protocols. If this request cannot be fulfilled in its entirety, the error NO_RESOURCES MUST be returned. If the Protocol is 0 but the Internal Port is non-zero, the error

MALFORMED_REQUEST MUST be returned.

If the requested lifetime is zero, it indicates a request to delete an existing mapping or set of mappings. Processing of the lifetime is described in Section 9.5.

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

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

If a mapping already exists for the requested Internal Address and Port and the PREFER_FAILURE Option is not present, the PCP server MUST refresh the lifetime of that already-existing mapping, and return the already-existing External Address and Port in its response, regardless of the Suggested External Address and Port in the request. If a mapping already exists for the requested Internal Address and Port and the request contains the PREFER_FAILURE Option, but the Suggested External Address and Port do not match the actual External Address and Port of the already existing mapping, the error CANNOT_PROVIDE_EXTERNAL is returned. If an implicit mapping already exists for the requested Internal Address and Port, a new explicit mapping should be made replicating the ports and addresses from the implicit mapping (but the implicit mapping continues to exist, and remains in effect if the explicit mapping is later deleted).

If no mapping exists for the Internal Address and Port, and the PCP server is able to create a mapping using the Suggested External Address and Port, it SHOULD do so. This is beneficial for re-establishing state lost in the PCP server (e.g., due to a reboot). If the PCP server cannot assign the Suggested External Address and Port but can assign some other External Address and Port (and the request did not contain the PREFER_FAILURE Option) the PCP server MUST do so and return the newly assigned External Address and Port in the response. Cases where a NAT gateway cannot assign the Suggested External Address and Port include:

- The Suggested External Address and Port is already assigned to another existing explicit, implicit, or static mapping (i.e., is already forwarding traffic to some other internal address and port).
- The Suggested External Address and Port is already used by the NAT gateway for one of its own services (e.g., port 80 for the NAT gateway's own configuration pages).
- The Suggested External Address and Port is otherwise prohibited by the PCP server's policy.
- The Suggested External Address or port is invalid (e.g., 127.0.0.1, ::1, multicast address, or the port 0 is not valid for the indicated protocol).
- The Suggested External Address does not belong to the NAT gateway.
- The Suggested External Address is not configured to be used as an external address of the firewall or NAT gateway.
- The PREFER_FAILURE option is included in the request and the Suggested External Address and Port are not assignable to the PCP client, which returns the CANNOT_PROVIDE_EXTERNAL error.

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

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

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

9.4. 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 an outstanding request by comparing the Protocol and Internal Port (and, if THIRD_PARTY, Internal IP Address). Other fields are not compared, because the PCP server sets those fields.

On a success response, the PCP client can use the External IP Address and Port as desired. 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 9.2, 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 12.3.1.

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

9.5. Mapping Lifetime and Deletion

The PCP client requests a certain lifetime, and the PCP server responds with the assigned lifetime. The PCP server MAY grant a lifetime smaller or larger than the requested lifetime. The PCP server SHOULD be configurable for permitted minimum and maximum lifetime, and the RECOMMENDED values are 120 seconds for the minimum value and 24 hours for the maximum. It is RECOMMENDED that the server be configurable to restrict lifetimes to less than 24 hours, because mappings will consume ports even if the Internal Host is no longer interested in receiving the traffic or is no longer connected to the network. These recommendations are not strict, and deployments should evaluate the trade offs to determine their own minimum and maximum lifetime values.

Once a PCP server has responded positively to a mapping 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.

If a PCP client sends a PCP MAP request to create a mapping that already exists as a static mapping, the PCP server will return a successful result, confirming that the requested mapping exists. The lifetime the PCP server returns for such a static mapping SHOULD be 4294967295 (0xFFFFFFFF). Upon receipt of such a MAP response with an absurdly long Assigned Lifetime the PCP client SHOULD behave as if it received a more sane value (e.g., 24 hours), and renew the mapping accordingly, to ensure that if the static mapping is removed the client will continue to maintain the mapping it desires.

If the requested lifetime is zero then:

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

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

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

- If the internal port is non-zero and the protocol is zero, then the request is invalid and the PCP Server MUST return a MALFORMED_REQUEST error to the client.

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

If the PCP client attempts to delete a single static mapping (i.e., a mapping created outside of PCP itself), the error NOT_AUTHORIZED is returned. 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 be idempotent). If the PCP MAP request was for port=0 (indicating 'all ports'), the PCP server deletes all of the explicit dynamic mappings it can (but not any implicit or static mappings), and returns a SUCCESS response. If the deletion request was properly formatted and successfully processed, a SUCCESS response is generated with lifetime of 0 and the server copies the protocol and internal port number from the request into the response. An explicit dynamic mapping MUST NOT have its lifetime reduced by transport protocol messages (e.g., TCP RST, TCP FIN).

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 mappings without using PCP, which will also consume port mappings. If there is a port mapping quota for the Internal Host, frequent restarts such as this may exhaust the quota. PCP provides some protections against such port consumption: When a PCP client first acquires a new IP address (e.g., reboots or joins a new network), it SHOULD remove mappings that may already be instantiated for that new Internal Address. To do this, the PCP client sends a MAP request with protocol, internal port, and lifetime set to 0. Some port mapping APIs (e.g., the "DNSServiceNATPortMappingCreate" API provided by Apple's Bonjour on Mac OS X, iOS, Windows, Linux [Bonjour]) automatically monitor for process exit (including application crashes) and automatically send port mapping deletion requests if the process that requested them goes away without explicitly relinquishing them.

To reduce unwanted traffic and data corruption, External UDP and TCP ports SHOULD NOT be re-used for an interval (TIME_WAIT interval [RFC0793]). However, the PCP server SHOULD allow the previous user of an External Port to re-acquire the same port during that interval.

9.6. Address Change Events

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

When an Internal Host changes its IP address (e.g., by having a different address assigned by the DHCP server) the NAT (or firewall) will continue to send traffic to the old IP address. Typically, the Internal Host will no longer receive traffic sent to that old IP address. Assuming the Internal Host wants to continue receiving traffic, it needs to install new mappings for its new IP address. The suggested external port field will not be fulfilled by the PCP server, in all likelihood, because it is still being forwarded to the old IP address. Thus, a mapping is likely to be assigned a new

external port number and/or public IP address. Note that such host renumbering is not expected to happen routinely on a regular basis for most hosts, since most hosts renew their DHCP leases before they expire (or re-request the same address after reboot) and most DHCP servers honor such requests and grant the host the same address it was previously using before the reboot.

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

9.7. Learning the External IP Address Alone

NAT-PMP [I-D.cheshire-nat-pmp] includes a mechanism to allow clients to learn the External IP Address alone, without also requesting a port mapping. In the case of PCP, this operation no longer makes sense. PCP supports Large Scale NATs (CGN) which may have a pool of External IP Addresses, not just one. A client may not be assigned any particular External IP Address from that pool until it has made at least one implicit or explicit port mapping, and even then only for as long as that implicit or explicit port 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.

10. PEER Opcode

This section defines an Opcode for controlling dynamic mappings.

PEER: Create an explicit dynamic mapping (or query an existing implicit dynamic mapping) to a remote peer's IP address and port.

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 mappings it creates automatically as a result of observing outgoing traffic from Internal Hosts.

10.1. PEER Operation Packet Formats

The PEER Opcode allows the PCP client to create an implicit dynamic mapping (which functions similar to the host sending a TCP SYN), and allows the PCP client to manage an implicit dynamic mapping by extending its lifetime.

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

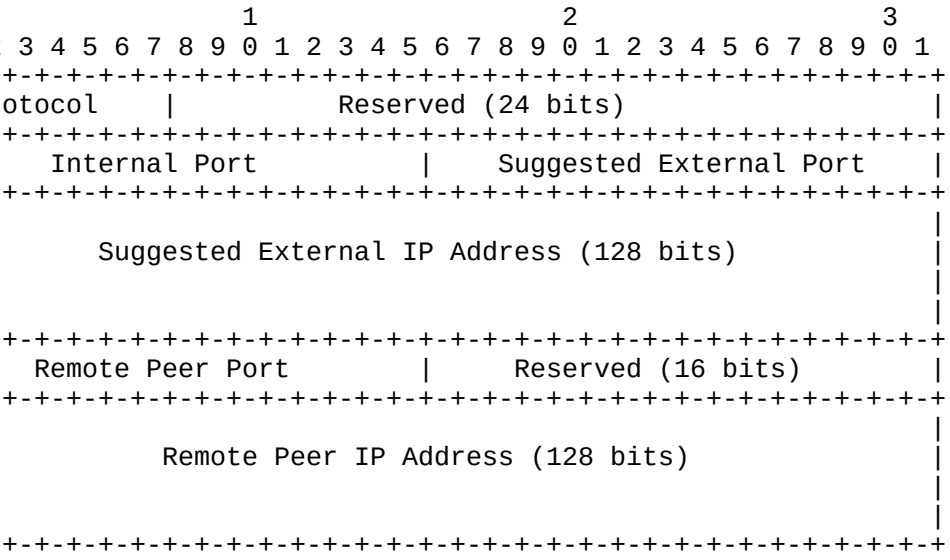


Figure 13: PEER Opcode Request Packet Format

These fields are described below:

Requested Lifetime (in common header): Requested lifetime of this mapping, in seconds. Note that, depending on the implementation of the PCP-controlled device, it may not be possible to reduce the lifetime of a mapping (or delete it, with requested lifetime=0)

using PEER.

Protocol: Upper-layer protocol associated with this Opcode. Values are taken from the IANA protocol registry [proto_numbers]. For example, this field contains 6 (TCP) if the Opcode is describing a TCP mapping.

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

Internal Port: Internal port for the mapping.

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

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

Remote Peer Port: Remote peer's port for the mapping.

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

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

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

Note that the PREFER_FAILURE semantics are automatically implied by PEER requests. If the Suggested External IP Address or Suggested External Port fields are non-zero, and the PCP server is unable to honor the Suggested External IP Address or Port, then the PCP server MUST return a CANNOT_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 response packet format for the PEER Opcode:

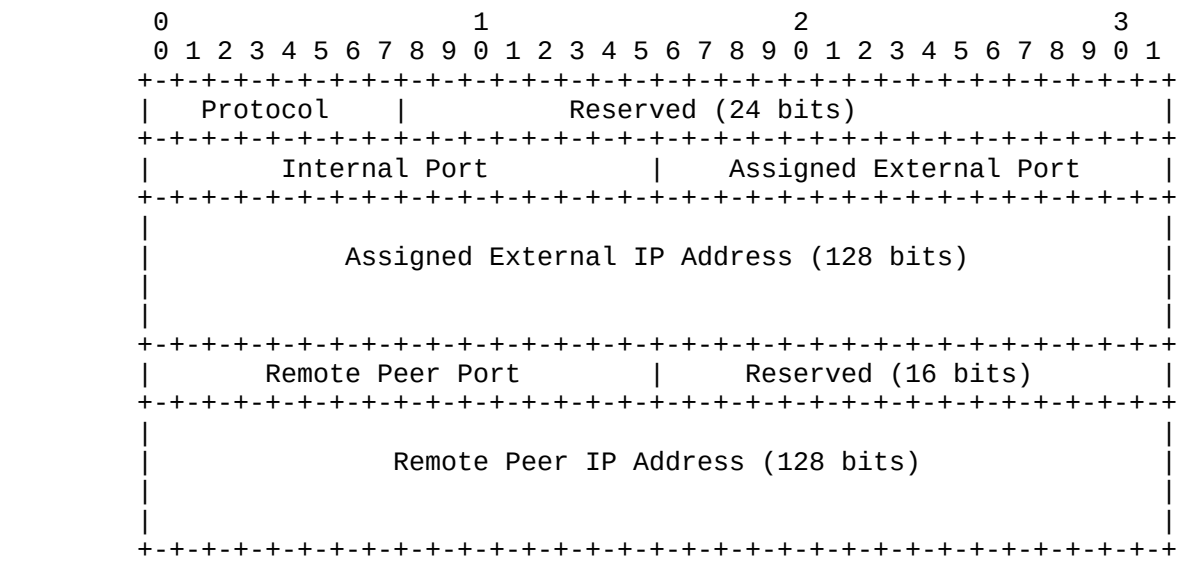


Figure 14: PEER Opcode Response Packet Format

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

Protocol: Copied from the request.

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

Internal Port: Copied from request.

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

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

Remote Peer port: Copied from request.

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

Remote Peer IP Address: Copied from the request.

10.2. 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 mapping in the PCP-controlled device, which will be used for translating traffic to and from the remote peer; this mapping functions the same as if an implicit dynamic mapping were created (e.g., because of a TCP SYN from the client). This is useful for restoring a mapping after a NAT has lost its implicit mapping state (e.g., due to a crash).

If sent after, this is considered an "implicit dynamic mapping". This allows the client to learn the IP address, port, and lifetime of the assigned External Address and Port for the implicit mapping, and to extend this lifetime (for the purpose described in Section 8.3).

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

10.3. Processing a PEER Request

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

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

Peer Port.

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

On receiving the PEER Opcode, the PCP server examines the mapping table. If the requested mapping does not yet exist, and the Suggested External Address and Port can be honored, the mapping is created. By having PEER create such a mapping, we avoid a race condition between the PEER request or the initial outgoing packet arriving at the NAT gateway first, and allow PEER to be used to recreate an implicit dynamic mapping (see last paragraph of Section 12.3.1). If the requested mapping does not yet exist, and Suggested External Address and Port cannot be honored, the error CANNOT_PROVIDE_EXTERNAL is returned. If the requested mapping already exists, it is a request to modify the lifetime of that existing mapping.

The PEER Opcode can extend the lifetime of an existing implicit dynamic mapping. The PCP server may grant the client's requested lifetime, or may grant a value higher or lower, depending on local policy. The PEER Opcode MAY reduce the lifetime of an existing implicit dynamic mapping, but not to less than the lifetime that would result from the gateway seeing outbound traffic using that 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, i.e. continued outbound traffic keeps the mapping alive. 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).

10.4. 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 a request by comparing the protocol, internal IP address (if THIRD_PARTY), internal port, remote peer address and remote peer port. Other fields are not compared, because the PCP server changes those fields to provide information about the mapping created by the Opcode.

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 PCP client wishes to keep this mapping alive beyond the indicated lifetime, it MAY issue a new PCP request prior to the expiration, or it MAY rely on continued inside-to-outside traffic to ensure the mapping will continue to exist. See Section 9.2.1 for recommended renewal timing.

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

11. 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.

11.1. 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.

The THIRD_PARTY Option is formatted as follows:

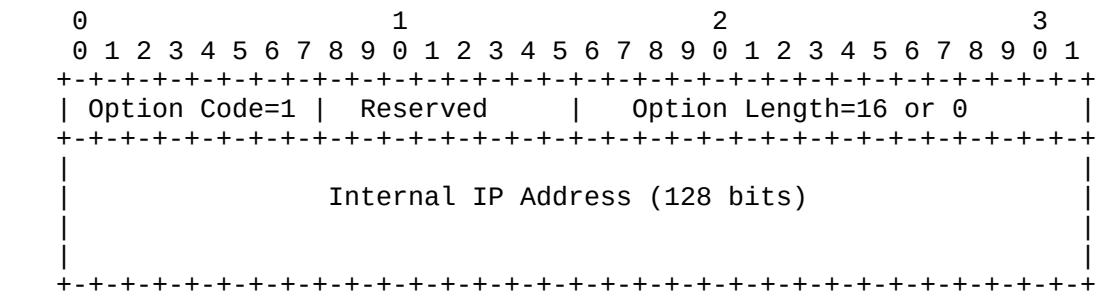


Figure 15: THIRD_PARTY Option packet format

The fields are described below:

Option Length: The only valid option lengths are 0 and 16. The length 0 has special meaning (see 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: 0 or 16 octets
May appear in: request. May appear in response only if it appeared in the associated request.
Maximum occurrences: 1

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

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

It is RECOMMENDED that customer premises equipment implementing a PCP Server be configured to prohibit third party mappings by default. With this default, if a user wants to create a third party mapping, the user needs to interact out-of-band with their customer premises router (e.g., using its HTTP administrative interface).

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

Determining which PCP clients are authorized to use the THIRD_PARTY Option for which other hosts is deployment-dependent. For example, an ISP using Dual-Stack Lite could choose to allow a client connecting over a given IPv6 tunnel to manage mappings for any other host connecting over the same IPv6 tunnel, or the ISP could choose to allow only the DS-Lite B4 element to manage mappings for other hosts connecting over the same IPv6 tunnel. A cryptographic authentication and authorization model is outside the scope of this specification. Note that the THIRD_PARTY Option is not needed for today's common scenario of an ISP offering a single IP address to a customer who is using NAT to share that address locally, since in this scenario all the customer's hosts appear to be a single host from the point of view of the ISP.

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

A PCP client can delete all PCP-created explicit dynamic mappings (i.e., those created by PCP MAP requests) that it is authorized to delete by sending a PCP MAP request including a zero-length THIRD_PARTY Option, zero in the Internal Port field, and zero in the Protocol field.

11.2. 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:

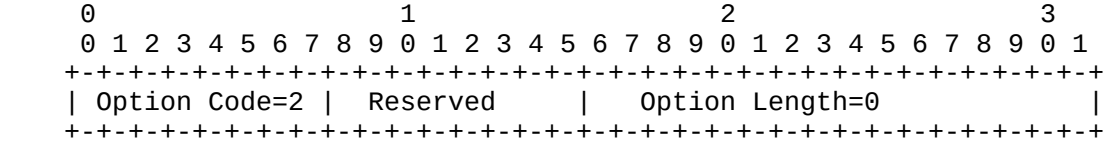


Figure 16: PREFER_FAILURE Option packet format

Option Name: PREFER_FAILURE
Number: 2
Purpose: indicates that the PCP server should not create an alternative mapping if the suggested external port and address cannot be mapped.
Valid for Opcodes: MAP
Length: 0
May appear in: request. May appear in response only if it appeared in the associated request.
Maximum occurrences: 1

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

This Option exists solely for use by UPnP IGD interworking [I-D.bpw-pcp-upnp-igd-interworking], where the semantics of UPnP IGD version 1 only allow the UPnP IGD client to dictate mapping a specific port. A PCP server MAY support this Option, if its designers wish to support downstream devices that perform UPnP IGD interworking. 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. PCP servers that are not intended to support downstream devices that perform UPnP IGD interworking are not required to support this Option. PCP clients other than UPnP IGD interworking clients 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 UPnP IGD interworking declines.

11.3. FILTER Option for MAP Opcode

This Option is only used with the MAP Opcode.

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

The FILTER Option is formatted as follows:

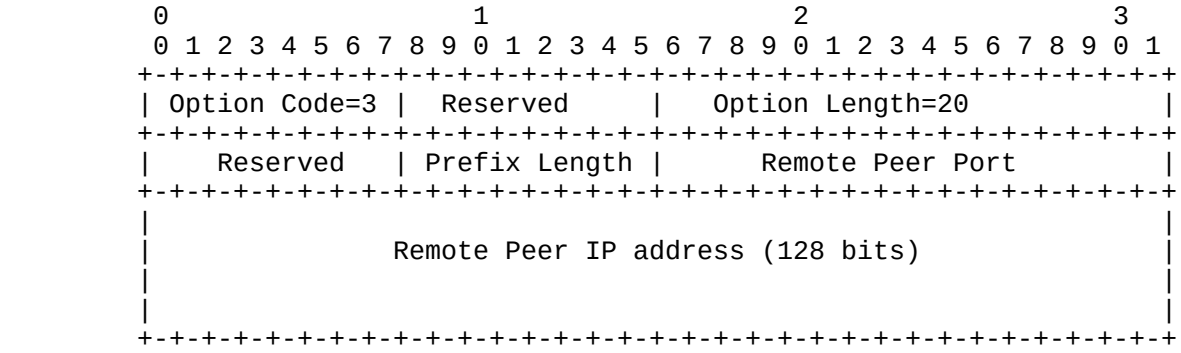


Figure 17: FILTER Option layout

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 IPv6 address or IPv4 address are used for the filter. For IPv4 addresses, which are represented using the IPv4-mapped address format (::FFFF:0:0/96), the value of the Prefix Length pertains only to to the IPv4 portion of the address. Thus, a Prefix Length of 32 with an IPv4-mapped address indicates "only this address". With IPv4-mapped addresses, the minimum Prefix length value is 0 and the maximum is 32; for IPv6 addresses the minimum value is 0 and the maximum is 128. Values outside those range cause the PCP server to return the MALFORMED_OPTION result code.

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

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

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

To change an existing filter, the PCP client sends a MAP request containing two FILTER Options, the first Option containing a Prefix Length of 0 (to delete all existing filters) and the second containing the new remote peer’s IP address and port. Other FILTER

Options in that PCP request, if any, add more allowed Remote Peers.

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

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

The use of the FILTER Option can be seen as a performance optimization. Since all software using PCP to receive incoming connections also has to deal with the case where it may be directly connected to the Internet and receive unrestricted incoming TCP connections and UDP packets, if it wishes to restrict incoming traffic to a specific source address or group of source addresses such software already needs to check the source address of incoming traffic and reject unwanted traffic. However, the FILTER Option is a particularly useful performance optimization for battery powered wireless devices, because it can enable them to conserve battery power by not having to wake up just to reject a unwanted traffic.

12. Implementation Considerations

12.1. Implementing MAP with EDM port-mapping NAT

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

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

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

1. Have implicit dynamic mappings use a different set of public ports than explicit dynamic 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 an implicit dynamic

mapping to process that packet. If none match, then the incoming packet should use the explicit dynamic mapping to process that packet. This effectively 'prioritizes' implicit dynamic mappings above explicit dynamic mappings.

12.2. Lifetime of Explicit and Implicit Dynamic Mappings

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

No matter if a NAT is EIM or EDM, it is possible that one (or more) implicit dynamic 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 implicit dynamic mapping (e.g., via a TCP FIN handshake) does not prematurely destroy the MAP-created mapping. On a NAT that implements endpoint-independent mapping with endpoint-independent filtering, this could be implemented by extending the lifetime of the implicit dynamic mapping to the lifetime of the explicit dynamic mapping.

12.3. PCP Failure Scenarios

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

If an event occurs that causes the PCP server to lose explicit 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].

12.3.1. Recreating Mappings

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

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

When the PCP server loses state and begins processing new PCP messages, its Epoch time is reset and begins counting again. As the result of receiving a packet where the Epoch time field is outside the expected range (Section 7.5), 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.

12.3.2. Maintaining Mappings

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

A PCP client refreshes a mapping by sending a new PCP request containing information from the earlier PCP response. The PCP server will respond indicating the new lifetime. It is possible, due to reconfiguration or failure of the PCP server, that the public IP address and/or public port, or the PCP server itself, has changed (due to a new route to a different PCP server). Such events are not an error. The PCP server will simply return a new External Address and/or External Port to the client, and the client should record this new External Address and Port with its rendezvous service. To detect such events more quickly, the PCP client may find it beneficial to use shorter lifetimes (so that it communicates with the PCP server more often).

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

If a PCP client changes its Internal IP Address (e.g., because the

Internal Host has moved to a new network), and the PCP client wishes to still receive incoming traffic, it needs create new mappings on that new network. New mappings will typically also require an update to the application-specific rendezvous server if the External Address or Port are different to the previous values (see Section 8.1 and Section 9.6).

12.3.3. 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]. Because implicit dynamic SCTP mappings use the verification tag of the association instead of the local and remote peer port numbers, explicit dynamic SCTP mappings need only be established by passive listeners expecting to receive new associations at the external port.

Because an SCTP-aware NAT does not rewrite SCTP port numbers (and firewalls never do), a PCP MAP or PEER request for an SCTP mapping SHOULD provide the same Internal Port and Suggested External Port. If the PCP server supports SCTP, and the suggested external port cannot be provided in an explicit dynamic SCTP mapping, then the error CANNOT_PROVIDE_EXTERNAL is returned.

12.4. Source Address Replicated in PCP Header

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

1. Create a UDP socket.
2. Bind the UDP socket.
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 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 bound UDP socket.

13. Deployment Considerations

13.1. Ingress Filtering

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

13.2. 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.

14. 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.

14.1. Simple Threat Model

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

Defending against attackers who can modify or drop packets between the internal network and the PCP server, or who can inject spoofed packets that appear to come from the internal network is out-of-scope.

A PCP Server is secure under this threat model if the PCP Server is constrained so that it does not configure any explicit mapping that it would not configure implicitly. In most cases, this means that PCP Servers running on NAT boxes or stateful firewalls that support the PEER Opcode can be secure under this threat model if all of their hosts are within a single administrative domain (or if the internal hosts can be securely partitioned into separate administrative domains, as in the DS-Lite B4 case), explicit mappings are created with the same lifetime as implicit mappings, the PCP server does not support deleting or reducing the lifetime of existing mappings, and the PCP server does not support the third party option. PCP Servers can also securely support the MAP Opcode under this threat model if the security policy on the device running the PCP Server would permit endpoint independent filtering of implicit mappings.

PCP Servers that comply with the Simple Threat Model and do not implement a PCP security mechanism described in Section 14.2 MUST enforce the constraints described in the paragraph above.

14.1.1. Attacks Considered

- If you allow multiple administrative domains to send PCP requests to a single PCP server that does not enforce a boundary between the domains, it is possible for a node in one domain to perform a denial of service attack on other domains, or to capture traffic that is intended for a node in another domain.

- If explicit mappings have longer lifetimes than implicit mappings, it makes it easier to perpetrate a denial of service attack than it would be if the PCP Server was not present.

- If the PCP Server supports deleting or reducing the lifetime of existing mappings, this allows an attacking node to steal an existing mapping and receive traffic that was intended for another node.

- If the THIRD_PARTY Option is supported, this also allows an attacker to open a window for an external node to attack an internal node, allows an attacker to steal traffic that was intended for another node, or may facilitate a denial of service attack. One example of how the THIRD_PARTY Option could grant an attacker more capability than a spoofed implicit mapping is that the PCP server (especially if it is running in a service provider's network) may not be aware of internal filtering that would prevent spoofing an equivalent implicit mapping, such as filtering between a guest and corporate network.

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

14.1.2. 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.

14.1.2.1. Residential Gateway Deployment

Parity with many currently-deployed residential gateways can be achieved using a PCP Server that is constrained as described in Section 14.1.1 above.

14.1.2.2. DS-Lite Deployment

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

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

DS-Lite's security policies may also permit use of the MAP Opcode.

14.2. Advanced Threat Model

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

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

- Security infrastructure equipment, such as corporate firewalls, that does not create implicit mappings.
- Equipment (such as CGNs or service provider firewalls) that serve multiple administrative domains and do not have a mechanism to securely partition traffic from those domains.
- Any implementation that wants to be more permissive in authorizing explicit mappings than it is in authorizing implicit mappings.
- Implementations that support the THIRD_PARTY Option (unless they can meet the constraints outlined in Section 14.1.2.2).
- Implementations that wish to support any deployment scenario that does not meet the constraints described in Section 14.1.

To protect against attacks under this threat model, a PCP security mechanism which 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 14.1 above, in their default configuration, when processing unauthenticated requests.

14.3. Residual Threats

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

14.3.1. 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 13.2 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 effective deny service. Through such actions, the PCP

client might not be aware the PCP server might have actually processed the PCP request.

14.3.2. 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.

14.3.3. 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"). A rapid recovery mechanism to immediately inform the PCP client of state loss would reduce this interval, but would not completely eliminate this threat. The PCP client can reduce this interval by using a relatively short lifetime; however, this increases the amount of PCP chatter. This threat is reduced by using persistent storage of explicit dynamic mappings in the PCP server (so it does not lose explicit dynamic mapping state), or by ensuring the previous external IP address and port cannot be used by another host (e.g., by using a different IP address pool).

14.3.4. Attacks Against Server Discovery

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

15. IANA Considerations

IANA is requested to perform the following actions:

15.1. Port Number

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

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

15.2. Opcodes

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

value	Opcode
0	Reserved for "no-op" operation code
1	MAP
2	PEER
3-95	(specification required)
96-126	(private use)
127	Reserved

The values 0 and 127 are Reserved and may be assigned via Standards Action [RFC5226]. The values in the range 3-95 can be assigned via Specification Required [RFC5226], and the range 96-126 is for Private Use [RFC5226].

15.3. Result Codes

IANA shall create a new registry for PCP result codes, numbered 0-255, initially populated with the result codes from Section 6.4. The value 255 is Reserved and may be assigned via Standards Action [RFC5226].

Result Codes in the range 13-191 can be assigned via Specification Required [RFC5226], and the range 192-254 is for Private Use [RFC5226].

15.4. Options

IANA shall create a new registry for PCP Options, numbered 0-255 with an associated mnemonic. The values 0-127 are mandatory-to-process, and 128-255 are optional to process. The initial registry contains the Options described in Section 7.8.1 and Section 11. The Option values 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 Specification Required [RFC5226], and the ranges 64-126 and 192-254 are for Private Use [RFC5226].

16. Acknowledgments

Thanks to Xiaohong Deng, Alain Durand, Christian Jacquenet, Jacni Qin, Simon Perreault, James Yu, Tina TSOU (Ting ZOU), Felipe Miranda

Costa, and James Woodyatt 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 Margaret Wasserman for writing the Security Considerations section.

17. References

17.1. 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.

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

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

[RFC4193] Hinden, R. and B. Haberman, "Unique Local IPv6 Unicast Addresses", RFC 4193, October 2005.

[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>.

17.2. Informative References

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

[I-D.arkko-dual-stack-extra-lite] Arkko, J. and L. Eggert, "Scalable Operation of Address Translators with Per-Interface Bindings", draft-arkko-dual-stack-extra-lite-03 (work in progress), October 2010.

[I-D.boucadair-pcp-failure] Boucadair, M., Dupont, F., and R. Penno, "Port Control Protocol (PCP) Failure Scenarios", draft-boucadair-pcp-failure-00 (work in progress), January 2011.

[I-D.bpw-pcp-upnp-igd-interworking] Boucadair, M., Penno, R., Wing, D., and F. Dupont, "Universal Plug and Play (UPnP) Internet Gateway Device (IGD)-Port Control Protocol (PCP) Interworking Function", draft-bpw-pcp-upnp-igd-interworking-01 (work in progress), December 2010.

[I-D.cheshire-dnsext-dns-sd] Cheshire, S. and M. Krochmal, "DNS-Based Service Discovery", draft-cheshire-dnsext-dns-sd-08 (work in progress), January 2011.

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

[I-D.dupont-pcp-dslite] Dupont, F., Tsou, T., and J. Qin, "The Port Control Protocol in Dual-Stack Lite environments", draft-dupont-pcp-dslite-00 (work in progress), August 2011.

[I-D.ietf-behave-lsn-requirements] Yamagata, I., Miyakawa, S., Nakagawa, A., and H. Ashida, "Common requirements for IP address sharing schemes", draft-ietf-behave-lsn-requirements-00 (work in progress), October 2010.

[I-D.ietf-behave-sctpnat] Stewart, R., Tuexen, M., and I. Ruengeler, "Stream Control Transmission Protocol (SCTP) Network Address Translation",

draft-ietf-behave-sctpnat-04 (work in progress), December 2010.

[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.

[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.

[RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, February 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.

[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.

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-16 to -17

- suggest acquiring a mapping to the Discard port if there is a desire to show the user their external address (Section 9.7).

B.2. Changes from draft-ietf-pcp-base-15 to -16

- fixed mistake in PCP request format (had 32 bits of extraneous fields)
- 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
- corrected text on Client Processing a Response related to receiving ADDRESS_MISMATCH error.
- updated Epoch text.
- Added text that MALFORMED_REQUEST is generated for MAP if Protocol is zero but Internal Port is non-zero.

B.3. Changes from draft-ietf-pcp-base-14 to -15

- Softened and removed text that was normatively explaining how PEER is implemented within a NAT.
- 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.
- combined SERVER_OVERLOADED and NO_RESOURCES into one error code, NO_RESOURCES.
- SCTP mappings have to use same internal and suggested external ports, and have implied PREFER_FAILURE semantics.
- Re-instated ADDRESS_MISMATCH error, which only checks the client address (not its port).

B.4. Changes from draft-ietf-pcp-base-13 to -14

- Moved discussion of socket operations for PCP source address into Implementation Considerations section.
- Integrated numerous WGLC comments.
- NPTv6 in scope.
- Re-written security considerations section. Thanks, Margaret!
- Reduced PEER4 and PEER6 Opcodes to just a single Opcode, PEER.
- Reduced MAP4 and MAP6 Opcodes to just a single Opcode, MAP.
- Rearranged the PEER packet formats to align with MAP.
- 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.
- The THIRD_PARTY Option from an unauthorized host generates UNSUPP_OPTION, so the PCP server doesn't disclose it knows how to process THIRD_PARTY Option.
- Added table to show which fields of MAP or PEER need IPv6/IPv4 addresses for IPv4 firewall, DS-Lite, NAT64, NAT44, etc.
- Accommodate the server's Epoch going up or down, to better detect switching to a different PCP server.
- Removed ADDRESS_MISMATCH; the server always includes its idea of the Client's IP Address and Port, and it's up to the client to detect a mismatch (and rectify it).

B.5. Changes from draft-ietf-pcp-base-12 to -13

- All addresses are 128 bits. IPv4 addresses are represented by IPv4-mapped IPv6 addresses (::FFFF/96)
- PCP request header now includes PCP client's port (in addition to the client's IP address, which was in -12).
- new ADDRESS_MISMATCH error.
- removed PROCESSING_ERROR error, which was too similar to MALFORMED_REQUEST.

- Tweaked text describing how PCP client deals with multiple PCP server addresses (Section 7.1)

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

- Clarified how PCP client chooses MAP4 or MAP6, depending on the presence of its own IPv6 or IPv4 interfaces (Section 8).

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

- clarified that MAP-created mappings have no filtering, and PEER-created mappings have whatever filtering and mapping behavior is normal for that particular NAT / firewall.

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

- allow new Options to be defined with a specification (rather than standards action)

B.6. Changes from draft-ietf-pcp-base-11 to -12

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

B.7. Changes from draft-ietf-pcp-base-10 to -11

- clarified what can cause CANNOT_PROVIDE_EXTERNAL error to be generated.

B.8. Changes from draft-ietf-pcp-base-09 to -10

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

B.9. Changes from draft-ietf-pcp-base-08 to -09

- Clarified in PEER Opcode introduction (Section 10) that they can also create mappings.

- More clearly explained how PEER can re-create an implicit dynamic mapping, for purposes of rebuilding state to maintain an existing session (e.g., long-lived TCP connection to a server).

- Added Suggested External IP Address to the PEER Opcodes, to allow more robust rebuilding of connections. Added related text to the PEER server processing section.

- Removed text encouraging PCP server to statefully remember its mappings from Section 12.3.1, as it didn't belong there. Text in Security Considerations already encourages persistent storage.

- More clearly discussed how PEER is used to re-establish TCP mapping state. Moved it to a new section, as well (it is now Section 8.4).

- MAP errors now copy the Suggested IP Address (and port) fields to Assigned IP Address (and port), to allow PCP client to distinguish among many outstanding requests when using PREFER_FAILURE.

- Mapping theft can also be mitigated by ensuring hosts can't re-use same IP address or port after state loss.

- the UNPROCESSED option is renumbered to 0 (zero), which ensures no other option will be given 0 and be unable to be expressed by the UNPROCESSED option (due to its 0 padding).

- created new Implementation Considerations section (Section 12) which discusses non-normative things that might be useful to implementers. Some new text is in here, and the Failure Scenarios text (Section 12.3) has been moved to here.

- Tweaked wording of EDM NATs in Section 12.1 to clarify the problem occurs both inside->outside and outside->inside.

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

- fixed zero/non-zero text in Section 9.5.

- removed duplicate text saying MAP is allowed to delete an implicit dynamic mapping. It is still allowed to do that, but it didn't need to be said twice in the same paragraph.

- Renamed error from UNAUTH_TARGET_ADDRESS to UNAUTH_THIRD_PARTY_INTERNAL_ADDRESS.

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

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

B.10. Changes from draft-ietf-pcp-base-07 to -08

- moved all MAP4-, MAP6-, and PEER-specific options into a single section.
- discussed NAT port-overloading and its impact on MAP (new section Section 12.1), which allowed removing the IMPLICIT_MAPPING_EXISTS error.
- eliminated NONEXIST_PEER error (which was returned if a PEER request was received without an implicit dynamic mapping already being created), and adjusted PEER so that it creates an implicit dynamic mapping.
- Removed Deployment Scenarios section (which detailed NAT64, NAT44, Dual-Stack Lite, etc.).
- Added Client's IP Address to PCP common header. This allows server to refuse a PCP request if there is a mismatch with the source IP address, such as when a non-PCP-aware NAT was on the path. This should reduce failure situations where PCP is deployed in conjunction with a non-PCP-aware NAT. This addition was consensus at IETF80.
- Changed UNSPECIFIED_ERROR to PROCESSING_ERROR. Clarified that MALFORMED_REQUEST is for malformed requests (and not related to failed attempts to process the request).
- Removed MISORDERED_OPTIONS. Consensus of IETF80.
- SERVER_OVERLOADED is now a common PCP error (instead of specific to MAP).
- Tweaked PCP retransmit/retry algorithm again, to allow more aggressive PCP discovery if an implementation wants to do that.
- Version negotiation text tweaked to soften NAT-PMP reference, and more clearly explain exactly what UNSUPP_VERSION should return.
- PCP now uses NAT-PMP's UDP port, 5351. There are no normative changes to NAT-PMP or PCP to allow them both to use the same port number.
- New Appendix A to discuss NAT-PMP / PCP interworking.
- improved pseudocode to be non-blocking.

- clarified that PCP cannot delete a static mapping (i.e., a mapping created by CLI or other non-PCP means).
- moved theft of mapping discussion from Epoch section to Security Considerations.

B.11. Changes from draft-ietf-pcp-base-06 to -07

- tightened up THIRD_PARTY security discussion. Removed "highest numbered address", and left it as simply "the CPE's IP address".
- removed UNABLE_TO_DELETE_ALL error.
- renumbered Opcodes
- renumbered some error codes
- assigned value to IMPLICIT_MAPPING_EXISTS.
- UNPROCESSED can include arbitrary number of option codes.
- Moved lifetime fields into common request/response headers
- We've noticed we're having to repeatedly explain to people that the "requested port" is merely a hint, and the NAT gateway is free to ignore it. Changed name to "suggested port" to better convey this intention.
- Added NAT-PMP transition section
- Separated Internal Address, External Address, Remote Peer Address definition
- Unified Mapping, Port Mapping, Port Forwarding definition
- adjusted so DHCP configuration is non-normative.
- mentioned PCP refreshes need to be sent over the same interface.
- renamed the REMOTE_PEER_FILTER option to FILTER.
- Clarified FILTER option to allow sending an ICMP error if policy allows.
- for MAP, clarified that if the PCP client changed its IP address and still wants to receive traffic, it needs to send a new MAP request.

- clarified that PEER requests have to be sent from same interface as the connection itself.
- for MAP opcode, text now requires mapping be deleted when lifetime expires (per consensus on 8-Mar interim meeting)
- PEER Opcode: better description of remote peer's IP address, specifically that it does not control or establish any filtering, and explaining why it is 'from the PCP client's perspective'.
- Removed latent text allowing DMZ for 'all protocols' (protocol=0). Which wouldn't have been legal, anyway, as protocol 0 is assigned by IANA to HOPOPT (thanks to James Yu for catching that one).
- clarified that PCP server only listens on its internal interface.
- abandoned 'target' term and reverted to simplier 'internal' term.

B.12. Changes from draft-ietf-pcp-base-05 to -06

- Dual-Stack Lite: consensus was encapsulation mode. Included a suggestion that the B4 will need to proxy PCP-to-PCP and UPnP-to-PCP.
- defined THIRD_PARTY Option to work with the PEER Opcode, too. This meant moving it to its own section, and having both MAP and PEER Opcodes reference that common section.
- used "target" instead of "internal", in the hopes that clarifies internal address used by PCP itself (for sending its packets) versus the address for MAPpings.
- Options are now required to be ordered in requests, and ordering has to be validated by the server. Intent is to ease server processing of mandatory-to-implement options.
- Swapped Option values for the mandatory- and optional-to-process Options, so we can have a simple lowest..highest ordering.
- added MISORDERED_OPTIONS error.
- re-ordered some error messages to cause MALFORMED_REQUEST (which is PCP's most general error response) to be error 1, instead of buried in the middle of the error numbers.
- clarified that, after successfully using a PCP server, that PCP server is declared to be non-responsive after 5 failed retransmissions.

- 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 WiFi+Ethernet. Added discussion of using IPv6 Privacy Addresses and RFC1918 as source addresses for PCP requests. This should finish the section on multi-interface issues.

- 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.13. 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.bpw-pcp-upnp-igd-interworking].

B.14. Changes from draft-ietf-pcp-base-03 to -04

- "Pinhole" and "PIN" changed to "mapping" and "MAP".
- Reduced from four MAP Opcodes to two. This was done by implicitly using the address family of the PCP message itself.
- New option THIRD_PARTY, to more carefully split out the case where a mapping is created to a different host within the home.
- Integrated a lot of editorial changes from Stuart and Francis.
- Removed nested NAT text into another document, including the IANA-registered IP addresses for the PCP server.
- Removed suggestion (MAY) that PCP server reserve UDP when it maps TCP. Nobody seems to need that.
- Clearly added NAT and NAPT, such as in residential NATs, as within scope for PCP.
- HONOR_EXTERNAL_PORT renamed to PREFER_FAILURE
- Added 'Lifetime' field to the common PCP header, which replaces the functions of the 'temporary' and 'permanent' error types of the previous version.
- Allow arbitrary Options to be included in PCP response, so that PCP server can indicate un-supported PCP Options. Satisfies PCP Issue #19
- Reduced scope to only deal with mapping protocols that have port numbers.
- Reduced scope to not support DMZ-style forwarding.
- Clarified version negotiation.

B.15. Changes from draft-ietf-pcp-base-02 to -03

- Adjusted abstract and introduction to make it clear PCP is intended to forward ports and intended to reduce application keepalives.
- First bit in PCP common header is set. This allows DTLS and non-DTLS to be multiplexed on same port, should a future update to this specification add DTLS support.

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

B.16. Changes from draft-ietf-pcp-base-01 to -02

- more error codes
- PCP client source port number should be random
- PCP message minimum 8 octets, maximum 1024 octets.
- tweaked a lot of text in section 7.4, "Opcode-Specific Server Operation".
- opening a mapping also allows ICMP messages associated with that mapping.
- PREFER_FAILURE value changed to the mandatory-to-process range.
- added text recommending applications that are crashing obtain short lifetimes, to avoid consuming subscriber’s port quota.

B.17. Changes from draft-ietf-pcp-base-00 to -01

- Significant document reorganization, primarily to split base PCP operation from Opcode operation.
- packet format changed to move ’protocol’ outside of PCP common header and into the MAP* opcodes
- Renamed Informational Elements (IE) to Options.
- Added REMOTE_PEER (for disambiguation with dynamic ports), REMOTE_PEER_FILTER (for simple packet filtering), and

PREFER_FAILURE (to optimize UPnP IGD interworking) options.

- Is NAT or router behind B4 in scope?
- 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.
- Result code most significant bit now indicates permanent/temporary error
- PCP Options are split into mandatory-to-process ("P" bit), and into Specification Required and Private Use.
- 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-ftgroup.com

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

Email: rpenno@juniper.net

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: 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

Table of Contents

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:

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

2.2. Supported Protocols

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

2.3. Single-homed Customer Premises Network

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

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

3. Terminology

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

4. Relationship between PCP Server and its NAT/firewall

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

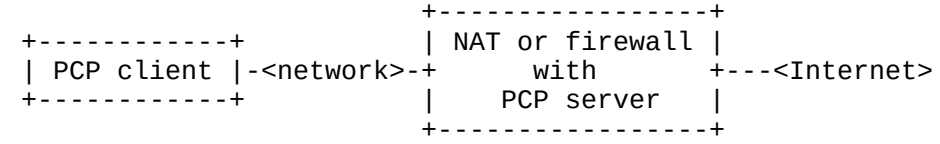


Figure 1: PCP-Enabled NAT or Firewall

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:

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

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

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

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

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

The exact times that client requests are sent are influenced by a client timing state machine taking into account 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:

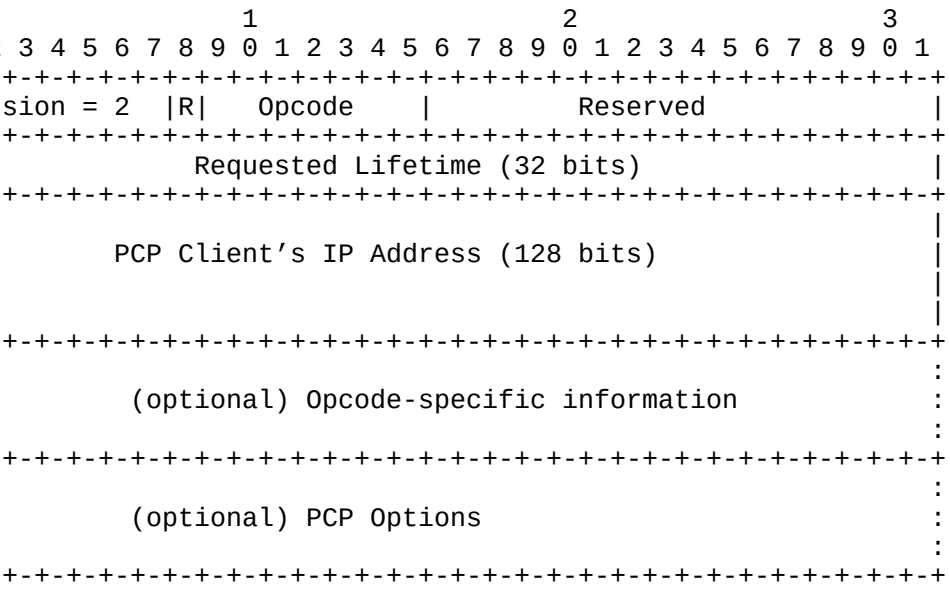


Figure 2: Common Request Packet Format

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:

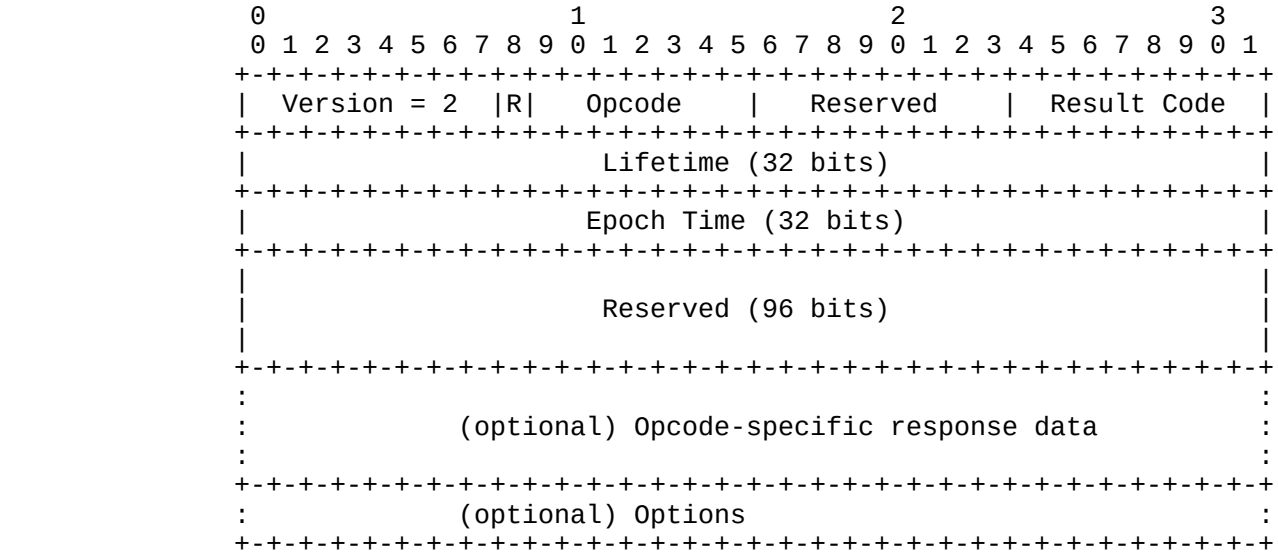


Figure 3: Common Response Packet Format

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:

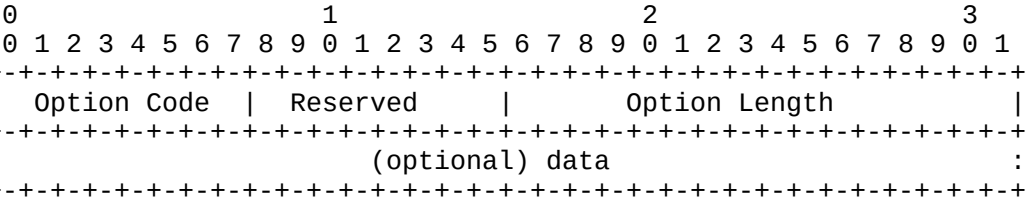


Figure 4: Options Header

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:

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

7.4. Result Codes

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

0 SUCCESS: Success.

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

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

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

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

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

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

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

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

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

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

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

See Section 13.2 for processing details. The error lifetime depends on the reason for the failure.

12 ADDRESS_MISMATCH: The source IP address of the request packet does not match the contents of the PCP Client's IP Address field, due to an unexpected NAT on the path between the PCP client and the PCP-controlled NAT or firewall. This is a long lifetime error.

13 EXCESSIVE_REMOTE_PEERS: The PCP server was not able to create the filters in this request. This result code MUST only be returned if the MAP request contained the FILTER Option. See Section 13.3 for processing information. This is a long lifetime error.

8. General PCP Operation

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

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

8.1. General PCP Client: Generating a Request

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

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

1. if a PCP server is configured (e.g., in a configuration file or via DHCP), that single configuration source is used as the list of PCP Server(s), else;

2. the default router list (for IPv4 and IPv6) is used as the list of PCP Server(s). Thus, if a PCP client has both an IPv4 and IPv6 address, it will have an IPv4 PCP server (its IPv4 default router) for its IPv4 mappings, and an IPv6 PCP server (its IPv6 default router) for its IPv6 mappings.

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

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

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

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

8.1.1. PCP Client Retransmission

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

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

RT: Retransmission timeout, calculated as described below

IRT: Initial retransmission time, SHOULD be 3 seconds

MRC: Maximum retransmission count, SHOULD be 0 (0 indicates no maximum)

MRT: Maximum retransmission time, SHOULD be 1024 seconds

MRD: Maximum retransmission duration, SHOULD be 0 (0 indicates no maximum)

RAND: Randomization factor, calculated as described below

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

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

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

The RT value is initialized based on IRT:

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

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

```
RT = (1 + RAND) * MIN (2 * RTprev, MRT)
```

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

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

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

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

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

8.2. General PCP Server: Processing a Request

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

If the length of the message exceeds 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:

- 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
- 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:

- 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.

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

- The "+2" in the calculations above is to accommodate quantization errors in client and server clocks (up to one second quantization error each in server and client time intervals).

- The "/16" in the calculations above is to accommodate inaccurate clocks in low-cost devices. This allows for a total discrepancy of up to 1/16 (6.25%) to be considered benign, e.g., if one clock were to run too fast by 3% while the other clock ran too slow by 3% then the client would not consider this difference to be anomalous or indicative of a restart having occurred. This tolerance is strict enough to be effective at detecting reboots, while not being so strict as to generate false alarms.

9. Version Negotiation

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

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

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

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

2. If the server supports that version it responds normally.

3. If the server does not support that version it replies giving a result containing the result code UNSUPP_VERSION, and the closest version number it does support (if the server supports a range of versions higher than the client's requested version, the server returns the lowest of that supported range; if the server supports a range of versions lower than the client's requested version, the server returns the highest of that supported range).

4. If the client receives an UNSUPP_VERSION result containing a version it does support, it records this fact and proceeds to use this message version for subsequent communication with this PCP server (until a possible future UNSUPP_VERSION response if the server is later updated, at which point the version negotiation process repeats).

5. If the client receives an UNSUPP_VERSION result containing a version it does not support then the client SHOULD try the next-lower version supported by the client. The attempt to use the next-lower version repeats until the client has tried version 2. If using version 2 fails, the client MAY log an error message to inform the user that it is too old to work with this server, and the client SHOULD set a timer to retry its request in 30 minutes or the returned Lifetime value, whichever is smaller. By automatically retrying in 30 minutes, the protocol accommodates an upgrade of the PCP server.

10. Introduction to MAP and PEER Opcodes

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

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

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

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

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

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

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

The 'external' address is the Suggested External Address field of the MAP or PEER request, and is address family is usually the same as the 'internal' address family, except when technologies like NAT64 are used.

The 'remote peer' address is the Remote Peer IP Address of the PEER request or the FILTER option of the MAP request, and is always the same address family as the 'internal' address, even when NAT64 is used.

In NAT64, the IPv6 PCP client is not necessarily aware of the NAT64 or aware of the actual IPv4 address of the remote peer, so it expresses the IPv6 address from its perspective, as shown in the table.

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

Figure 5: Address Families with MAP and PEER

10.1. For Operating a Server

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

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

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

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

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

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

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

     if (received_incoming_connection_or_packet())
         process_it(s);

     if (other_work_to_do())
         do_it();

     /* ... */

     block_until_we_need_to_do_something_else();
     }
```

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

10.2. For Operating a Symmetric Client/Server

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

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

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

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

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

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

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

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

     if (received_incoming_connection_or_packet())
         process_it(s);

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

     if (data_to_send())
         send_it(s);

     if (other_work_to_do())
         do_it();

     /* ... */

     block_until_we_need_to_do_something_else();
     }
```

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

10.3. For Reducing NAT or Firewall Keepalive Messages

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

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

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

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

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

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

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

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

     if (data_to_send())
         send_it(s);

     if (other_work_to_do())
         do_it();

     /* ... */

     block_until_we_need_to_do_something_else();
     }
```

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

10.4. For Restoring Lost Implicit TCP Dynamic Mapping State

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

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

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

11. MAP Opcode

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

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

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

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

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

On Operating Systems that allow multiple listening servers to bind to

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

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

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

11.1. MAP Operation Packet Formats

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

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

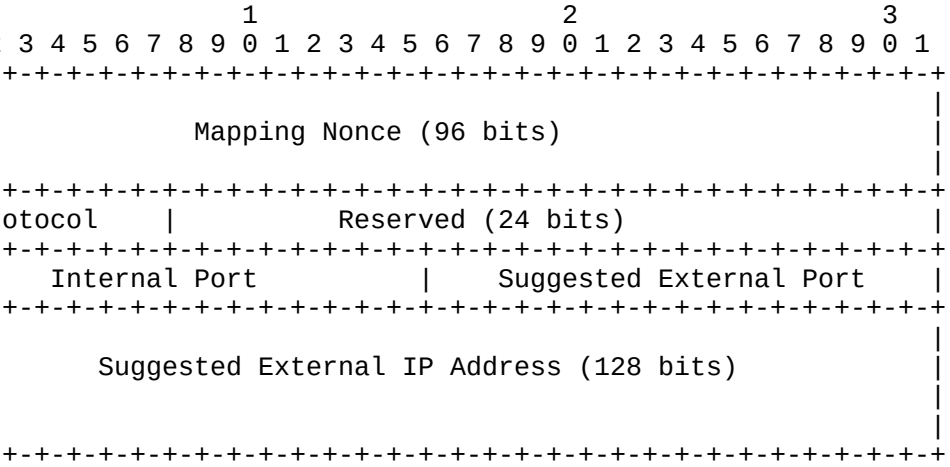


Figure 9: MAP Opcode Request

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:

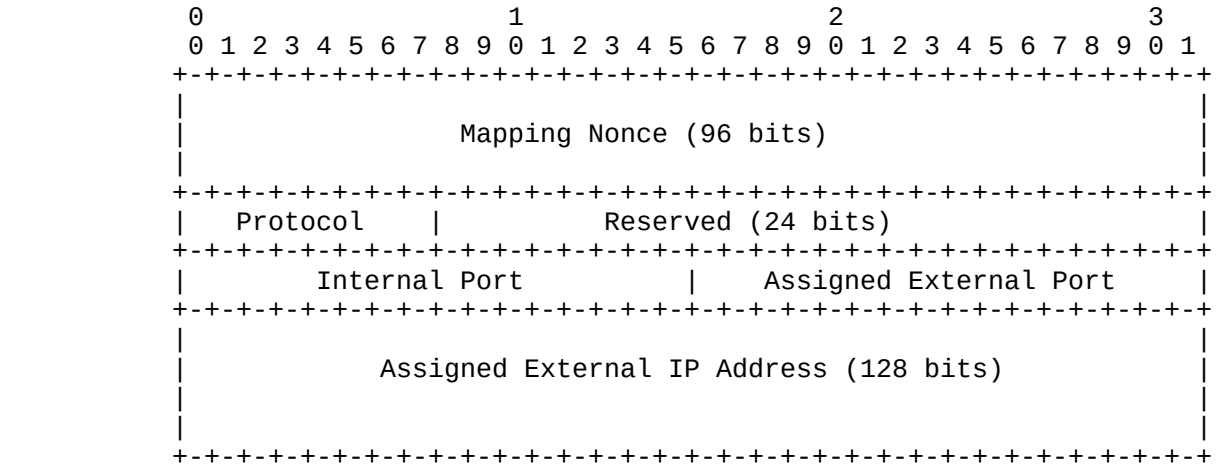


Figure 10: MAP Opcode Response

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:

- 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.

- 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.

- 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.

- If the protocol is zero and the internal port is non-zero, then the request is invalid and the PCP Server MUST return a MALFORMED_REQUEST error to the client.

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

Further processing of the lifetime is described in Section 15.

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

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

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

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

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

1. If the MAP request contains the PREFER_FAILURE Option, but the Suggested External Address and Port do not match the External Address and Port of the existing mapping, the PCP server MUST return CANNOT_PROVIDE_EXTERNAL.

2. If the existing mapping is static (created outside of PCP), the PCP server MUST return the External Address and Port of the existing mapping in its response and SHOULD indicate a Lifetime of 2^32-1 seconds, regardless of the Suggested External Address and Port in the request.

3. If the existing mapping is explicit dynamic inbound (created by a previous MAP request), the PCP server MUST return the existing External Address and Port in its response, regardless of the Suggested External Address and Port in the request. Additionally, the PCP server MUST update the lifetime of the existing mapping, in accordance with section 10.5.

4. If the existing mapping is dynamic outbound (created by outgoing traffic or a previous PEER request), the PCP server SHOULD create a new explicit inbound mapping, replicating the ports and addresses from the outbound mapping (but the outbound mapping continues to exist, and remains in effect if the explicit inbound mapping is later deleted).

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

- The Suggested External Address, Protocol, and Port is already assigned to another existing explicit or implicit mapping (i.e., is already forwarding traffic to some other internal address and port).

- The Suggested External Address, Protocol, and Port is already used by the NAT gateway for one of its own services. For example, TCP port 80 for the NAT gateway's own configuration web pages, or UDP ports 5350 and 5351, used by PCP itself. A PCP server MUST NOT create client mappings for External UDP ports 5350 or 5351.

- The Suggested External Address, Protocol, and Port is otherwise prohibited by the PCP server's policy.

- The Suggested External IP Address, Protocol, or Suggested Port are invalid or invalid combinations (e.g., External Address 127.0.0.1,

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

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

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

- If the request contained the PREFER_FAILURE Option, then the PCP server MUST return CANNOT_PROVIDE_EXTERNAL.
- 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:

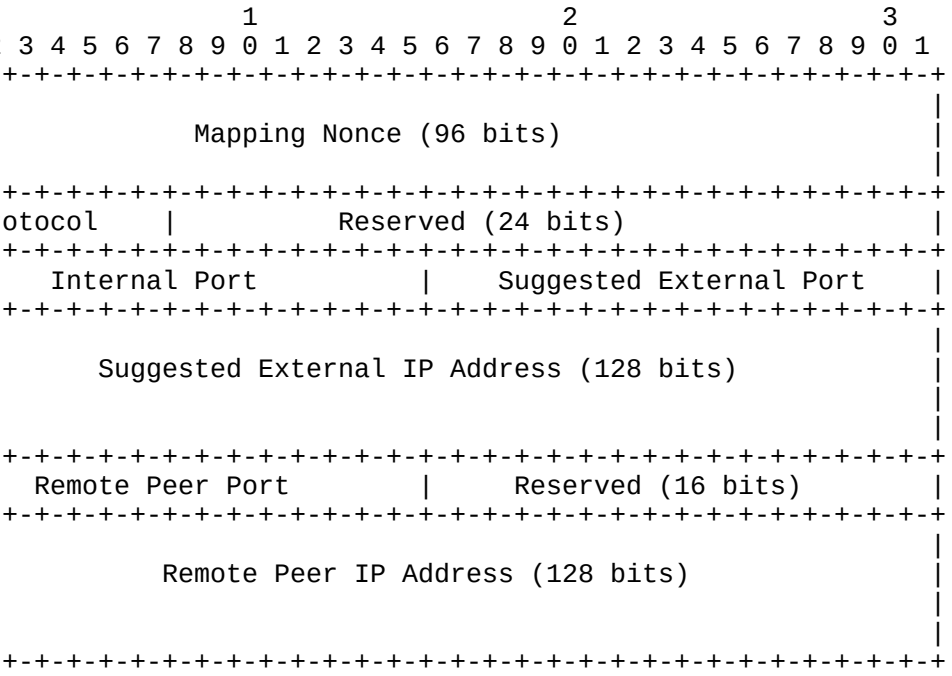


Figure 11: PEER Opcode Request

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:

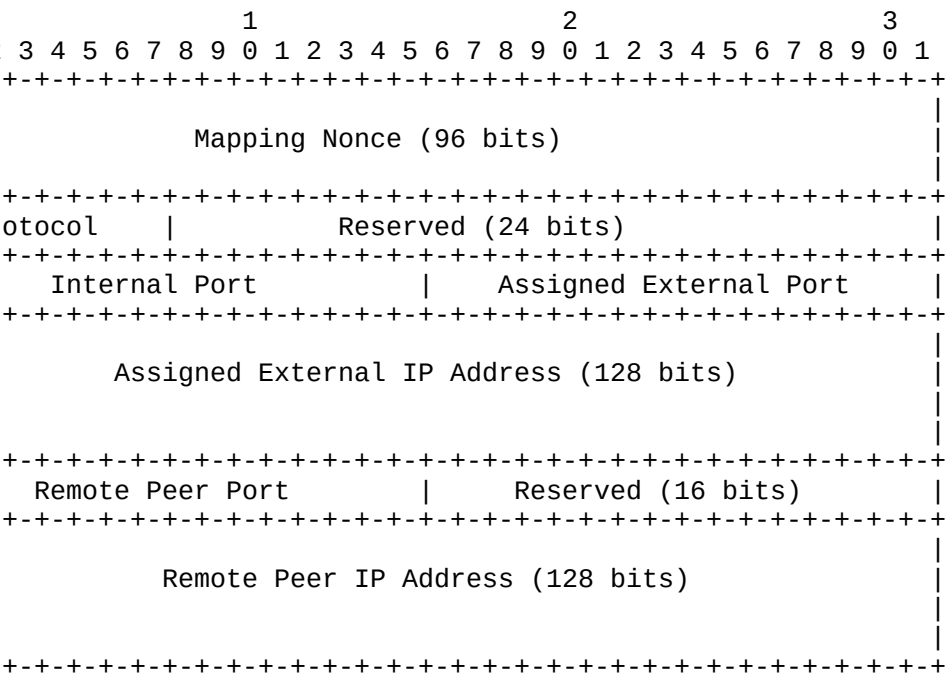


Figure 12: PEER Opcode Response

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 pervious PEER Opcode was successfully processed for this mapping, then the Suggested External Address and Port values in the request are ignored, Lifetime of that mapping is adjusted as described below, and information about the existing mapping is returned. This allows a client to explicitly extend the lifetime of an existing mapping and/or to learn an existing mapping’s External Address, Port and lifetime. The Mapping

Nonce is remembered for this mapping.

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

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

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

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

12.4. Processing a PEER Response

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

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

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

On a successful response, the application can use the assigned

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

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

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

13. Options for MAP and PEER Opcodes

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

13.1. THIRD_PARTY Option for MAP and PEER Opcodes

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

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

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

The THIRD_PARTY Option is formatted as follows:

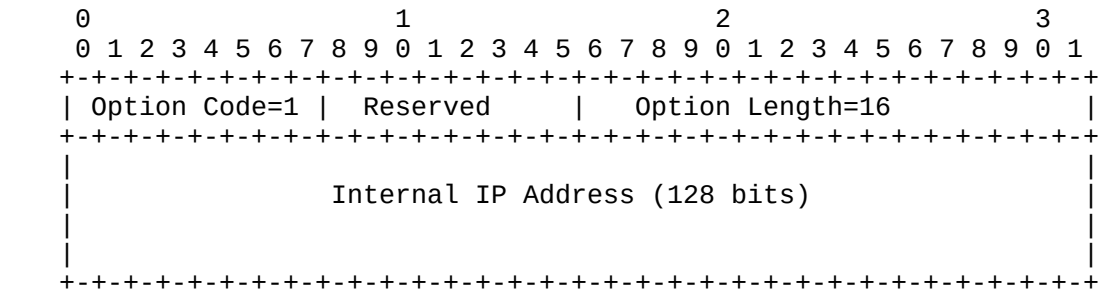


Figure 13: THIRD_PARTY Option

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:

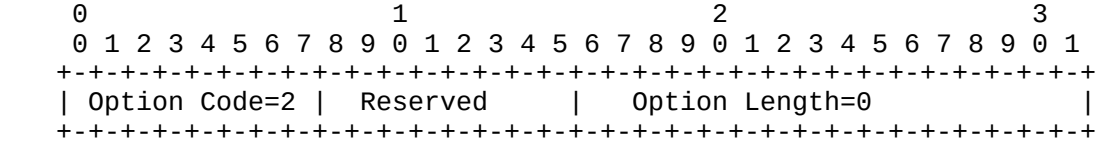


Figure 14: PREFER_FAILURE Option

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:

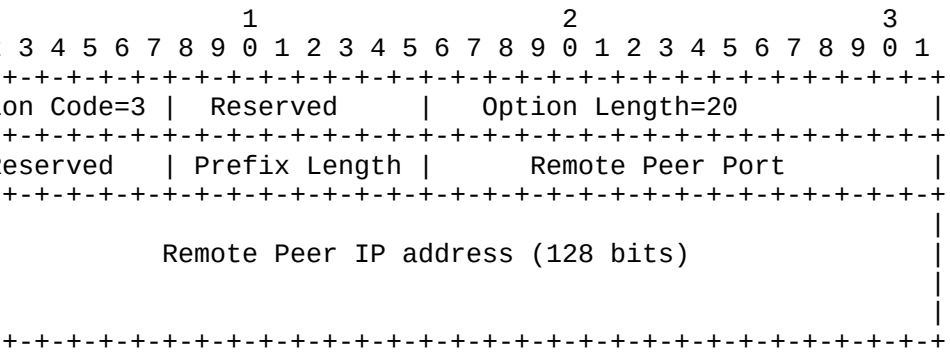


Figure 15: FILTER Option layout

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

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

Options with the ANNOUNCE Opcode.

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

14.1.2. Generating and Processing a Solicited ANNOUNCE Message

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

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

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

14.1.3. Generating and Processing an Unsolicited ANNOUNCE Message

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

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

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

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

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

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

14.2. PCP Mapping Update

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

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

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

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

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

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

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

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

15. Mapping Lifetime and Deletion

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

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

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

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

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

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

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

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

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

15.1. Lifetime Processing for the MAP Opcode

If the the requested lifetime is zero then:

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

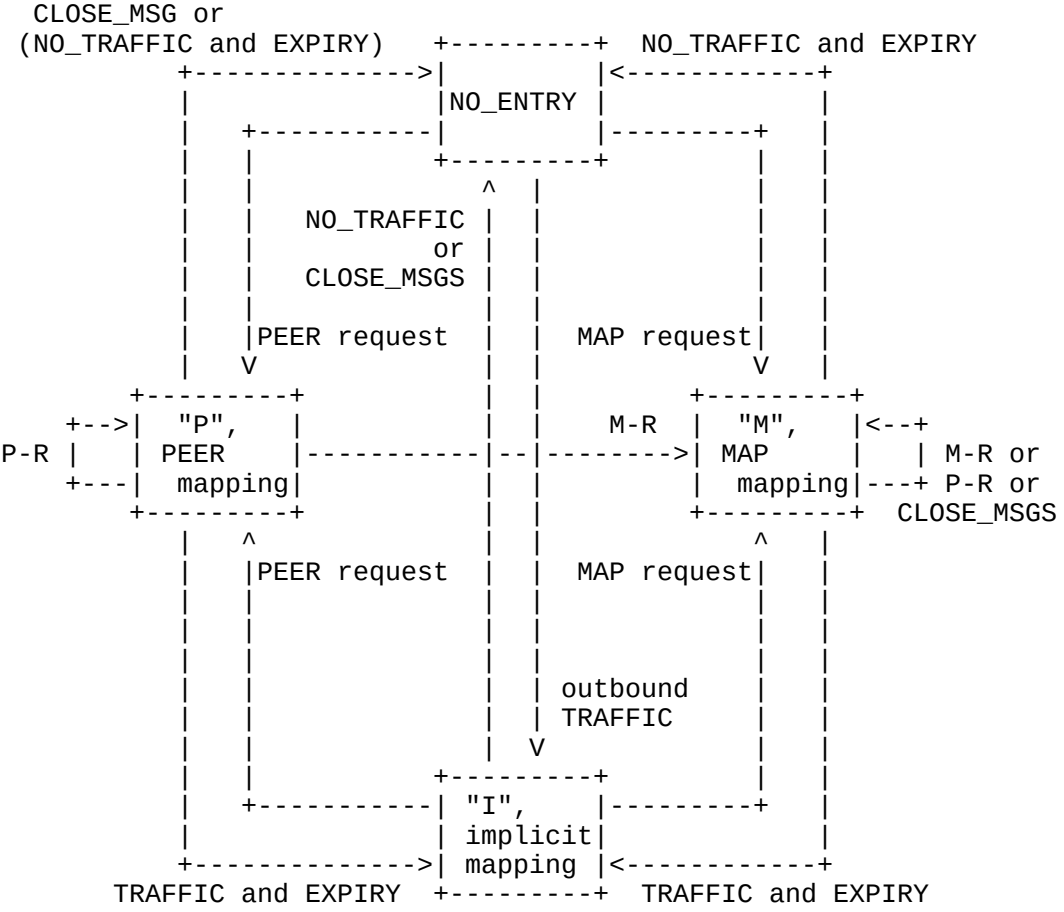


Figure 16: PCP State Diagram

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

- If you allow multiple administrative domains to send PCP requests to a single PCP server that does not enforce a boundary between the domains, it is possible for a node in one domain to perform a denial of service attack on other domains, or to capture traffic that is intended for a node in another domain.

- If explicit mappings have longer lifetimes than implicit mappings, it makes it easier to perpetrate a denial of service attack than it would be if the PCP Server was not present.

- If the PCP Server supports deleting or reducing the lifetime of existing mappings, this allows an attacking node to steal an existing mapping and receive traffic that was intended for another node.

- If the THIRD_PARTY Option is supported, this also allows an attacker to open a window for an external node to attack an internal node, allows an attacker to steal traffic that was intended for another node, or may facilitate a denial of service attack. One example of how the THIRD_PARTY Option could grant an attacker more capability than a spoofed implicit mapping is that the PCP server (especially if it is running in a service provider’s network) may not be aware of internal filtering that would prevent spoofing an equivalent implicit mapping, such as filtering between a guest and corporate network.

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

18.1.2. Deployment Examples Supporting the Simple Threat Model

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

18.1.2.1. Residential Gateway Deployment

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

18.2. Advanced Threat Model

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

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

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

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

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

18.3. Residual Threats

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

18.3.1. Denial of Service

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

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

18.3.2. Ingress Filtering

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

18.3.3. Mapping Theft

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

by using a different IP address pool).

18.3.4. Attacks Against Server Discovery

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

19. IANA Considerations

IANA is requested to perform the following actions:

19.1. Port Number

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

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

19.2. Opcodes

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

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

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

19.3. Result Codes

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

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

19.4. Options

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

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

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

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

20. Acknowledgments

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

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

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

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

Thanks to Peter Lothberg for clock skew information.

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

Thanks to authors of DHCPv6 for retransmission text.

21. References

21.1. Normative References

[RFC0768] Postel, J., "User Datagram Protocol", STD 6, RFC 768, August 1980.

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

[RFC2827] Ferguson, P. and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing", BCP 38, RFC 2827, May 2000.

[RFC4086] Eastlake, D., Schiller, J., and S. Crocker, "Randomness Requirements for Security", BCP 106, RFC 4086, June 2005.

[RFC4193] Hinden, R. and B. Haberman, "Unique Local IPv6 Unicast Addresses", RFC 4193, October 2005.

[RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, February 2006.

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.

[RFC6056] Larsen, M. and F. Gont, "Recommendations for Transport-Protocol Port Randomization", BCP 156, RFC 6056, January 2011.

[proto_numbers] IANA, "Protocol Numbers", 2011, <http://www.iana.org/assignments/protocol-numbers/protocol-numbers.xml>.

21.2. Informative References

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

[I-D.cheshire-dnsext-dns-sd]
Cheshire, S. and M. Krochmal, "DNS-Based Service Discovery", draft-cheshire-dnsext-dns-sd-11 (work in progress), December 2011.

[I-D.cheshire-nat-pmp]
Cheshire, S. and M. Krochmal, "NAT Port Mapping Protocol (NAT-PMP)", draft-cheshire-nat-pmp-05 (work in progress), September 2012.

[I-D.ietf-behave-lsn-requirements]
Perreault, S., Yamagata, I., Miyakawa, S., Nakagawa, A., and H. Ashida, "Common requirements for Carrier Grade NATs (CGNs)", draft-ietf-behave-lsn-requirements-09 (work in progress), August 2012.

[I-D.ietf-behave-sctpnat]
Stewart, R., Tuexen, M., and I. Ruengeler, "Stream Control Transmission Protocol (SCTP) Network Address Translation", draft-ietf-behave-sctpnat-07 (work in progress), October 2012.

[I-D.ietf-pcp-upnp-igd-interworking]
Boucadair, M., Dupont, F., Penno, R., and D. Wing, "Universal Plug and Play (UPnP) Internet Gateway Device (IGD)-Port Control Protocol (PCP) Interworking Function", draft-ietf-pcp-upnp-igd-interworking-04 (work in progress), September 2012.

[I-D.miles-behave-l2nat]
Miles, D. and M. Townsley, "Layer2-Aware NAT", draft-miles-behave-l2nat-00 (work in progress), March 2009.

[IGDv1] UPnP Gateway Committee, "WANIPConnection:1", November 2001, <http://upnp.org/specs/gw/UPnP-gw-WANIPConnection-v1-Service.pdf>.

[RFC0793] Postel, J., "Transmission Control Protocol", STD 7, RFC 793, September 1981.

[RFC1918] Rekhter, Y., Moskowitz, R., Karrenberg, D., Groot, G., and E. Lear, "Address Allocation for Private Internets", BCP 5, RFC 1918, February 1996.

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

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

[RFC3022] Srisuresh, P. and K. Egevang, "Traditional IP Network Address Translator (Traditional NAT)", RFC 3022, January 2001.

[RFC3581] Rosenberg, J. and H. Schulzrinne, "An Extension to the Session Initiation Protocol (SIP) for Symmetric Response Routing", RFC 3581, August 2003.

[RFC3587] Hinden, R., Deering, S., and E. Nordmark, "IPv6 Global Unicast Address Format", RFC 3587, August 2003.

[RFC4303] Kent, S., "IP Encapsulating Security Payload (ESP)", RFC 4303, December 2005.

[RFC4340] Kohler, E., Handley, M., and S. Floyd, "Datagram Congestion Control Protocol (DCCP)", RFC 4340, March 2006.

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

[RFC4941] Narten, T., Draves, R., and S. Krishnan, "Privacy Extensions for Stateless Address Autoconfiguration in IPv6", RFC 4941, September 2007.

[RFC4960] Stewart, R., "Stream Control Transmission Protocol", RFC 4960, September 2007.

[RFC4961] Wing, D., "Symmetric RTP / RTP Control Protocol (RTCP)", BCP 131, RFC 4961, July 2007.

[RFC5382] Guha, S., Biswas, K., Ford, B., Sivakumar, S., and P. Srisuresh, "NAT Behavioral Requirements for TCP", BCP 142, RFC 5382, October 2008.

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

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

[RFC6146] Bagnulo, M., Matthews, P., and I. van Beijnum, "Stateful NAT64: Network Address and Protocol Translation from IPv6

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

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

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

[RFC6619] Arkko, J., Eggert, L., and M. Townsley, "Scalable Operation of Address Translators with Per-Interface Bindings", RFC 6619, June 2012.

Appendix A. NAT-PMP Transition

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

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

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

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

Appendix B. Change History

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

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

- 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

- When processing MAP request or processing PEER request, Mapping Nonce validation only applies to Basic Threat Model, and not to THIRD_PARTY.

- A maximum payload size of 1100 keeps PCP packets below IPv6's 1280 MTU limit while still allowing some room for encapsulation. This accomodates EAP over PANA over PCP (EAP needs 1020 octets, per RFC3748), should PCP authentication decide to use EAP over PANA over PCP.

- Both MAP and PEER-created mappings cannot have their lifetimes reduced beyond normal UDP/TCP timesouts.

- Disallow re-assigning External Port to same internal host.

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

- For table, reverted the NAT64 remote peer to IPv6 -- because from the IPv6 PCP client's perspective, the remote peer really is IPv6.

- "list of PCP server addresses" changed to "longer list of PCP server addresses"

- Clarify that unsolicited ANNOUNCE messages are sent from the PCP server IP address and PCP port.

- "1024 bytes" changed to "1024 octets".

- Clarify that re-transmitted requests must use same Mapping Nonce value (beginning of Section 8.1.1).

- Describe that de-synchronization that can occur (end of Section 8.1.1).

- 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.

- 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.

- 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.
- 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.
- Send Mapping Update only 3 times (used to be 10 times).
- General PCP processing now requires validating Mapping Nonce, if the opcode uses a Mapping Nonce Section 8.3.
- 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.
- Clarified SCTP NAT behavior (port numbers stay the same, causing grief).
- added EIM definition.
- Clarified Mapping Type definitions.
- PCP Client definition simplified to no longer obliquely and erroneously reference UPnP IGD.
- Clarified using network-byte order.
- Epoch time comparison now allows slight packet re-ordering.
- Encourage that when new address is assigned (e.g., DHCP) that PCP as well as non-PCP mappings be cleaned up.
- Simplified formatting of retransmission, but no normative change.
- Clarified how server chooses ports and how Suggested External Port can gently influence that decision.
- Described how PCP client can use PCP Client Address with a non-PCP-aware inner NAT (Section 8.1.)
- Clarified 1024 octet length applies to UDP payload itself, and that error responses copy 1024 of UDP payload.

- 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).

- Lifetime section was (mistakenly) a subsection of the MAP section, but referenced by both MAP and PEER. It is now a top-level section.

- 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.

- 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).

- Other minor text changes; consult diffs.

B.4. Changes from draft-ietf-pcp-base-25 to -26

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

- Improved wording of THIRD_PARTY restrictions.

- 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

- Clarified the port used by the PCP server when sending unsolicited unicast ANNOUNCE.

- Removed parenthetical comment implying ANNOUNCE was not a normal Opcode; it is a normal Opcode.

- Explain that non-PCP-speaking host-based and network-based firewalls need to allow incoming connections for MAP to work.

- 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.

- For table, the NAT64 remote peer is IPv4 (was IPv6).

- 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.

- 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.
- Removed reference to DS-Lite from Security Considerations, as part of the changes to THIRD_PARTY from IESG review.
- Rapid Recovery is now a SHOULD implement.
- Clarify behavior of PREFER_FAILURE with zeros in Suggested External Port or Address fields.
- PCP server is now more robust and insistent about informing PCP client of state changes.
- 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.
- THIRD_PARTY is now prohibited on subscriber PCP clients.

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

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

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

- 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 Clairification that an existing implicit mapping (created e.g., by TCP SYN) can become managed by a MAP request (Section 11.3.
- o Clarified the ANNOUNCE Opcode is being defined in Section 14.1, and that the length of requests (as well as responses) is zero.

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

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

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

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

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

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

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

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

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

- Fully integrated Rapid Recovery, with a separate Opcode having its own processing description.

- 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

- Removed UNPROCESSED option. Instead, unprocessed options are simply not included in responses.

- Updated terminology section for Implicit/Explicit and Outbound/ Inbound.

- PEER requests cannot delete or shorten the lifetime of a mapping.

- Clarified that PCP clients only retransmit mapping requests for as long as they actually want the mapping.

- Revised Epoch time calculations and explanation.

- Renamed the announcement opcode from No-Op to ANNOUNCE.

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

- suggest acquiring a mapping to the Discard port if there is a desire to show the user their external address (Section 11.6).

- Added Restart Announcement.

- Tweaked terminology.

- Detailed how error responses are generated.

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

- fixed mistake in PCP request format (had 32 bits of extraneous fields)

- 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

- corrected text on Client Processing a Response related to receiving ADDRESS_MISMATCH error.

- updated Epoch text.
- 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

- Softened and removed text that was normatively explaining how PEER is implemented within a NAT.
- 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.
- combined SERVER_OVERLOADED and NO_RESOURCES into one error code, NO_RESOURCES.
- SCTP mappings have to use same internal and suggested external ports, and have implied PREFER_FAILURE semantics.
- 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

- Moved discussion of socket operations for PCP source address into Implementation Considerations section.
- Integrated numerous WGLC comments.
- NPTv6 in scope.
- Re-written security considerations section. Thanks, Margaret!
- Reduced PEER4 and PEER6 Opcodes to just a single Opcode, PEER.
- Reduced MAP4 and MAP6 Opcodes to just a single Opcode, MAP.
- Rearranged the PEER packet formats to align with MAP.
- 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.
- The THIRD_PARTY Option from an unauthorized host generates UNSUPP_OPTION, so the PCP server doesn’t disclose it knows how to process THIRD_PARTY Option.

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

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

- Removed ADDRESS_MISMATCH; the server always includes its idea of the Client's IP Address and Port, and it's up to the client to detect a mismatch (and rectify it).

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

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

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

- new ADDRESS_MISMATCH error.

- removed PROCESSING_ERROR error, which was too similar to MALFORMED_REQUEST.

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

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

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

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

- clarified that MAP-created mappings have no filtering, and PEER-created mappings have whatever filtering and mapping behavior is normal for that particular NAT / firewall.

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

- allow new Options to be defined with a specification (rather than standards action)

B.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).

- 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.

- Tweaked wording of EDM NATs in Section 16.1 to clarify the problem occurs both inside->outside and outside->inside.

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

- fixed zero/non-zero text in Section 15.

- removed duplicate text saying MAP is allowed to delete an implicit dynamic mapping. It is still allowed to do that, but it didn't need to be said twice in the same paragraph.

- Renamed error from UNAUTH_TARGET_ADDRESS to UNAUTH_THIRD_PARTY_INTERNAL_ADDRESS.

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

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

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

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

- discussed NAT port-overloading and its impact on MAP (new section Section 16.1), which allowed removing the IMPLICIT_MAPPING_EXISTS error.

- eliminated NONEXIST_PEER error (which was returned if a PEER request was received without an implicit dynamic mapping already being created), and adjusted PEER so that it creates an implicit dynamic mapping.

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

- Added Client's IP Address to PCP common header. This allows server to refuse a PCP request if there is a mismatch with the source IP address, such as when a non-PCP-aware NAT was on the path. This should reduce failure situations where PCP is deployed in conjunction with a non-PCP-aware NAT. This addition was

consensus at IETF80.

- Changed UNSPECIFIED_ERROR to PROCESSING_ERROR. Clarified that MALFORMED_REQUEST is for malformed requests (and not related to failed attempts to process the request).
- Removed MISORDERED_OPTIONS. Consensus of IETF80.
- SERVER_OVERLOADED is now a common PCP error (instead of specific to MAP).
- Tweaked PCP retransmit/retry algorithm again, to allow more aggressive PCP discovery if an implementation wants to do that.
- Version negotiation text tweaked to soften NAT-PMP reference, and more clearly explain exactly what UNSUPP_VERSION should return.
- PCP now uses NAT-PMP's UDP port, 5351. There are no normative changes to NAT-PMP or PCP to allow them both to use the same port number.
- New Appendix A to discuss NAT-PMP / PCP interworking.
- improved pseudocode to be non-blocking.
- clarified that PCP cannot delete a static mapping (i.e., a mapping created by CLI or other non-PCP means).
- moved theft of mapping discussion from Epoch section to Security Considerations.

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

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

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

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

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

- defined THIRD_PARTY Option to work with the PEER Opcode, too. This meant moving it to its own section, and having both MAP and PEER Opcodes reference that common section.

- used "target" instead of "internal", in the hopes that clarifies internal address used by PCP itself (for sending its packets) versus the address for MAPpings.

- Options are now required to be ordered in requests, and ordering has to be validated by the server. Intent is to ease server processing of mandatory-to-implement options.

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

- added MISORDERED_OPTIONS error.

- re-ordered some error messages to cause MALFORMED_REQUEST (which is PCP's most general error response) to be error 1, instead of buried in the middle of the error numbers.

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

- tightened up text (which was inaccurate) about how long general PCP processing is to delay when receiving an error and if it should honor Opcode-specific error lifetime. Useful for MAP errors which have an error lifetime. (This all feels awkward to have only some errors with a lifetime.)

- Added better discussion of multiple interfaces, including highlighting 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.

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

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

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

- Server's processing of PEER now says it is implementation-specific if a PCP server continues to allow the mapping to exist after a PEER message. Client's processing of PEER says that if client

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

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

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

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

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

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

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

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

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

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

- opening a mapping also allows ICMP messages associated with that mapping.
- PREFER_FAILURE value changed to the mandatory-to-process range.
- 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

- Significant document reorganization, primarily to split base PCP operation from Opcode operation.
- packet format changed to move 'protocol' outside of PCP common header and into the MAP* opcodes
- Renamed Informational Elements (IE) to Options.
- Added REMOTE_PEER (for disambiguation with dynamic ports), REMOTE_PEER_FILTER (for simple packet filtering), and PREFER_FAILURE (to optimize UPnP IGDv1 interworking) options.
- Is NAT or router behind B4 in scope?
- 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.
- Result code most significant bit now indicates permanent/temporary error
- PCP Options are split into mandatory-to-process ("P" bit), and into Specification Required and Private Use.
- 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: March 12, 2012

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

September 09, 2011

DHCP and DHCPv6 Options for the Port Control Protocol (PCP)
draft-ietf-pcp-dhcp-00

Abstract

This document specifies DHCP (IPv4 and IPv6) options to configure hosts with Port Control Protocol (PCP) Server addresses. The use of IPv4 DHCP 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 March 12, 2012.

Copyright Notice

Table of Contents

1. Introduction

This document defines IPv4 DHCP [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 Fully Qualified Domain Name (FQDN, 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 (see Section 6.1.1 of [RFC1123]) are available on the host client.

The use of IPv4 DHCP or DHCPv6 depends on the PCP deployment scenarios.

2. Terminology

This document makes use of the following terms:

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

3. Rationale

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

In addition to the operational considerations:

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

- For the DS-Lite case [RFC6333], if the encapsulation mode is used to send PCP messages, an IP address may be used since the AFTR selection is already done via the AFTR_NAME DHCPv6 option [RFC6334]. Of course, this assumes that the PCP Server is co-located with the AFTR function. If these functions are not co-located, conveying the FQDN would be more convenient.

If the PCP Server is located in a LAN, a simple FQDN such as "pcp-server.local" can be used.

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. 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.

5.1. Format

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

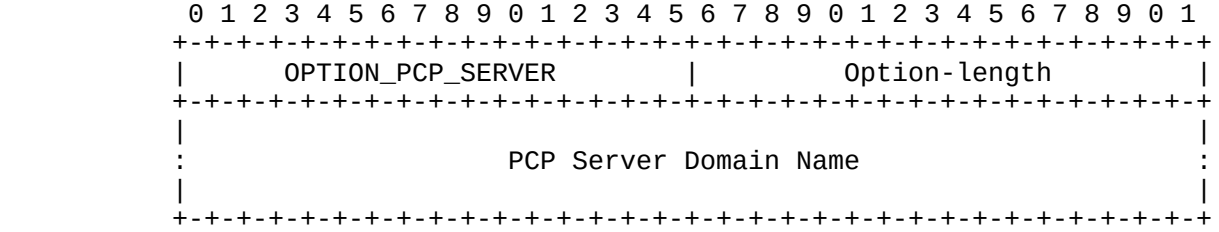


Figure 1: PCP Server FQDN DHCPv6 Option

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

- Option-code: OPTION_PCP_SERVER (TBA, see Section 9)
- Option-length: Length of the 'PCP Server Domain Name' field in octets.
- PCP Server Domain Name: The domain name of the PCP Server to be used by the PCP Client. The domain name is encoded as specified in Section 8 of [RFC3315]. Any possible future updates to Section 8 of [RFC3315] also apply to this option.

5.2. 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, only the first instance of that option MUST be used.

Upon receipt of an OPTION_PCP_SERVER option, the DHCPv6 client MUST verify that the option length does not exceed 255 octets [RFC1035]). The DHCPv6 client MUST verify the FQDN is a properly encoded as detailed in Section 8 of [RFC3315].

Once the FQDN conveyed in a OPTION_PCP_SERVER option is validated, the included Name is passed to the name resolution library (see Section 6.1.1 of [RFC1123] or [RFC6055]) to retrieve the corresponding IP address (IPv4 or IPv6). If more than one IPv6/IPv4 address are retrieved, the PCP Client MUST use the procedure defined in [I-D.ietf-pcp-base] for address selection.

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.

5.3. Server Behaviour

A DHCPv6 server MUST NOT reply with a value for the OPTION_PCP_SERVER if the DHCPv6 client has not explicitly included OPTION_PCP_SERVER in its OPTION_ORO.

If OPTION_PCP_SERVER option is requested by the DHCPv6 client, the DHCPv6 server MUST NOT send more than one OPTION_PCP_SERVER option in the response. The DHCPv6 server MUST include only one FQDN in a OPTION_PCP_SERVER option. The DHCPv6 server MUST NOT include an FQDN having a length exceeding 255 octets.

6. IPv4 DHCP PCP Option

6.1. Format

The PCP Server IPv4 DHCP option can be used to configure a FQDN to be used by the PCP Client to contact a PCP Server. The generic format of this option is illustrated in Figure 2.

Because of the depletion of IPv4 DHCP option codes and in order to anticipate future PCP-related IPv4 DHCP options, the proposed option uses a sub-option field.

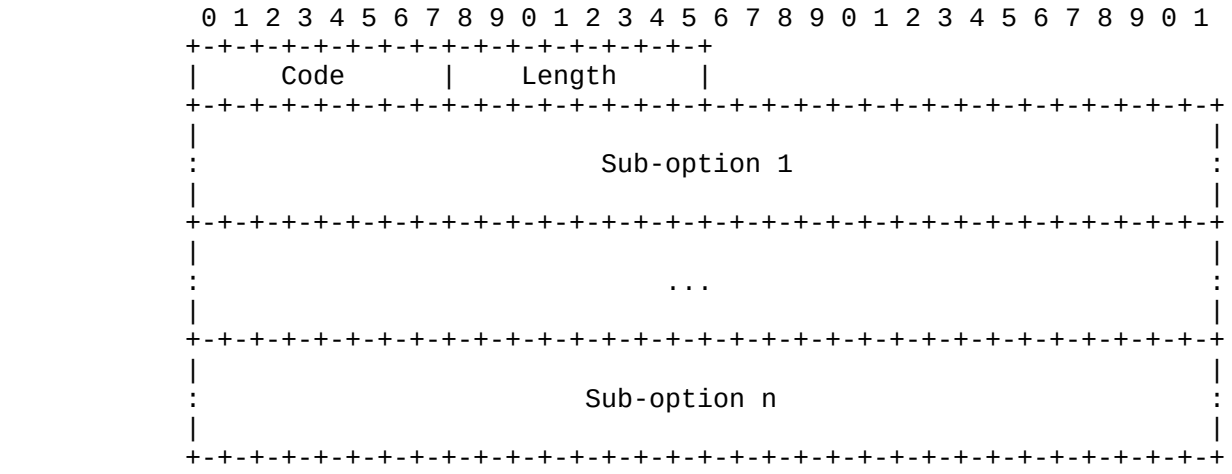


Figure 2: IPv4 DHCP PCP Option

The description of the fields is as follows:

- Code: OPTION_PCP_SERVER (TBA, see Section 9);
- Length: Includes the length of included sub-options in octets; The maximum length is 255 octets.
- One or several sub-options can be included in a PCP IPv4 DHCP option. The format of each sub-option follows the structure shown in Figure 3.

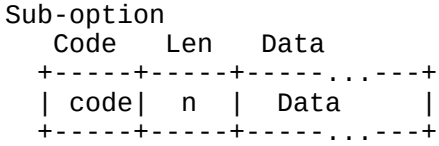


Figure 3: PCP Server sub-option

Only one sub-option is defined in this document:

1: PCP Server Domain Name Sub-option (OPTION_PCP_SERVER_D (Figure 4)). This sub-option includes an FQDN of the PCP Server to be used by the PCP Client when issuing PCP messages.

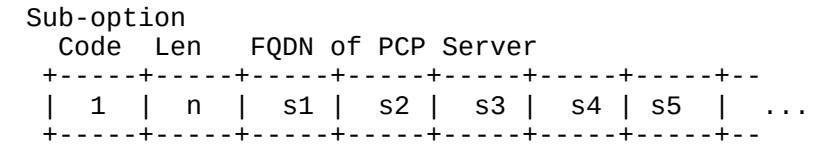


Figure 4: PCP Server FQDN DHCP Sub-option

The fields of the PCP Server Domain Name sub-option shown in Figure 4 are:

- Sub-option Code: 1.
- Len: Length of the "PCP Server Domain Name" field in octets.
- PCP Server Domain Name: The domain name of the PCP Server to be used by the PCP Client. The encoding of the domain name is described in Section 3.1 of [RFC1035].

A side effect of having the sub-option format is the risk to have a large option exceeding the maximum permissible within a single option (254 octets + the length octets). In such case, it is RECOMMENDED to use [RFC3396].

6.2. Server Behaviour

IPv4 DHCP server MUST NOT provide this option, unless the client requested it in Parameter Request List Option.

If OPTION_PCP_SERVER option is requested by the IPv4 DHCP client, the IPv4 DHCP server MUST NOT send more than one OPTION_PCP_SERVER option and more than one OPTION_PCP_SERVER_D sub-option in the response. The IPv4 DHCP server MUST include only one FQDN in a OPTION_PCP_SERVER_D sub-option.

6.3. Client Behaviour

IPv4 DHCP client expresses the intent to get OPTION_PCP_SERVER by specifying it in Parameter Request List Option [RFC2132].

If the IPv4 DHCP client receives more than one OPTION_PCP_SERVER option from the IPv4 DHCP server, only the first instance of that option MUST be used. If the selected OPTION_PCP_SERVER includes more than one OPTION_PCP_SERVER_D sub-option, only the first instance of that option MUST be used.

When the PCP Server Domain Name Sub-option is used, the client

invokes the underlying name resolution library (see Section 6.1.1 of [RFC1123] or [RFC6055]) to retrieve the IPv4 address(es) of the PCP server(s).

7. Dual-Stack Hosts

A PCP Server configured using OPTION_PCP_SERVER over IPv4 DHCP 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 IPv4 DHCP 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.

8. Security Considerations

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

9. IANA Considerations

Authors of this document request the following DHCPv6 option code:

OPTION_PCP_SERVER

Authors of this document request the following IPv4 DHCP option code:

OPTION_PCP_SERVER

Authors of this document request also to create a sub-option registry for OPTION_PCP_SERVER option; a code for the following sub-option is requested:

OPTION_PCP_SERVER_D

10. Acknowledgements

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

11. References

11.1. 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-13 (work in progress), July 2011.

[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.

[RFC3396] Lemon, T. and S. Cheshire, "Encoding Long Options in the Dynamic Host Configuration Protocol (DHCPv4)", RFC 3396, November 2002.

[RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, February 2006.

11.2. Informative References

[I-D.ietf-behave-lsn-requirements] Perreault, S., Yamagata, I., Miyakawa, S., Nakagawa, A., and H. Ashida, "Common requirements for Carrier Grade NAT (CGN)", draft-ietf-behave-lsn-requirements-03 (work in progress), August 2011.

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

[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-ftgroup.com

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

Email: rpenno@juniper.net

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

Email: dwing@cisco.com

PCP Working Group
Internet-Draft
Intended status: Standards Track
Expires: October 16, 2014

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

April 14, 2014

DHCP Options for the Port Control Protocol (PCP)
draft-ietf-pcp-dhcp-13

Abstract

This document specifies DHCP (IPv4 and IPv6) options to configure hosts with Port Control Protocol (PCP) server IP addresses. The use of DHCPv4 or DHCPv6 depends on the PCP deployment scenarios. The set of deployment scenarios to which use of DHCPv4 or DHCPv6 apply are outside the scope of this document.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

Status of This Memo

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

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

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

This Internet-Draft will expire on October 16, 2014.

Copyright Notice

Table of Contents

1. Introduction

This document defines DHCPv4 [RFC2131] and DHCPv6 [RFC3315] options that can be used to configure hosts with PCP server [RFC6887] IP addresses.

This specification assumes a PCP server is reachable with one or multiple IP addresses. As such, a list of IP addresses can be returned in the DHCP PCP server option.

This specification allows returning one or multiple lists of PCP server IP addresses. This is used as a hint to guide the PCP client when determining whether to send PCP requests to one or multiple PCP servers. Concretely, the PCP client needs an indication to decide whether entries need to be instantiated in all PCP servers (e.g.,

multi-homing, multiple PCP-controlled devices providing distinct services , etc.) or using one IP address from the list (e.g., redundancy group scenario, proxy-based model, etc.). Refer to [I-D.boucadair-pcp-deployment-cases] for a discussion on PCP deployment scenarios.

For guidelines on how a PCP client can use multiple IP addresses and multiple PCP servers, see [I-D.ietf-pcp-server-selection].

2. Terminology

This document makes use of the following terms:

- PCP server denotes a functional element that receives and processes PCP requests from a PCP client. A PCP server can be co-located with or be separated from the function (e.g., NAT, Firewall) it controls. Refer to [RFC6887].
- PCP client denotes a PCP software instance responsible for issuing PCP requests to a PCP server. Refer to [RFC6887].
- DHCP refers to both DHCPv4 [RFC2131] and DHCPv6 [RFC3315].
- DHCP client denotes a node that initiates requests to obtain configuration parameters from one or more DHCP servers.
- DHCP server refers to a node that responds to requests from DHCP clients.

3. DHCPv6 PCP Server Option

3.1. Format

The DHCPv6 PCP server option can be used to configure a list of IPv6 addresses of a PCP server.

The format of this option is shown in Figure 1.

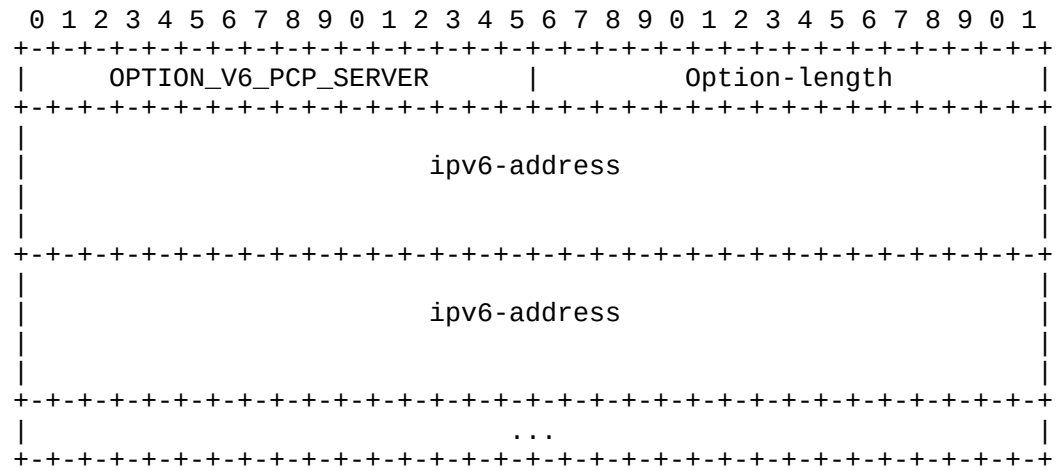


Figure 1: DHCPv6 PCP server option

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

- Option-code: OPTION_V6_PCP_SERVER (TBA, see Section 9.1)
- Option-length: Length of the 'PCP server IP Address(es)' field in octets. MUST be a multiple of 16.
- PCP server IPv6 Addresses: Includes one or more IPv6 addresses [RFC4291] of the PCP server to be used by the PCP client. Note, IPv4-mapped IPv6 addresses (Section 2.5.5.2 of [RFC4291]) are allowed to be included in this option.

To return more than one PCP server to the DHCPv6 client (as opposed to more than one address for a single PCP server), the DHCPv6 server returns multiple instances of OPTION_V6_PCP_SERVER.

3.2. DHCPv6 Client Behavior

To discover one or more PCP servers, the DHCPv6 client requests PCP server IP addresses by including OPTION_V6_PCP_SERVER in an Option Request Option (ORO), as described in Section 22.7 of [RFC3315].

The DHCPv6 client MUST be prepared to receive multiple instances of OPTION_V6_PCP_SERVER; each instance is to be treated as a separate PCP server.

If an IPv4-mapped IPv6 address is received in OPTION_V6_PCP_SERVER, it indicates that the PCP server has the corresponding IPv4 address.

Note: When presented with the IPv4-mapped prefix, current versions of Windows and Mac OS generate IPv4 packets, but will not send IPv6 packets [RFC6052]. Representing IPv4 addresses as IPv4-mapped IPv6 addresses follows the same logic as in section 5 of [RFC6887].

The DHCPv6 client MUST silently discard multicast and host loopback addresses [RFC6890] conveyed in OPTION_V6_PCP_SERVER.

4. DHCPv4 PCP Option

4.1. Format

The DHCPv4 PCP server option can be used to configure a list of IPv4 addresses of a PCP server. The format of this option is illustrated in Figure 2.

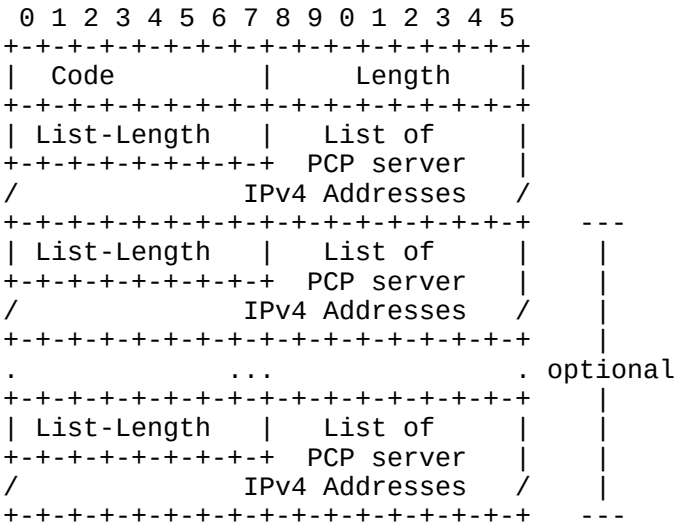
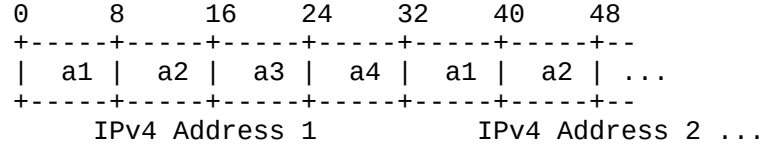


Figure 2: DHCPv4 PCP server option

The description of the fields is as follows:

- Code: OPTION_V4_PCP_SERVER (TBA, see Section 9.2);
- Length: Length of all included data in octets. The minimum length is 5.
- List-Length: Length of the "List of PCP server IPv4 Addresses" field in octets; MUST be a multiple of 4.

- o List of PCP server IPv4 Addresses: Contains one or more IPv4 addresses of the PCP server to be used by the PCP client. The format of this field is shown in Figure 3.
- o OPTION_V4_PCP_SERVER can include multiple lists of PCP server IPv4 addresses; each list is treated as a separate PCP server. When several lists of PCP server IPv4 addresses are to be included, "List-Length" and "PCP server IPv4 Addresses" fields are repeated.



This format assumes that an IPv4 address is encoded as a1.a2.a3.a4.

Figure 3: Format of the List of PCP server IPv4 Addresses

OPTION_V4_PCP_SERVER is a concatenation-requiring option. As such, the mechanism specified in [RFC3396] MUST be used if OPTION_V4_PCP_SERVER exceeds the maximum DHCPv4 option size of 255 octets.

4.2. DHCPv4 Client Behavior

To discover one or more PCP servers, the DHCPv4 client requests PCP server IP addresses by including OPTION_V4_PCP_SERVER in a Parameter Request List Option [RFC2132].

The DHCPv4 client MUST be prepared to receive multiple lists of PCP server IPv4 addresses in the same DHCPv4 PCP server option; each list is to be treated as a separate PCP server.

The DHCPv4 client MUST silently discard multicast and host loopback addresses [RFC6890] conveyed in OPTION_V4_PCP_SERVER.

5. DHCP Server Configuration Guidelines

DHCP servers supporting the DHCP PCP server option can be configured with a list of IP addresses of the PCP server(s). If multiple IP addresses are configured, the DHCP server MUST be explicitly configured whether all or some of these addresses refer to:

1. the same PCP server: the DHCP server returns multiple addresses in the same instance of the DHCP PCP server option.

2. distinct PCP servers: the DHCP server returns multiple lists of PCP server IP addresses to the requesting DHCP client (encoded as

multiple OPTION_V6_PCP_SERVER or in the same OPTION_V4_PCP_SERVER); each list is referring to a distinct PCP server. For example, multiple PCP servers may be configured to a PCP client in some deployment contexts such as multi-homing. It is out of scope of this document to enumerate all deployment scenarios that require multiple PCP servers to be returned.

Precisely how DHCP servers are configured to separate lists of IP addresses according to which PCP server they address is out of scope for this document. However, DHCP servers MUST NOT combine the IP addresses of multiple PCP servers and return them to the DHCP client as if they belong to a single PCP server, and DHCP servers MUST NOT separate the addresses of a single PCP server and return them as if they belonged to distinct PCP servers. For example, if an administrator configures the DHCP server by providing a Fully Qualified Domain Name (FQDN) for a PCP server, even if that FQDN resolves to multiple addresses, the DHCP server MUST deliver them within a single server address block.

DHCPv6 servers that implement this option and that can populate the option by resolving FQDNs will need a mechanism for indicating whether to query for A records or only AAAA records. When a query returns A records, the IP addresses in those records are returned in the DHCPv6 response as IPv4-mapped IPv6 addresses.

Discussion: The motivation for this design is to accommodate deployment cases where an IPv4 connectivity service is provided while only DHCPv6 is in use (e.g., an IPv4-only PCP server in a DS-Lite context [RFC6333]).

Since this option requires support for IPv4-mapped IPv6 addresses, a DHCPv6 server implementation will not be complete if it does not query for A records and represent any that are returned as IPv4-mapped IPv6 addresses in DHCPv6 responses. This behavior is neither required nor suggested for DHCPv6 options in general: it is specific to OPTION_V6_PCP_SERVER. The mechanism whereby DHCPv6 implementations provide this functionality is beyond the scope of this document.

For guidelines on providing context-specific configuration information (e.g., returning a regional-based configuration), and information on how a DHCP server might be configured with FQDNs that get resolved on demand, see [I-D.ietf-dhc-topo-conf].

6. Dual-Stack Hosts

A Dual-Stack host might receive PCP server option via both DHCPv4 and DHCPv6. For guidance on how a DHCP client can handle PCP server IP lists for the same network but obtained via different mechanisms, see [I-D.ietf-pcp-server-selection].

7. Hosts with Multiple Interfaces

A host may have multiple network interfaces (e.g, 3G, IEEE 802.11, etc.); each configured differently. Each PCP server learned MUST be associated with the interface via which it was learned.

Refer to [I-D.ietf-pcp-server-selection] and Section 8.4 of [RFC6887] for more discussion on multi-interface considerations.

8. Security Considerations

The security considerations in [RFC2131] and [RFC3315] are to be considered. PCP-related security considerations are discussed in [RFC6887].

The PCP Server option targets mainly the simple threat model (Section 18.1 of [RFC6887]). It is out of scope of this document to discuss potential implications of the use of this option in the advanced threat model (Section 18.2 of [RFC6887]).

9. IANA Considerations

9.1. DHCPv6 Option

IANA is requested to assign the following new DHCPv6 Option Code in the registry maintained in http://www.iana.org/assignments/dhcpv6-parameters:

Option Name	Value
OPTION_V6_PCP_SERVER	TBA

9.2. DHCPv4 Option

IANA is requested to assign the following new DHCPv4 Option Code in the registry maintained in http://www.iana.org/assignments/bootp-dhcp-parameters/:

Option Name	Value	Data length	Meaning
OPTION_V4_PCP_SERVER	TBA	Variable; the minimum length is 5.	Includes one or multiple lists of PCP server IP addresses; each list is treated as a separate PCP server.

10. Acknowledgements

Many thanks to C. Jacquenet, R. Maglione, D. Thaler, T. Mrugalski, T. Reddy, S. Cheshire, M. Wasserman, C. Holmberg, A. Farrel, S. Farrel, B. Haberman, and P. Resnick for their review and comments.

Special thanks to T. Lemon and B. Volz for the review and their effort to enhance this specification.

11. References

11.1. Normative References

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

[RFC2131] Droms, R., "Dynamic Host Configuration Protocol", RFC 2131, March 1997.

[RFC2132] Alexander, S. and R. Droms, "DHCP Options and BOOTP Vendor Extensions", RFC 2132, March 1997.

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

[RFC3396] Lemon, T. and S. Cheshire, "Encoding Long Options in the Dynamic Host Configuration Protocol (DHCPv4)", RFC 3396, November 2002.

[RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, February 2006.

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

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

11.2. Informative References

[I-D.boucadair-pcp-deployment-cases]
Boucadair, M., "PCP Deployment Models", draft-boucadair-pcp-deployment-cases-01 (work in progress), December 2013.

[I-D.ietf-dhc-topo-conf]
Lemon, T. and T. Mrugalski, "Customizing DHCP Configuration on the Basis of Network Topology", draft-ietf-dhc-topo-conf-01 (work in progress), February 2014.

[I-D.ietf-pcp-server-selection]
Boucadair, M., Penno, R., Wing, D., Patil, P., and T. Reddy, "PCP Server Selection", draft-ietf-pcp-server-selection-02 (work in progress), January 2014.

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

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

Authors' Addresses

Mohamed Boucadair
France Telecom
Rennes 35000
France

Email: mohamed.boucadair@orange.com

Reinaldo Penno
Cisco
USA

Email: repenno@cisco.com

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

Email: dwing@cisco.com

pcp
Internet-Draft
Intended status: Standards Track
Expires: December 24, 2011

R. Maglione
Telecom Italia
D. Cheng
Huawei Technologies

June 22, 2011

RADIUS Extensions for Port Control Protocol
draft-maglione-pcp-radius-ext-02

Abstract

This memo proposes a new RADIUS attribute to carry the FQDN of a PCP server, such that while the PCP server information is configured on a RADIUS server, the information can be conveyed to NAS via RADIUS protocol, and the co-located DHCP/DHCPv6 server can then populate the information to PCP client.

Status of this Memo

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

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

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

This Internet-Draft will expire on December 24, 2011.

Table of Contents

1. Introduction

Port Control Protocol (PCP) [I-D.ietf-pcp-base] provides a mechanism to control how incoming packets are forwarded by upstream devices such as NATs and firewalls. PCP is a client-server protocol where a PCP client may reside on a host, a CPE, etc., which communicates with a PCP server that may reside anywhere in a network.

A PCP client must know the Fully Qualified Domain Name (FQDN) of a PCP server, before it can communicate with the later in order to perform the relevant PCP functions.

[I-D.bpw-pcp-dhcp] defines DHCPv6 and DHCP options which are meant to be used by a PCP client to discover a PCP server name. However, provisioning for name of the PCP server is required on a DHCP/DHCPv6 server before it can populate these information.

Auto-configuration on a DHCP/DHCPv6 is possible in a broadband network, where typically, user profile is maintained on a RADIUS server and RADIUS protocol [RFC2865] is used to convey user related information to other network elements including a host and CPE. [I-D.ietf-radext-ipv6-access] describes a typical broadband network scenario in which the Network Access Server (NAS) acts as the access gateway for the users (hosts or CPEs) and the NAS embeds a DHCPv6 Server function that allows it to locally handle any DHCPv6 requests issued by the clients.

In such environment, PCP server’s name can be configured on a RADIUS server, which then passes the information to a NAS that co-locates with the DHCP/DHCPv6 server, which in turn populates the location of the PCP server.

This memo defines a new RADIUS attribute that can be used to carry the FQDN of a PCP server.

The approach described above is already used for providing the FQDN of the AFTR in the DS-Lite scenario and the equivalent RADIUS attribute for the DS-Lite Tunnel Name is defined [I-D.ietf-softwire-dslite-radius-ext].

2. Terminology

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

The following terms are defined in [I-D.ietf-pcp-base]:

- Port forwarding
- PCP
- PCP client
- PCP Server

3. PCP Server Configuration using RADIUS and DHCP/DHCPv6

Figure 1 illustrates how RADIUS protocol works together with DHCPv6, to allow a host to learn automatically the FQDN of a PCP server in case of a PPP session that carries IPv6 traffic.

The Network Access Server (NAS) operates as a client of RADIUS and as DHCPv6 Server for DHCPv6 protocol. The NAS initially sends a RADIUS Access Request message to the RADIUS server, requesting authentication. Once the RADIUS server receives the request, it validates the sending client and if the request is approved, the RADIUS server replies with an Access Accept message including a list of attribute-value pairs that describe the parameters to be used for this session. This list may also contain the name of a PCP server. When the NAS receives a DHCPv6 message containing the PCP Server Option, the NAS shall use the name returned in the RADIUS attribute as defined in this memo to populate the DHCPv6 PCP Server option defined in [I-D.bpw-pcp-dhcp]

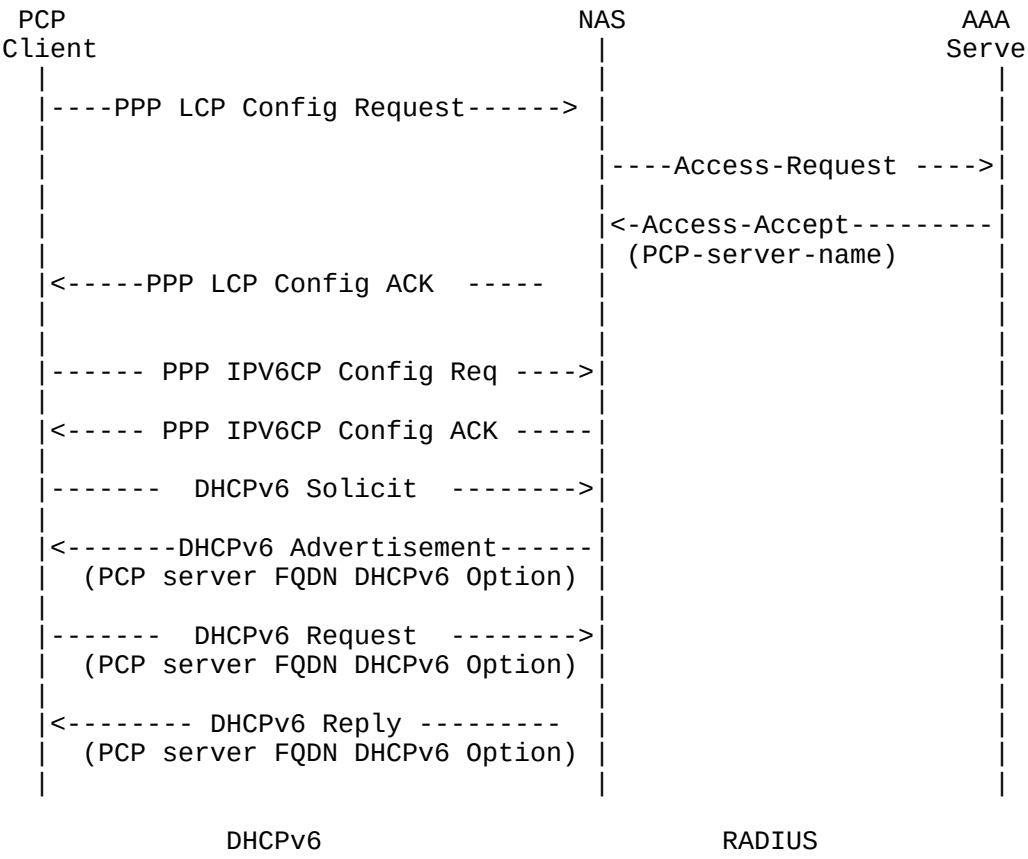


Figure 1: RADIUS and DHCPv6 Message Flow for a PPP Session

The Figure 2 illustrates how the RADIUS protocol and DHCPv6 work together to accomplish PCP client configuration when DHCPv6 is used to provide connectivity to the user.

The only difference between this message flow and previous one is that in this scenario the interaction between NAS and AAA/ RADIUS Server is triggered by the DHCPv6 Solicit message received by the NAS from the B4 acting as DHCPv6 client, while in case of a PPP Session the trigger is the PPP LCP Config Request message received by the NAS.

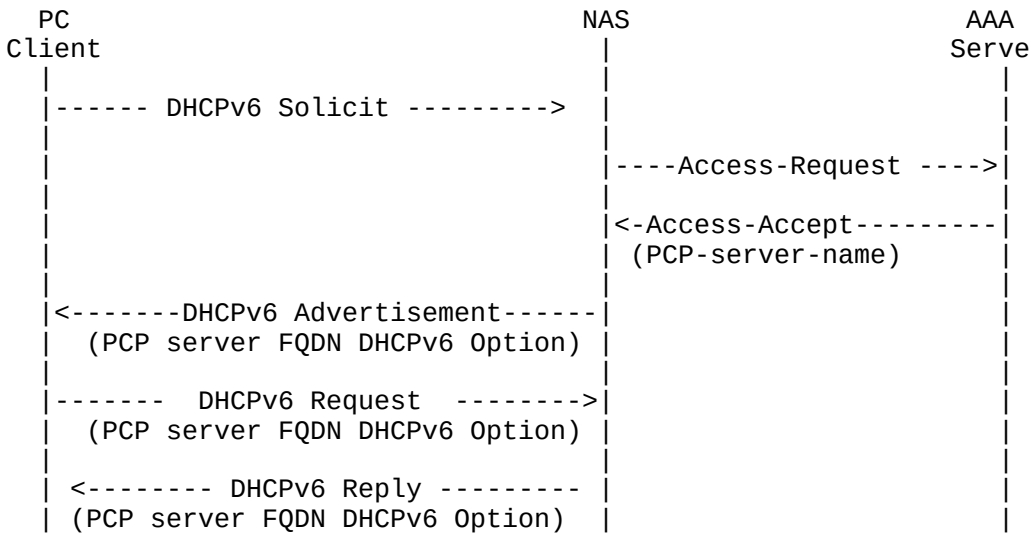


Figure 2: RADIUS and DHCPv6 Message Flow for an IP Session

A similar message flow also applies to the IPv4 scenario when DHCPv4 is used to provide connectivity to the user (Figure 3).

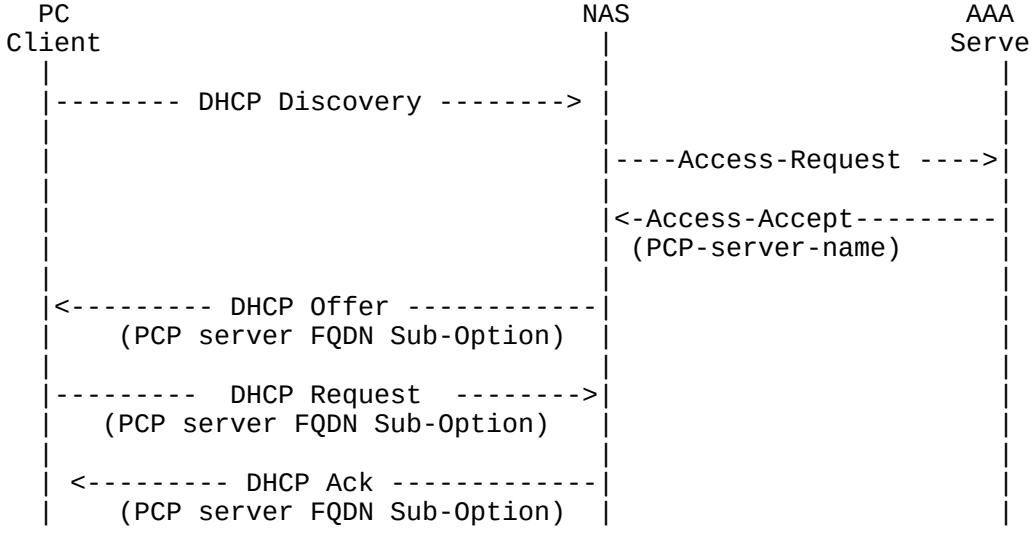


Figure 3: RADIUS and DHCPv4 Message Flow for an IP Session

The scenario with PPP Session and IPv4 only connectivity does not require the DHCP protocol: the whole configuration of the client is performed by PPP. This case is out of scope of this document because in order to complete the configuration of the PCP client a new PPP IPC option would be required.

4. RADIUS Attribute

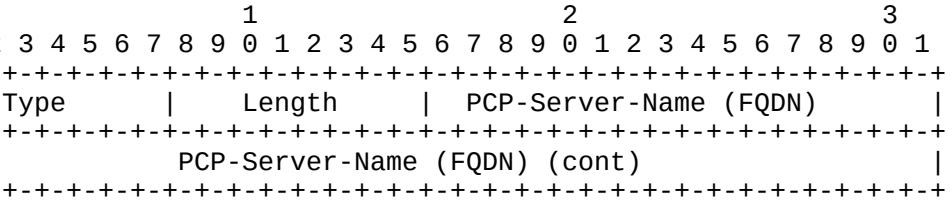
A new RADIUS attribute, called PCP-Server-Name, along with its format is defined below.

Description

The PCP-server-name attribute contains a Fully Qualified Domain Name (FQDN) that refers to a PCP server the client requests to establish a connection to for PCP related service. The NAS shall use the name returned in the RADIUS PCP-server-name attribute to populate the PCP Server FQDN DHCP Sub-Option in IPv4 addressing context, or the PCP Server FQDN DHCPv6 Option in IPv6 addressing context, as determined by the DHCP server [I-D.bpw-pcp-dhcp]

The PCP-server-name attribute MAY appear in an Access-Accept packet, and may also appear in an Accounting-Request packet. In either case, the attribute MUST NOT appear more than once in a single packet. The PCP-server-name MUST NOT appear in any other RADIUS packets.

A summary of the PCP-Server-Name RADIUS attribute format is shown below. The fields are transmitted from left to right.



Type:

TBA1 for PCP-Server-Name.

Length:

This field indicates the total length in octets of this attribute including the Type, the Length fields and the length in octets of the PCP-Server-Name field

PCP-Server-Name:

A single Fully Qualified Domain Name of the PCP-Server. The domain name is encoded as specified in [RFC1035]

5. Table of attributes

The following table provides a guide to which attributes may be found in which kinds of packets, and in what quantity.

Request	Accept	Reject	Challenge	Accounting Request	#	Attribute
0-1	0-1	0	0	0-1	TBA1	PCP-Server-Name

The following table defines the meaning of the above table entries.

0 This attribute MUST NOT be present in packet.
0+ Zero or more instances of this attribute MAY be present in packet.
0-1 Zero or one instance of this attribute MAY be present in packet.

6. Security Considerations

This document has no additional security considerations beyond those already identified in [RFC2865].

7. IANA Considerations

This document requests the allocation of a new Radius attribute types from the IANA registry "Radius Attribute Types" located at http://www.iana.org/assignments/radius-types

PCP-Server-Name - TBA1

8. Acknowledgments

The authors would like to thank Mohamed Boucadair and Mario Ullio for their valuable comments.

9. Normative References

[I-D.bpw-pcp-dhcp]
Boucadair, M., Penno, R., and D. Wing, "DHCP and DHCPv6 Options for the Port Control Protocol (PCP)", draft-bpw-pcp-dhcp-04 (work in progress), April 2011.

[I-D.ietf-pcp-base]
Wing, D., Cheshire, S., Boucadair, M., Penno, R., and P. Selkirk, "Port Control Protocol (PCP)", draft-ietf-pcp-base-12 (work in progress), May 2011.

[I-D.ietf-radext-ipv6-access]
Lourdelet, B., Dec, W., Sarikaya, B., Zorn, G., and D. Miles, "RADIUS attributes for IPv6 Access Networks", draft-ietf-radext-ipv6-access-04 (work in progress), March 2011.

[I-D.ietf-softwire-dslite-radius-ext]
Maglione, R. and A. Durand, "RADIUS Extensions for Dual-Stack Lite", draft-ietf-softwire-dslite-radius-ext-02 (work in progress), March 2011.

[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.

[RFC2865] Rigney, C., Willens, S., Rubens, A., and W. Simpson, "Remote Authentication Dial In User Service (RADIUS)", RFC 2865, June 2000.

Authors' Addresses

Roberta Maglione
Telecom Italia
Via Reiss Romoli 274
Torino 10148
Italy

Phone:
Email: roberta.maglione@telecomitalia.it

Dean Cheng
Huawei Technologies
2330 Central Expressway
Santa Clara, CA 95050
USA

Phone: +1 408 330 4754
Fax:
Email: Chengd@huawei.com
URI:

Port Control Protocol
Internet-Draft
Intended status: Standards Track
Expires: April 23, 2012

R. Penno
Juniper Networks
D. Wing
Cisco
P. Selkirk
Internet Systems Consortium
M. Boucadair
France Telecom

October 21, 2011

PCP Support for Nested NAT Environments
draft-penno-pcp-nested-nat-01

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 current 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 level 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."

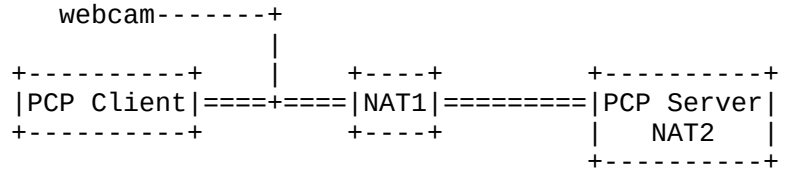
This Internet-Draft will expire on April 23, 2012.

Table of Contents

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:

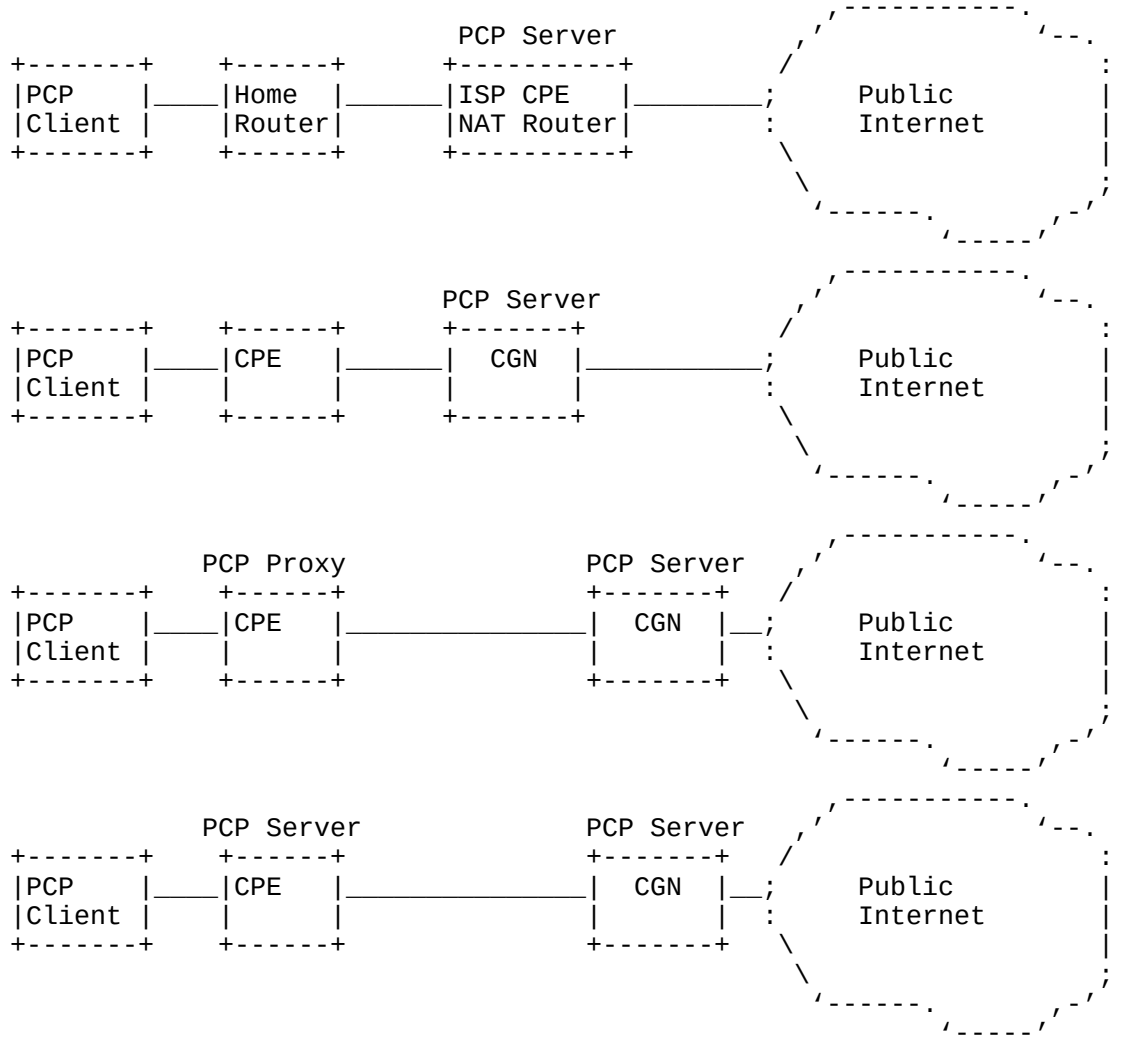
- NAT1 is detected in the path between the PCP Client and the PCP Server owing to the use of the RCEIVED_PORT Option and returned perceived IP address in PCP response;
- After learning about that NAT, the PCP Client uses UPnP IGD, NAT-PMP or manual configuration to interact with NAT1 and open the necessary port on NAT1 (e.g., IP address= IPx, port=X);
- 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 s 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:

- PCP and UPnP unaware or disabled
- PCP Server
- UPnP Server
- PCP Proxy

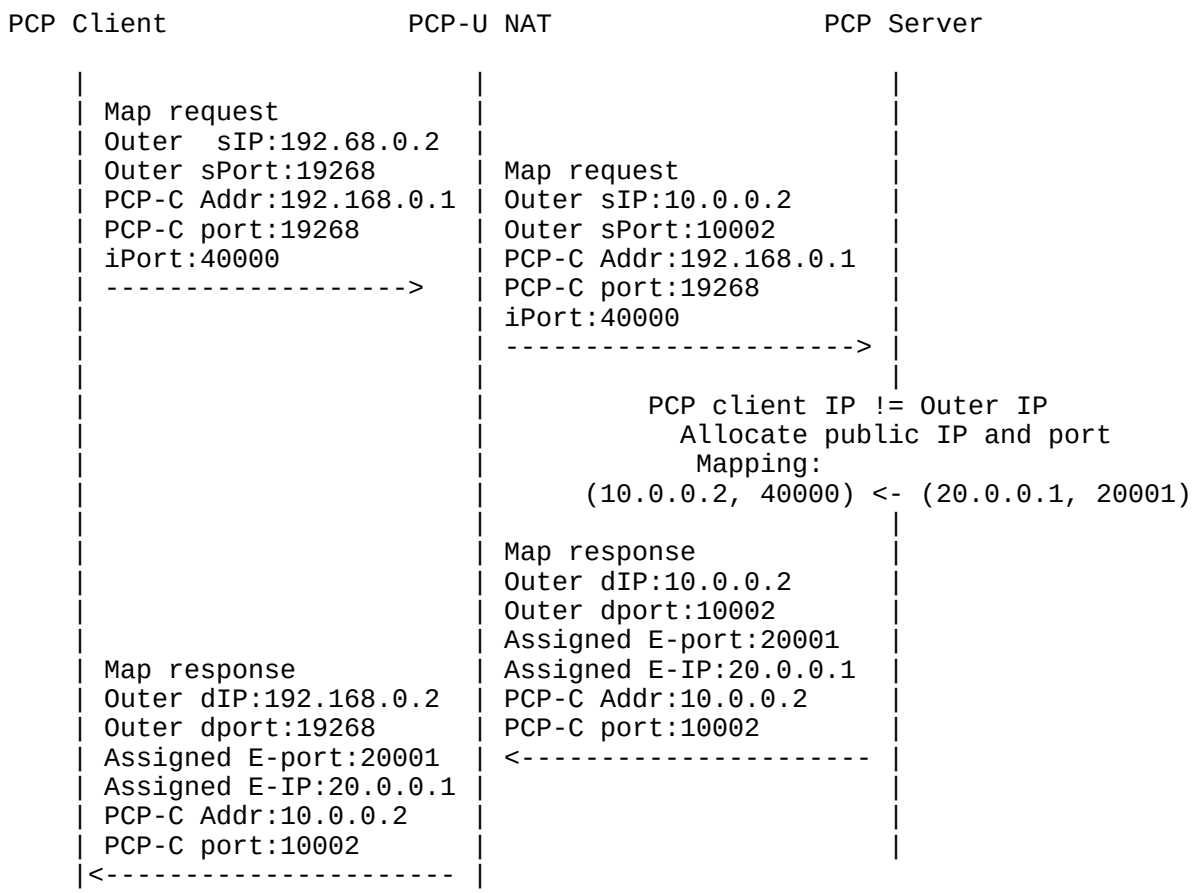
The next sections discuss each scenario on the basis of protocol support on intermediate NATs.

2.1. NAT and UPnP unaware Intermediate NATs

This method will most likely be used by PCP clients in nested NAT environments while PCP Proxy support in 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
```

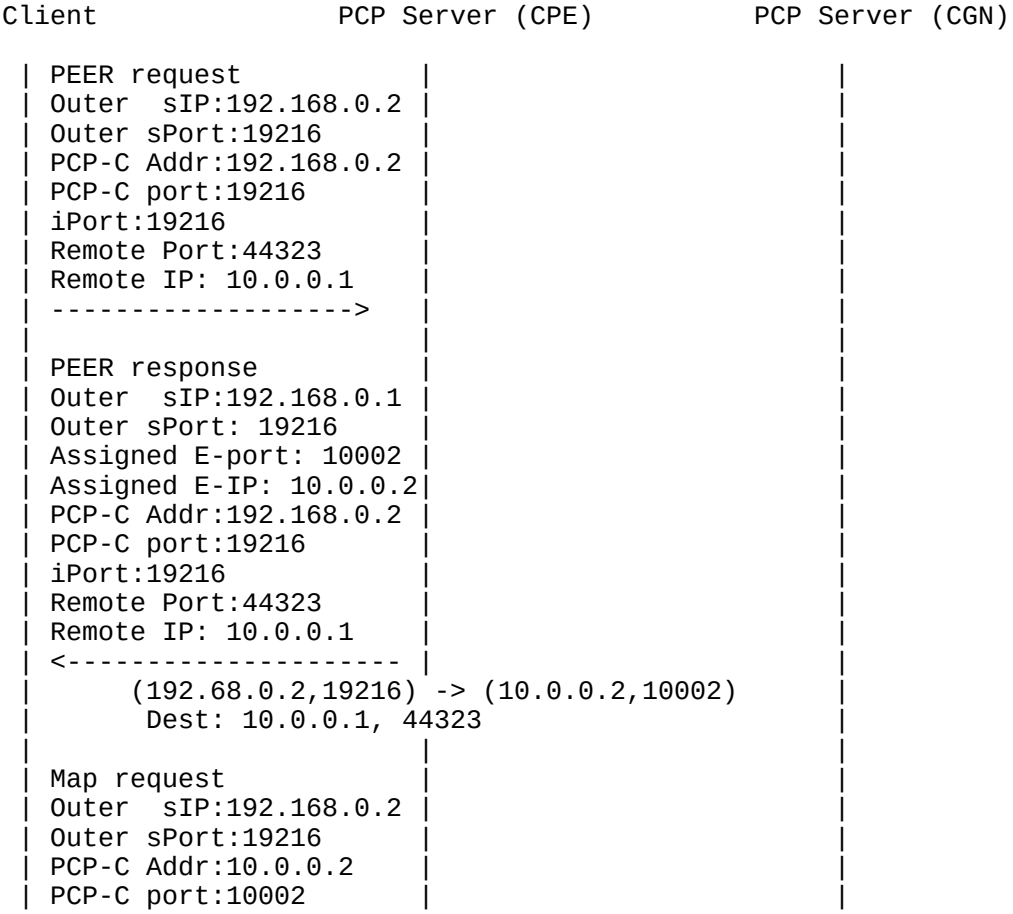


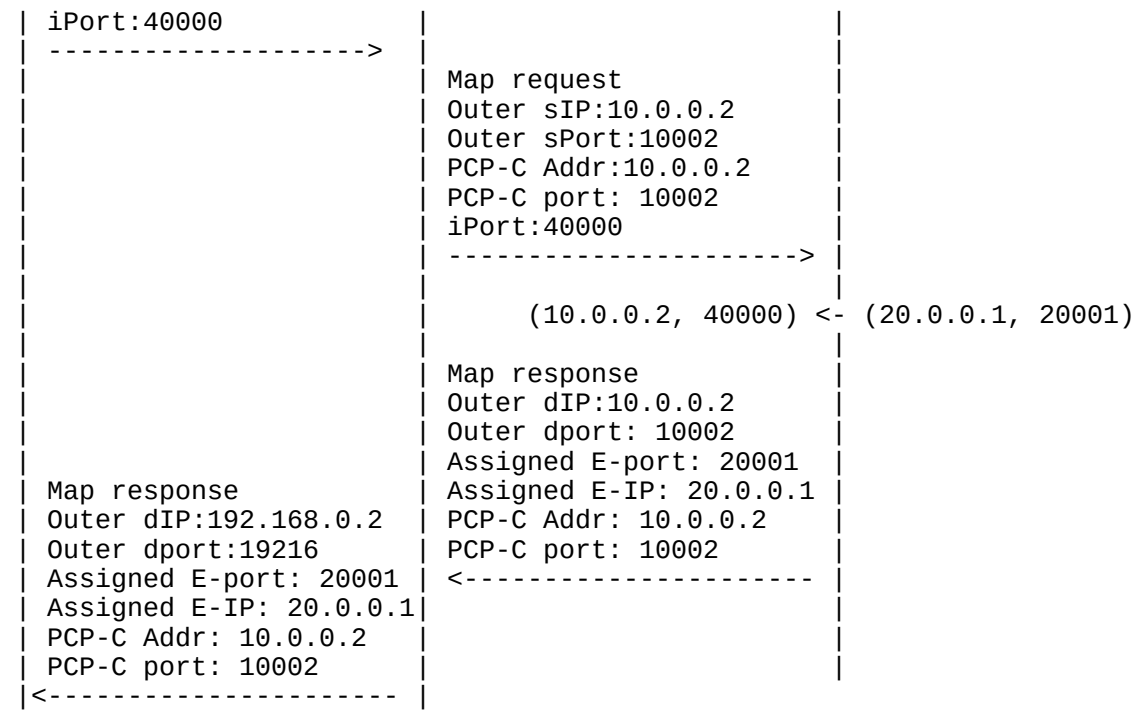
- Subscriber installs a port forwarding or DMZ entry on its home CPE (PCP U-NAT) through manual configuration. The entry would be (*, 40000) -> (10.0.0.1, 40000). Alternatively the application could use UPnP for the same purpose.

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 needs 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.





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 its 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

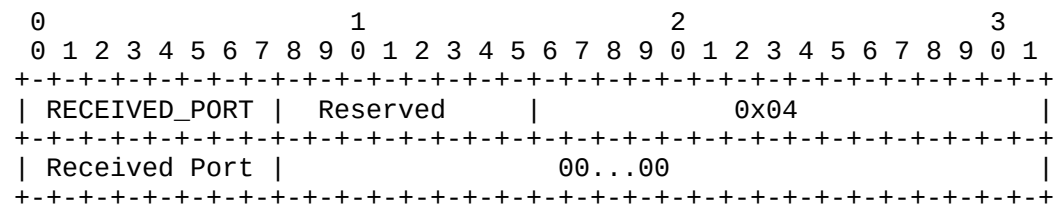
TBD

3. RECEIVED_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 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

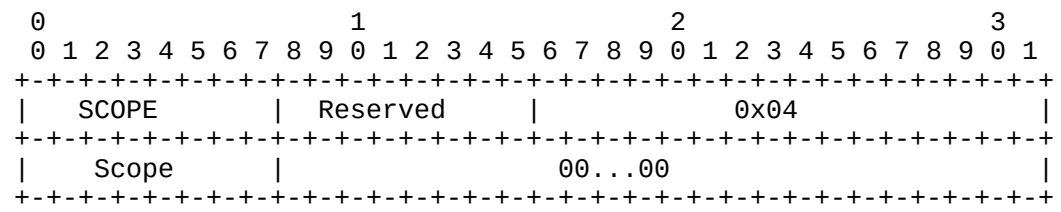


Figure 2: Scope Option

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-16 (work in progress), October 2011.

[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
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

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

Port Control Protocol
Internet-Draft
Intended status: Standards Track
Expires: April 22, 2012

R. Penno
Juniper Networks

October 20, 2011

PCP Support for Multi-Zone Environments
draft-penno-pcp-zones-01

Abstract

A zone is a notion which denotes a routing instance, a set interfaces or prefixes characterized by having a different address realm and/or security policy. A NAT device can route packets with the same source IP address to different zones depending on configuration policies such as destination IP address. This functionality has been present for many years in NAT devices from multiple vendors. PCP allows a host to interact with a PCP-controlled NAT device and request an external IP and port. Therefore a PCP Server that controls the NAT device and receives a PCP request from a host needs to know from which NAT pool to allocate an external IP address and port. This document specifies an extension to PCP to support the zone concept.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

Status of this Memo

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

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

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

This Internet-Draft will expire on April 22, 2012.

Copyright Notice

Table of Contents

1. Introduction

A zone is a routing instance, set interfaces or prefixes characterized by having a different address domain or security policy. A NAT device is present on each zone through NAT pools which are used to translate packet to and from a zone. The PCP protocol allows a host to interact with a NAT device and request a external IP and port. Since a NAT Device can route packets with the same source IP address to different Zones depending on policy or packet match conditions, the PCP Server that interacts with the NAT device and receives a PCP request from a host needs to know from which NAT pool to allocate an IP address and port.

1.1. Terminology

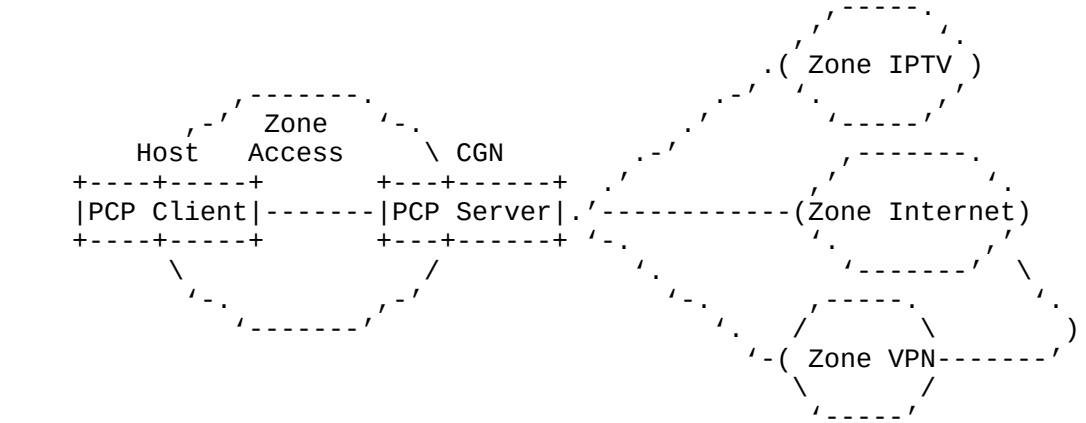
This document uses PCP terminology defined in [I-D.ietf-pcp-base]]. In addition the following terms are defined in this document:

- Zone: A routing instance, set of interfaces or network prefixes that has a separate addressing domain or security policy.

- Address Domain: A collection of IP addresses. A NAT device is present on each domain through one or more NAT pools associated with each Zone.

1.2. Problem Statement

A PCP Server can control a NAT attached to distinct zones; each zone is characterised by one or several address pools. In such environment the NAT must rely on a pre-configured policy to determine which address pool to use when handling an IP packet coming from an internal host. An example of such policy may be to rely on the destination IP address, DSCP value(s), protocol (e.g., SIP, RTP, RTSP), etc.



The core of the problem is that packets from the same source IP address can be routed to any of the zones depending on match conditions based on the 5-tuple. Moreover, sessions could be initiated from any of these zones toward the host. These zones many times have different addressing domains and therefore different NAT pools. This means that packets from the host will use a different NAT pool depending on the destination zone.

It is important to notice that zones (or similar concept) has been present in Enteprise NAT and CGN from multiple vendors for many years. It is the advent and interaction with PCP that has created a need for a standardized approach.

1.3. Scope

The matching conditions that ultimately decide where to route a packet can be very elaborate including even application layer information. But the scope of this document is to abstract such implementation specific approaches behind the concept of a Zone-ID.

2. PCP Base Support for Multiple Zones

Before discussing extensions to the PCP protocol in the following sections we discuss how to support multiple zones with the current methods present in the base PCP protocol.

2.1. PCP PEER Request

A PCP PEER request could contains the destination IP address, port and Transport protocol of the peer the host will be trying to communicate . In that case, if the NAT device maintains a mapping of

zones (and associated NAT pools) to network prefixes it can choose the appropriate NAT pool. It is important to understand that this will only work if the policy that decides to which Zone to route packets is only based on the information present on the PCP PEER request.

Therefore if the PCP Client knows it is behind a NAT with zone support, it is RECOMMENDED that it includes the remote peer's 5-tuple in the PCP PEER request in the connect-then-lifetime case. If the peer's 5-tuple is not present in the PCP request, the external IP and port returned in the message is non-deterministic.

2.2. PCP MAP Request

In the case of PCP MAP request the NAT device does not know from which zone to install a mapping and consequently from which NAT pool to choose an external IP address and port. A FILTER Option may be included to allow the PCP Server select the external address pool to use. If other information than the destination IP address is used to drive the selection of the external address pool, additional information is required to be conveyed in the PCP MAP request (e.g., DSCP marking policy (see http://tools.ietf.org/html/draft-boucadair-pcp-extensions-01#section-3).

3. PCP Extension for Multiple Zones

The proposed PCP extension is a new PCP Option that would convey the Zone-ID. The Zone-ID is an opaque identifier that is known by the PCP Client and the PCP-controlled NAT device. The procedure to provision the Zone-ID is out of scope.

When the NAT device receives a PCP request with a Zone-ID, it will use that or a derivative of it to determine the NAT pool from which to allocate an IP address and port.

Option Name: ZONEID

Number: TBA (IANA); Mandatory to process

Purpose: It allows the client request and server indicate from which Zone-ID the external IP:port were allocated.

Valid for Opcodes: MAP, PEER

Length: Variable

May appear in: both

Maximum occurrences: 1

4. IANA Considerations

TBD

5. Security Considerations

Subscribers can only request ports for the specific Zone-IDs allowed in their security profile. For example, in a typical Wireless deployment, mobile terminals could request mappings in zones 'Internet', 'HTTP Proxy Farm', and 'Video Farm'. A PCP request that contains a zone-id considered a security violation would be silently dropped.

6. Acknowledgements

Thanks to Mohamed Boucadair for early review comments

7. References

7.1. Normative References

[RFC0959] Postel, J. and J. Reynolds, "File Transfer Protocol", STD 9, RFC 959, October 1985.

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

[RFC2766] Tsirtsis, G. and P. Srisuresh, "Network Address Translation - Protocol Translation (NAT-PT)", RFC 2766, February 2000.

[RFC2960] Stewart, R., Xie, Q., Morneault, K., Sharp, C., Schwarzbauer, H., Taylor, T., Rytina, I., Kalla, M., Zhang, L., and V. Paxson, "Stream Control Transmission Protocol", RFC 2960, October 2000.

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

[RFC4966] Aoun, C. and E. Davies, "Reasons to Move the Network Address Translator - Protocol Translator (NAT-PT) to Historic Status", RFC 4966, 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.

[RFC5508] Srisuresh, P., Ford, B., Sivakumar, S., and S. Guha, "NAT Behavioral Requirements for ICMP", BCP 148, RFC 5508, April 2009.

7.2. Informative References

[I-D.ietf-behave-address-format]
Bao, C., Huitema, C., Bagnulo, M., Boucadair, M., and X. Li, "IPv6 Addressing of IPv4/IPv6 Translators", draft-ietf-behave-address-format-10 (work in progress), August 2010.

[I-D.ietf-behave-dns64]
Bagnulo, M., Sullivan, A., Matthews, P., and I. Beijnum, "DNS64: DNS extensions for Network Address Translation from IPv6 Clients to IPv4 Servers", draft-ietf-behave-dns64-11 (work in progress), October 2010.

[I-D.ietf-behave-ftp64]
Beijnum, I., "An FTP ALG for IPv6-to-IPv4 translation", draft-ietf-behave-ftp64-12 (work in progress), July 2011.

[I-D.ietf-behave-v6v4-framework]
Baker, F., Li, X., Bao, C., and K. Yin, "Framework for IPv4/IPv6 Translation", draft-ietf-behave-v6v4-framework-10 (work in progress), August 2010.

[I-D.ietf-behave-v6v4-xlate-stateful]
Bagnulo, M., Matthews, P., and I. Beijnum, "Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers", draft-ietf-behave-v6v4-xlate-stateful-12 (work in progress), July 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-16 (work in progress), October 2011.

[I-D.wing-behave-dns64-config]
Wing, D., "IPv6-only and Dual Stack Hosts on the Same Network with DNS64", draft-wing-behave-dns64-config-03 (work in progress), February 2011.

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

[RFC5245] Rosenberg, J., "Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols", RFC 5245, April 2010.

[RFC5853] Hautakorpi, J., Camarillo, G., Penfield, R., Hawrylyshen, A., and M. Bhatia, "Requirements from Session Initiation Protocol (SIP) Session Border Control (SBC) Deployments", RFC 5853, April 2010.

Author's Address

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

Email: rpenno@juniper.net

Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: January 9, 2012

C. Zhou
Huawei Technologies
T. Tsou
Huawei Technologies (USA)
X. Deng
M. Boucadair
France Telecom
Q. Sun
China Telecom

July 8, 2011

Using PCP To Coordinate Between the CGN and Home Gateway Via Port Allocation
draft-tsou-pcp-natcoord-03

Abstract

Consider a situation where a subscriber’s packets are subject to two levels of NAT, with both NATs operating under the control of the ISP. An example of this would be a NATing Home Gateway forwarding packets to a Large Scale NAT. This memo proposes that advantage be taken of the presence of the second NAT, to offload the burden on the Large Scale NAT by delegation to the Home Gateway. Enhancements to the Port Control Protocol are specified to achieve this. The proposed solution applies also for DS-Lite where the AFTR offloads it NAT to the B4 element.

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 9, 2012.

Copyright Notice

Table of Contents

1. Application Scenario

A Large Scale NAT (LSN) is responsible for translating source addresses and ports for packets passing into and out of the provider network. Especially for large scale service providers, one LSN may need to support at least tens of thousands of customers, resulting in heavy processing requirements for the LSN.

In some broadband scenarios an additional NAT is present at the edge of the customer network. For convenience we will call this the Home Gateway. The load on the LSN could be reduced if address and port translation were actually done at the Home Gateway. Achieving such an outcome would require coordination between the two devices. This memo makes a detailed proposal for the required coordination mechanism.

2. Proposed Solution

2.1. Delegation of Port Sets

The basic proposal made in this memo is to provide the means for the Home Gateway to request that the LSN delegate to it a set of ports and optionally an external address that will be associated with those ports. It is proposed to use the Port Control Protocol (PCP) [ID.port-control-protocol] to achieve this. The procedure is illustrated in Figure 1.

> The LSN allocation of port sets MAY take into account the advice given in [ID.behave-natx4-log-reduction].
>
> [Open Issue: if we want to make the port sets discontinuous, we must either allow negotiation of the algorithm or parameters of that algorithm for determining the complete set from a given starting point, or specify it here. Specifying it all here is probably counter-productive, given that this is a security measure to make port guessing harder.]

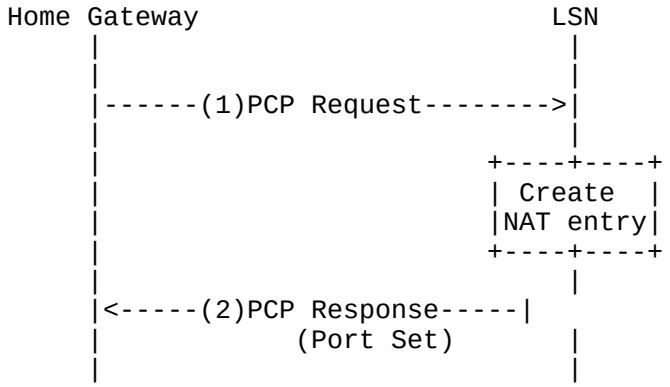


Figure 1: Acquiring a Delegated Port Set

If the Home Gateway allocates all of the ports that have been delegated to it for a given protocol, it MAY send a request to the LSN for another delegated set of ports. If the LSN satisfies that request, the Home Gateway MUST release the additional set as soon as possible. To achieve this, the Home Gateway May follow a policy for allocation of additional ports to flows, that has the same effect as searching for "free" ports in the port sets in the order in which they were delegated to the Home Gateway. A port SHOULD be considered "free" if no traffic has been observed through it for the timeout interval specified for the protocol concerned, as discussed in [ID.behave-natx4-log-reduction], or if the Home Gateway knows through other means (e.g., host reboot) that it is no longer in use.

2.2. Packet Processing At the Home Gateway and LSN

The Home Gateway maps outgoing flows to the delegated ports. If an external address was received it uses that for the source address; otherwise it retains the private address of the Home Gateway as the source address.

> The procedures are more complicated, of course, if the IP version running externally to the LSN is different from the IP version running between the Home Gateway and the LSN, since the destination address also has to be translated. The details depend on the particular transition mechanism in use, and are left as an exercise for the reader.

If the private address is retained, the LSN recognizes it from the original delegation request and changes the source address but not the port before forwarding the packet. If the external public address was used, the LSN is not useful and another device may be needed to allocate the port set.

In the reverse direction, the LSN recognizes the public destination address and port of an incoming packet as belonging to a delegated set for the Home Gateway. It translates the destination address, if necessary, leaving the destination port unchanged. The Home Gateway translates the destination port and address to the corresponding values in the customer network and forwards the packet in turn.

2.3. Proposed Enhancements To and Usage Of the Port Control Protocol

This document proposes the following new option for MAP opcodes: PORT_SET_REQUESTED.

option number: to be allocated

is valid for OpCodes: MAP44, MAP64, MAP46, or MAP66

is included in responses: MUST

has length: 0 in requests, 4 in successful responses. [As mentioned above, if non-consecutive sets of ports are allocated, we may want to add parameters of the algorithm for deriving the complete set from the initial value provided in the "assigned external port" field of the response.]

may appear more than once: no

When constructing a PCP request with the PORT_SET_REQUESTED option, the client MUST set the "internal port" field of the request to zero. If requesting a new set of delegated ports, the client MAY set the "requested external port" field to a non-zero value. If releasing a set of delegated ports (i.e., by setting the "Requested lifetime" field to zero), the client MUST set the "requested external port" field to the value of the "assigned external port" field of the earlier response from the server. The remaining fields of the PCP request MUST be set as directed by [ID.port-control-protocol]

[Open issue: for a release, should the PORT_SET_REQUESTED option have the same contents as it had in the earlier response?]

Upon receiving a PCP request with the PORT_SET_REQUESTED option, the server MAY reject it using return codes 151 - NOT_AUTHORIZED, or 152 - USER_EX_QUOTA. In this case, the PORT_SET_REQUESTED option in the response MUST have zero length (no data). If the server chooses to honour the request, it MUST place the value of the first port in the assigned set in the "assigned external port" field of the response. It MUST set the length of the PORT_SET_REQUESTED option in the response to 4, and MUST provide the number of ports in the delegated set as the value of the option.

3. Port Range Option

The Port_Range option is used to specify one set of ports (contiguous or not contiguous) pertaining to a given IP address. The starting point of the ports and the number of delegated ports are used to infer a set of allowed port values.

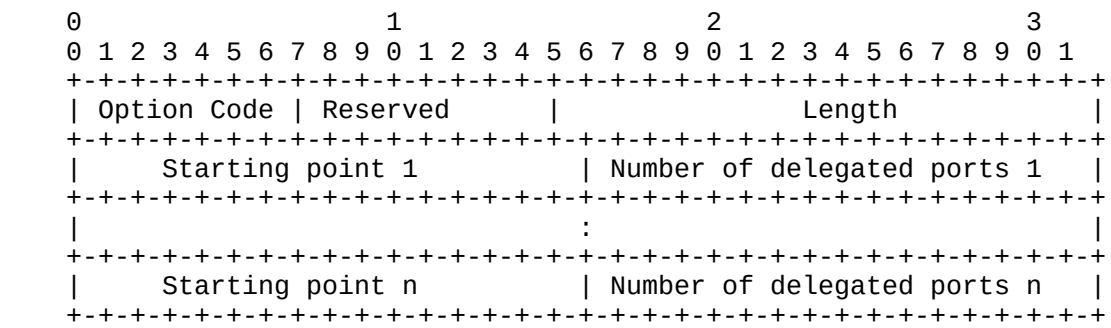


Figure 2: Port_Range_Option

This option:

- name: Port range option
- number: TBA
- purpose: A PCP Client inserts this option in a PCP request to specify one set of ports (contiguous or not contiguous) pertaining to a given IP address.
- is valid for OpCodes:all.
- length:The length MUST be set to 0.
- may appear in:request
- maximum occurrences:none

4. Security Considerations

Will do later. Trust issues between the client and server, plus the port randomization issues discussed in [ID.behave-natx4-log-reduction] and [ID.zhou-softwire-b4-nat].

5. Additional Author

Xiaohong Deng <xiaohong.deng@orange-ftgroup.com> joined the list of authors for version -03 of this draft.

6. IANA Considerations

Will register the new option if this draft goes through as a standalone document rather than being incorporated into the base protocol.

7. Additional Author

Gabor Bajko

Nokia

Email: gabor.bajko@nokia.com

8. References

8.1. Normative References

[ID.port-control-protocol]
Wing, D., "Port Control Protocol (PCP)", January 2011.

8.2. informative References

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

[ID.zhou-softwire-b4-nat]
Deng, X., Zhou, C., Boucadair, M., Bajko, G., and T. Tsou, "DS-Lite AFTR NAT Bypass: Co-located B4 and NAT Model", June 2011.

Appendix A. NAT By-pass PCP

A.1. Introduction

This section defines a new PCP option denoted NAT by-pass option. The purpose of this option is to instruct a PCP- controlled device to not envoke NAT operation on a set of flows destined to a given device

located behind the PCP-controlled device.

A.1.1. Use Cases

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.

A.1.2. Scope

As currently defined in PCP Base document, PCP is unable to instruct a PCP-controlled device to de-activate the NAT for a given customer, given flows, etc.

This document defines new PCP options which are meant to instruct a PCP-controlled device to by-pass the NAT function whenever required.

A.2. NAT Bypass PCP Option

This option (Figure 3) is used by a PCP Client to indicate to the PCP Server to not apply any NAT operation to a corresponding binding.

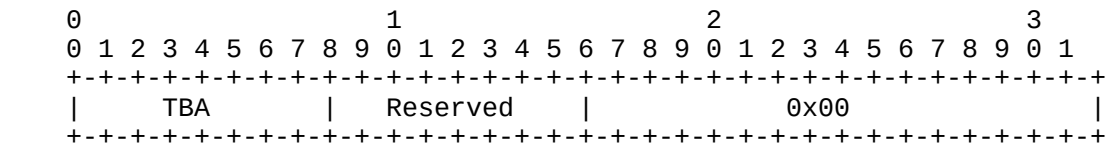


Figure 3: NAT Bypass option

This option:

- name: NAT Bypass option
- number: TBA
- purpose: A PCP Client inserts this option in a PCP request to indicate to the PCP Server to not apply the NAT function. The NAT is then by-passed in the PCP-controlled device.
- is valid for OpCodes:all.
- length:The length MUST be set to 0.
- may appear in:request
- maximum occurrences:none

A PCP Client inserts this option in a PCP request to indicate to the PCP Server to not apply the NAT function. The NAT is then by-passed in the PCP-controlled device.

A PCP Server which supports the NAT by-pass feature MUST include this option in its response to the requesting PCP Client. In particular, when the PCP Server does not include this option in its response, the PCP Client should deduce that the NAT will be enforced in the PCP-controlled device; a NAT will be then enforced in the PCP-controlled device.

The NAT bypass feature can be associated with a plain IP address. In such case, a full external IP address is returned to the requesting PCP Client. The client is then able to use all ports associated with that IP address (i.e., without any restriction). Furthermore, this "full" address can be used to access services which do not rely on protocols embedding a port number (e.g., some IPsec modes).

In some cases, the PCP Client can request the by-pass of the NAT but without requiring a full IP address (e.g., for the use cases described in bullet 2 and 3 of Appendix A.1.1). In such scenario, in

addition to the NAT by-pass option, the PCP Client includes in its PCP request a Port Set Option (Appendix A.3). More information about this option is provided hereafter.

The requested lifetime in the PCP MAP request is set to the available lifetime of the port set. If the lifetime is set to zero, it means that the requested port set should be deleted. Internal port, external port and the external address are all invalid.

A.3. Port Set Option

This option (Figure 4) is used to indicate a request for a contiguous port set. This option conveys the length of the requested ports set. It is up to the PCP Server to decide whether the request will be satisfied or not. In particular, the PCP Server may discard the request or accept to assign a port range with a length distinct than the one requested by the PCP Client. The PCP Server can assign a bigger or shorter ports set compared to is actually requested by a PCP Client.

If the PCP Server supports the ability to delegate a set of ports to a requesting PCP Client, it should include in its PCP response the external port set option described in Figure 5.

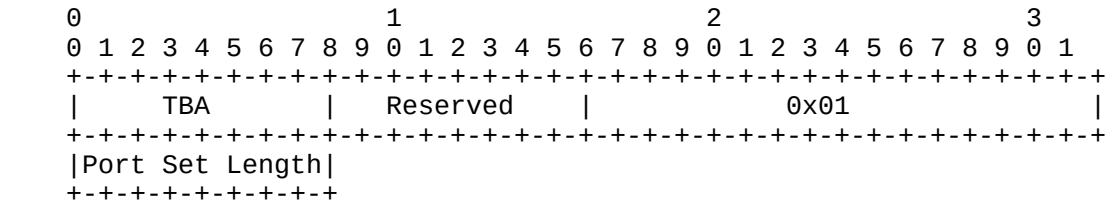


Figure 4: Port Set Option

This option:

- name: Port Set Length option
- number: TBA
- purpose: This option is used to indicate a request for a contiguous port set. This option indicates the length of the requested ports set.
- is valid for OpCodes:all.

- o length:The length MUST be set to 1.
- o may appear in:request
- o maximum occurrences:none

If the PCP Server is configured to assign port ranges, it should use the External Port Set option (Appendix A.4) in its response to convey a range of port to a requesting PCP Client.

A.4. External Port Set

This option is used to enclose contiguous ports set in a PCP message sent by the PCP Server to a requesting Client. This option may be included in a PCP response to delegate a set of ports associated with the same external IP address.

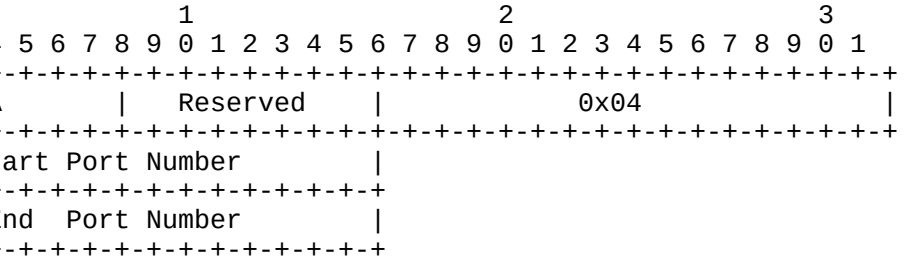


Figure 5: External Ports Set

This option:

- o name: External Ports Set option
- o number: TBA
- o purpose: This option is used to enclose contiguous ports set in a PCP message sent by the PCP Server to a requesting Client.
- o is valid for OpCodes:all.
- o length:The length MUST be set to 4.
- o may appear in:response
- o maximum occurrences:none

The data part of this option indicate the bounds of the assigned

ports range.

A PCP Client which receives this option from a PCP Server is delegated all the port numbers within that range.

A.5. External Non-Contiguous Port Set

This option is used to enclose non-contiguous ports set in a PCP message sent by the PCP Server to a requesting Client. This option may be included in a PCP response to delegate non-overlapping sets of non-contiguous ports associated with the same external IP address to different PCP Client.

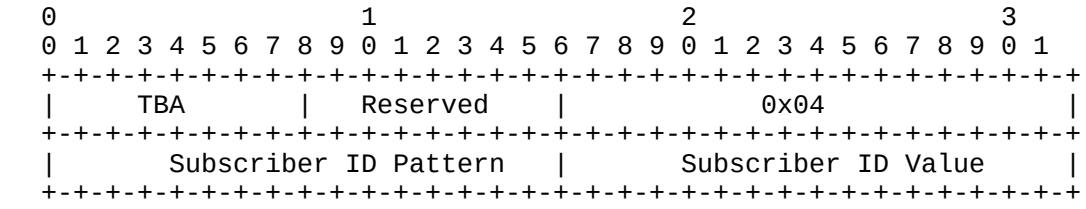


Figure 6: External Non-Contiguous Ports Set

This option:

- name: External Non-Contiguous Ports Set
- number: TBA
- purpose: This option is used to enclose non-contiguous ports set in a PCP message sent by the PCP Server to a requesting Client.
- is valid for OpCodes:all.
- length:The length MUST be set to 4.
- may appear in:response
- maximum occurrences:none

As described in [ID.ietf-intarea-shared-addressing-issues] , a bulk of incoming ports can be reserved as a centralized resource shared by all subscribers using a given restricted IPv4 address. In order to distribute incoming ports as scattered as possible among subscribers sharing the same restricted IPv4 address, other than allocating a continuous range of ports to per subscriber, a solution to distribute

bulks of non-continuous ports among subscribers, which also takes port randomization of CPE NAT into account, because port randomization is one protection among others against blind attacks, is elaborated thereby.

On every restricted IPv4 address, according to port set size N, log2(N)bits are randomly chose as subscribers identification bits(s bit) among 1st and 16th bits. Take a sharing ration 1:32 for example, Figure 4 shows an example of 5bits (2nd, 5th, 7th, 9th, 11th) being chose as s bit.

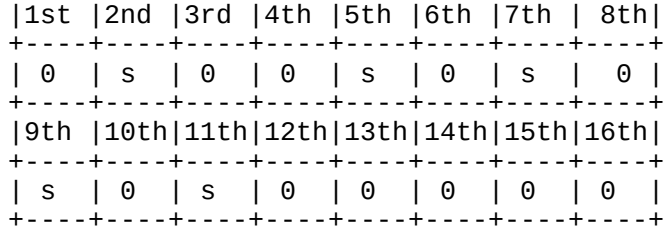


Figure 7: An s bit selection example (on a sharing ration 1:32 address).

Subscriber ID pattern is then formed by setting all the s bits to 1 and other trivial bits to 0. Figure 5 illustrates an example of subscriber ID pattern which follows the s bit selection of figure 4. Note that the subscriber ID pattern can be different, ensured by the random s bit selection, per restricted IP address no matter whether the sharing ratio varies.

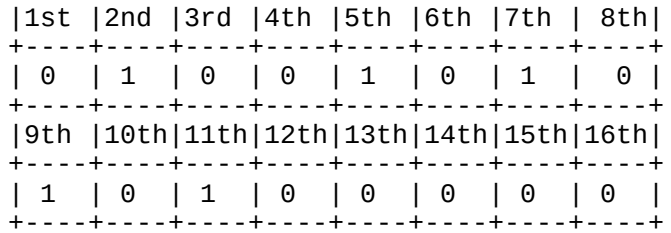


Figure 8: A subscriber ID pattern example (on a sharing ration 1:32 address).

Subscribers ID value is then assigned by setting subscriber ID pattern bits (s bits shown in figure 4) to a unique customer value and setting other trivial bits to 1. An example of subscriber ID value, having a subscriber ID pattern shown in the figure 5 and a

customer value 0, is shown in the figure 6.

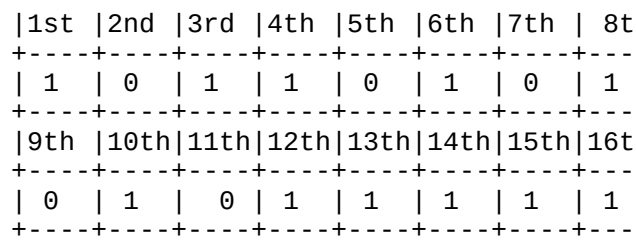


Figure 9: A subscriber ID value example (customer value: 0).

Subscriber ID pattern and subscriber ID value together uniquely defines a restricted port set (Non-contiguous port sets or a contiguous port range, depends on Subscriber ID pattern and subscriber ID value) on a restricted IP address.

Pseudo-code shown in the figure 7 describes how to use subscriber ID pattern and subscriber ID value to implement a random ephemeral port selection function within the defined restricted port sets on a customer NAT.

```
do{
    restricted_next_ephemeral = (random()|subscriber_ID_pattern)
                                & subscriber_ID_value;
    if(five-tuple is unique)
    return restricted_next_ephemeral;
}
```

Figure 10: Random ephemeral port selection within the restricted port set.

Authors' Addresses

Cathy Zhou
Huawei Technologies
Bantian, Longgang District
Shenzhen 518129
P.R. China

Phone:
Email: cathyzhou@huawei.com

Tina Tsou
Huawei Technologies (USA)
2330 Central Expressway
Santa Clara, CA 95050
USA

Phone: +1 408 330 4424
Email: tena@huawei.com

Xiaohong Deng
France Telecom

Email: xiaohong.deng@orange-ftgroup.com

Mohamed Boucadair
France Telecom
Rennes, 35000
France

Email: mohamed.boucadair@orange-ftgroup.com

Qiong Sun
China Telecom
P.R.China

Phone: 86 10 58552936
Email: sunqiong@ctbri.com.cn

Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: August 26, 2013

Q. Sun
China Telecom
M. Boucadair
France Telecom
S. Sivakumar
Cisco Systems
C. Zhou
Huawei Technologies
T. Tsou
Huawei Technologies (USA)
S. Perreault
Viagenie
February 22, 2013

Port Control Protocol (PCP) Extension for Port Set Allocation
draft-tsou-pcp-natcoord-10

Abstract

This document defines an extension to PCP allowing clients to manipulate sets of ports as a whole. This is accomplished by a new MAP option: PORT_SET.

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 August 26, 2013.

Copyright Notice

Table of Contents

1. Introduction

This section describes a few (and non-exhaustive) envisioned use cases. Note that the PCP extension defined in this document is generic and is expected to be applicable to other use cases.

1.1. Lightweight 4over6

In the Lightweight 4over6 [I-D.cui-softwire-b4-translated-ds-lite] architecture, shared global addresses can be allocated to customers. It allows moving the Network Address Translation (NAT) function, otherwise accomplished by a Carrier-Grade NAT (CGN) [I-D.ietf-behave-lsn-requirements], to the Customer-Premises Equipment (CPE). This provides more control over the NAT function to the user, and more scalability to the ISP.

In the lw4o6 architecture, the PCP-controlled device corresponds to the lwAFTR, and the PCP client corresponds to the lwB4. The client

sends a PCP MAP request containing a PORT_SET option to trigger shared address allocation on the lwAFTR. The PCP response contains the shared address information, including the port set allocated to the lwB4.

1.2. Applications Using Port Sets

Some applications require not just one port, but a port set. One example is a Session Initiation Protocol (SIP) User Agent Server (UAS) [RFC3261] expecting to handle multiple concurrent calls, including media termination. When it receives a call, it needs to signal media port numbers to its peer. Generating individual PCP MAP requests for each of the media ports during call setup would introduce unwanted latency. Instead, the server can pre-allocate a set of ports such that no PCP exchange is needed during call setup.

Using PORT_SET, an application can manipulate port sets much more efficiently than with individual MAP requests.

1.3. Firewall Control

Port sets are often used in firewall rules. For example, defining a range for RTP [RFC3550] traffic is common practice. The MAP request can already be used for firewall control. The PORT_SET option brings the additional ability to manipulate firewall rules operating on port sets instead of single ports.

2. Terminology

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

3. The need for PORT_SET

Multiple MAP requests can be used to manipulate a set of ports, having roughly the same effect as a single use of a MAP request with a PORT_SET option. However, use of the PORT_SET option is more efficient when considering the following aspects:

Network Traffic: A single request uses less network resources than multiple requests.

Latency: Even though MAP requests can be sent in parallel, we can expect the total processing time to be longer for multiple requests than a single one.

Client-side simplicity: The logic that is necessary for maintaining a set of ports using a single port set entity is much simpler than that required for maintaining individual ports, especially when considering failures, retransmissions, lifetime expiration, and re-allocations.

Server-side efficiency: Some PCP-controlled devices can allocate port sets in a manner such that data passing through the device is processed much more efficiently than the equivalent using individual port allocations. For example, a CGN having a "bulk" port allocation scheme (see [I-D.ietf-behave-lsn-requirements] section 5) often has this property.

Server-side scalability: The number of mapping entries in PCP-controlled devices is often a limiting factor. Allocating port sets in a single request can result in a single mapping entry being used, therefore allowing greater scalability.

Therefore, while it is functionally possible to obtain the same results using plain MAP, the extension proposed in this document allows greater efficiency, scalability, and simplicity, while lowering latency and necessary network traffic. In a nutshell, PORT_SET is a necessary optimization.

In addition, PORT_SET supports parity preservation. Some protocols (e.g. RTP [RFC3550]) assign meaning to a port number's parity. When mapping sets of ports for the purpose of using such kind of protocol, preserving parity can be necessary.

4. The PORT_SET Option

Option Name: PORT_SET

Number: TBD

Purpose: To map sets of ports.

Valid for Opcodes: MAP

Length: 2 bytes

May appear in: Both requests and responses

Maximum occurrences: 1

NOTE TO IANA (to be removed prior to publication as an RFC): The number is to be assigned by IANA in the range 1-63 (i.e., mandatory to process and created via Standards Action).

The PORT_SET Option indicates that the client wishes to reserve a set of ports. The requested number of ports in that set is indicated in the option.

The PORT_SET Option is formatted as shown in Figure 1.

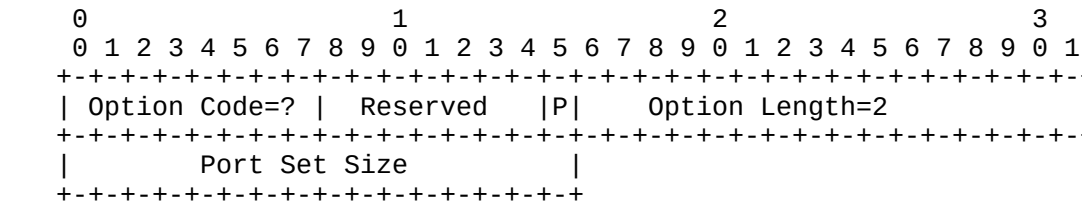


Figure 1: PORT_SET Option

The fields are as follows:

P: 1 if parity preservation is requested, 0 otherwise.

Port Set Size: Number of ports requested. MUST NOT be zero nor one.

NOTE: In its current form, PORT_SET does not support allocating discontinuous port sets. That feature could be added in the future depending on input from the working group.

The Internal Port Set is defined as being the range of Port Set Size ports starting from the Internal Port. The External Port Set is respectively defined as being the range of Port Set Size ports starting from the Assigned External Port. The two ranges always have the same size (i.e., the Port Set Size returned by the server).

4.1. Client Behavior

To retrieve a set of ports, the PCP client adds a PORT_SET option to its PCP MAP request. If port preservation is required, the PCP Client MUST set the parity bit (to 1) to ask the server to preserve the port parity (i.e., the Assigned External Port and Internal Port have the same parity). The PCP client MUST indicate a suggested Port Set Size. A non-null value MUST be used.

The PCP Client MUST NOT include more than one PORT_SET option in a MAP request. If several port sets are needed, the PCP client MUST issue as many MAP requests each of them include a PORT_SET option. These individual MAP request MUST include distinct Internal Port.

If the PORT_SET option is not supported by the server, the PCP client will have to issue individual MAP requests with no PORT_SET option.

4.2. Server Behavior

In addition to regular MAP request processing, the following checks are made upon receipt of a PORT_SET option with non-zero Requested Lifetime:

- If multiple PORT_SET options are present in a single MAP request, a MALFORMED_OPTION error is returned.
- If the Port Set Size is zero or one, a MALFORMED_OPTION error is returned.

If the PREFER_FAILURE option is present and the server is unable to map all ports in the requested External Port Set or is unable to preserve parity (P = 1), the CANNOT_PROVIDE_EXTERNAL error is returned.

If the PREFER_FAILURE option is absent, the server MAY map fewer ports than the value of Port Set Size from the request. It MUST NOT map more ports than the client asked for. In any case, the Internal Port Set MUST always begin from the Internal Port indicated by the client. In particular, if the port mapping failed either because of the unavailability of ports, the PCP Server SHOULD reserve only one external port (i.e., the PCP server ignores the PORT_SET option). If the server ends up mapping only a single port, for any reason, the PORT_SET option MUST NOT be present in the response.

If the PREFER_FAILURE option is absent and port parity preservation is requested (P = 1), the server MAY preserve port parity. In that case, the External Port is set to a value having the same parity as the Internal Port.

If a mapping already exists and the PORT_SET option can be honored, the PCP server updates the mapping with port set information and sends back a positive answer to the requesting PCP client.

If the mapping is successful, the MAP response’s Assigned External Port is set to the first port in the External Port Set, and the PORT_SET option’s Port Set Size is set to number of ports in the mapped port set.

4.3. Port Set Renewal and Deletion

Port set mappings are renewed and deleted as a single entity. That is, the lifetime of all port mappings in the set is set to the Assigned Lifetime at once.

The PORT_SET option MUST be present in a renewal or deletion request. If a server receives a MAP request without a PORT_SET option and whose Internal Port is inside a mapped Internal Port Set, it replies with a MALFORMED_REQUEST error.

5. Operational Considerations

It is totally up to the PCP server to determine the port-set quota for each PCP client. In addition, when the PCP-controlled device supports multiple port-sets delegation for a given PCP client, the PCP client MAY re-initiate a PCP request to get another port set when it has exhausted all the ports within the port-set.

If the PCP server is configured to allocate multiple port-set allocation for one subscriber, the same Assigned External IP Address SHOULD be assigned to one subscriber in multiple port-set requests.

To optimize the number of mapping entries maintained by the PCP server, it is RECOMMENDED to configure the server to assign the maximum allowed port set in a single response. This policy SHOULD be configurable.

The failover mechanism in MAP [section 14 in [I-D.ietf-pcp-base]] and [I-D.boucadair-pcp-failure] can also be applied to port sets.

6. Security Considerations

It is believed that no additional security considerations beyond those discussed in [I-D.ietf-pcp-base] apply to this extension.

7. IANA Considerations

IANA shall allocate a code in the range 1-63 for the new PCP option defined in Section 4.

8. Authors List

The following are extended authors who contributed to the effort:

Yunqing Chen

China Telecom

Room 502, No.118, Xizhimennei Street

Beijing 100035

P.R.China

Chongfeng Xie

China Telecom

Room 502, No.118, Xizhimennei Street

Beijing 100035

P.R.China

Yong Cui

Tsinghua University

Beijing 100084

P.R.China

Phone: +86-10-62603059

Email: yong@csnet1.cs.tsinghua.edu.cn

Qi Sun

Tsinghua University

Beijing 100084

P.R.China

Phone: +86-10-62785822

Email: sunqibupt@gmail.com

Gabor Bajko

Nokia

Email: gabor.bajko@nokia.com

Xiaohong Deng

France Telecom

Email: xiaohong.deng@orange-ftgroup.com

9. Acknowledgements

The authors would like to show sincere appreciation to Alain Durand, Dan Wing, Dave Thaler, Reinaldo Penno, Sam Hartman, and Yoshihiro Ohba, for their useful comments and suggestions.

10. References

10.1. Normative References

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

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

10.2. informative References

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

[I-D.cui-softwire-b4-translated-ds-lite]
Cui, Y., Sun, Q., Boucadair, M., Tsou, T., and Y. Lee, "Lightweight 4over6: An Extension to DS-Lite Architecture", Feb 2012.

[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-10 (work in progress), December 2012.

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

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

Authors' Addresses

Qiong Sun
China Telecom
P.R.China

Phone: 86 10 58552936
Email: sunqiong@ctbri.com.cn

Mohamed Boucadair
France Telecom
Rennes 35000
France

Email: mohamed.boucadair@orange.com

Senthil Sivakumar
Cisco Systems
7100-8 Kit Creek Road
Research Triangle Park, North Carolina 27709
USA

Phone: +1 919 392 5158
Email: ssenthil@cisco.com

Cathy Zhou
Huawei Technologies
Bantian, Longgang District
Shenzhen 518129
P.R. China

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

Simon Perreault
Viagenie
246 Aberdeen
Quebec, QC G1R 2E1
Canada

Phone: +1 418 656 9254
Email: simon.perreault@viagenie.ca

Network Working Group
Internet-Draft
Intended status: Experimental
Expires: May 3, 2012

M. Wasserman
S. Hartman
Painless Security
D. Zhang
Huawei
October 31, 2011

Port Control Protocol (PCP) Authentication Mechanism
draft-wasserman-pcp-authentication-01

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 May 3, 2012.

Copyright Notice

Table of Contents

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 un-authorized users from accessing such devices. This document proposes a PCP security extension which enables PCP servers to authenticate the clients that they are communicating with using Extensible Authentication Protocol (EAP). The following issues are considered in the design of this extension:

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

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

- On Security infrastructure equipment, such as corporate firewalls, that does not create implicit mappings.
- On equipment (such as CGNs or service provider firewalls) that serve multiple administrative domains and do not have a mechanism to securely partition traffic from those domains.
- For any implementation that wants to be more permissive in authorizing explicit mappings than it is in authorizing implicit mappings.
- For implementations that support the THIRD_PARTY Option (unless they can meet the constraints outlined in Section 14.1.2.2).
- For implementations that wish to support any deployment scenario that does not meet the constraints described in Section 14.1.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

Most of the terms used in this document are introduced in [I-D.ietf-pcp-base].

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

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

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

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.

Session Lifetime: A duration associated with a PA session. For an established PA session, the session lifetime is bound to the lifetime of the current authorization decision given to the PCC. The session lifetime can be extended by a new round of EAP authentication before it expires. Until a PA session is established, the lifetime SHOULD be set to a value that allows the PCC to detect a failed session in a reasonable amount of time.

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

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

3. Protocol Details

3.1. Session Initiation

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

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

When a PCC intends to initiate a PA session with a PCS, it sends a PCC-Initiation message to the PCS. The Session ID and Sequence Number fields of the OpCode in the PCC-Initiation are set as 0. 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 identifier assigned by the PCS for this session. Otherwise, the PCS discards the message silently. If the PCC intends to simplify the authentication process, it can append an EAP Identity Response message within the PCC-Initiation request so as to skip over the step of waiting for the EAP Identity Request and inform the PCS that it would like to perform EAP authentication.

In the scenario where a PCS receives a PCP message other than a PCC-Initiation 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 initial 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 and cannot generate a response within a pre-specified period due to certain reasons (e.g., waiting for human input to construct a EAP message), the PCP device needs to reply with a PA-Acknowledge message to indicate that the message has been received. 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, and so a successful EAP authentication process will result in a MSK. If the PCC and the PCS want to generate a traffic key using the MSK, they need to agree upon a Pseudo-Random Function (PRF) for the transport key derivation and a MAC algorithm to provide data origin authentication for subsequent PCP signaling packets. On this occasion, the PCS needs to append the initial PA-Request message with a set of PRF Options and MAC Algorithm Options which contain the PRFs and the MAC (Message Authentication Code) algorithms the PCS supports respectively. After receiving the request, the PCC selects a PRF and a MAC algorithm which it intends to support, and sends back a PA-Answer with a PRF Option and a MAC Algorithm Option for the selected algorithms.

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

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

If the EAP authentication fails, the result code of the last PA-Request is Authentication-Failed. If the EAP authentication 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.

3.2. Session Termination

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

3.3. Result Codes

Following result codes are defined in the solution:

XXX AUTHENTICATION-REQUIRED

XXX AUTHENTICATION-FAILED

XXX AUTHENTICATION-SUCCESS

XXX AUTHORIZATION-FAILED

4. PA Security Association

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

- IP address and UDP port number of the PCC
- IP address and UDP port number of the PCS
- Session Identifier
- Sequence number for the next outgoing PCP message
- Sequence number for the next incoming PCP message
- Last transmitted message payload
- Retransmission interval
- MSK
- MAC algorithm: The algorithm that the transport key should use to generate digests for PCP messages.

- Pseudo-random function: The pseudo random function negotiated in the initial PA-Request and PA-Answer exchange for the transport key derivation

- 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), where:

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

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

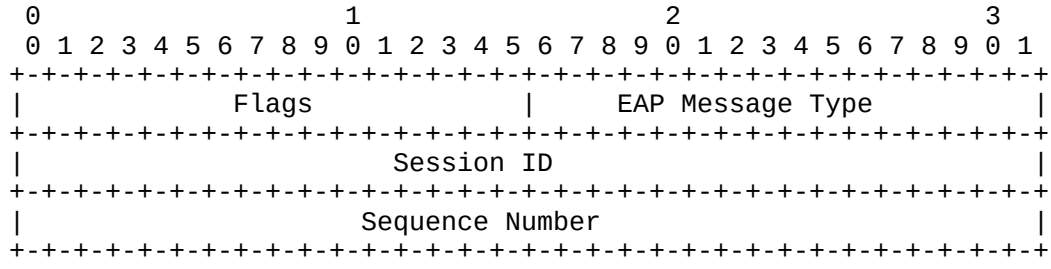
- "IETF PCP": The ASCII code representation of the non-NULL terminated string (excluding the double quotes around it).

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

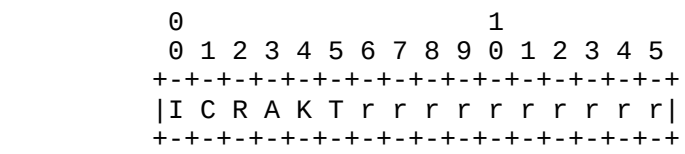
5. Packet Format

5.1. 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.

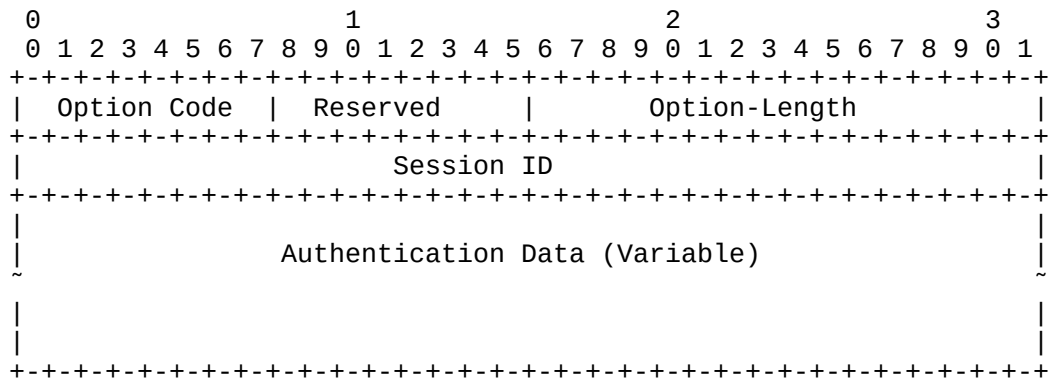
* T (Termination): If this bit is set in a PA-Acknowledgement message, the message is used for session-termination indication.

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

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

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

5.2. Authentication Tag Option

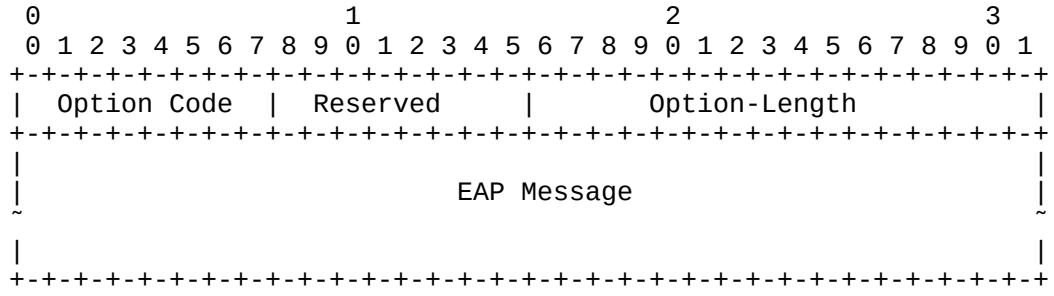


Option-Length: The length of the Authentication Tag Option (in octet), including the 8 octet fixed header and the variable length of the authentication data.

Session ID: A 32-bit field used to indicates 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.

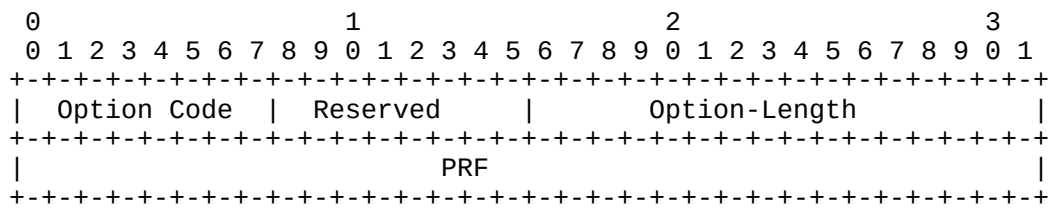
Authentication Data: A Variable length field that carries the Message Authentication Code for the PCP packet. The generation of the digest can be various according to the algorithms specified in different PCP SAs.

5.3. EAP Payload Option



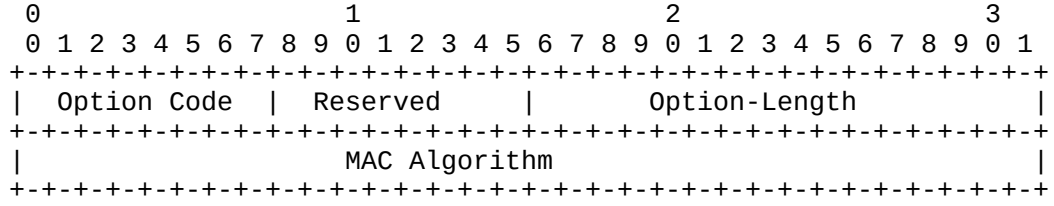
EAP Message: The EAP message transferred. Note this field MUST end on a 32-bit boundary, padded with 0's when necessary.

5.4. PRF Option



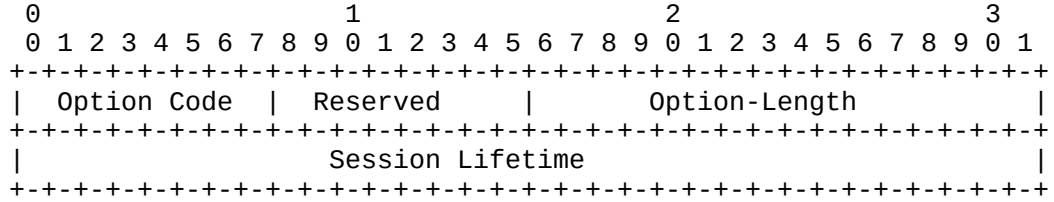
PRF: The pseudo-random Function which the sender supports to generate a MSK.

5.5. Hash Algorithm Option



MAC Algorithm: The MAC algorithm which the sender supports to generate authentication data.

5.6. Session Lifetime Option



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

6. Processing Rules

6.1. Authentication Data Generation

If a PCP SA is generated as the result of an successful EAP authentication process, every subsequent PCP message within the session needs carry an Authentication Tag Option which contains the

digest of the PCP message for data orgin authentication and integrity protection.

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

6.2. Authentication Data Validation

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

6.3. Sequence Number

PCP adopts UDP to transport signaling messages. As an un-reliable transporting protocol, UDP does not guarantee the ordered packet delivery and does not provide any protection from packet loss. In order to ensure the EAP messages are exchanged in a reliable way, every PCP packet exchanged during EAP authentication must 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. If the outgoing packet, the device then increments the sequence number by 1. When receiving a PCP packet from its session partner, the device will not accept it if the sequence number carried in the packet matches the incoming sequence number the device maintains. After confirming that the received packet is valid, the device increments the incoming sequence number by 1. However, the above rules are not applied to PA-Acknowledgement messages. When receiving or sending out a PA-Acknowledgement message, the device MUST not inicrease the correspondent sequence number. Another

exception is message retransmission. When a device does not receive any response message from its session partner in a certain period, it needs to retransmit the last sent message with a limited rate. The value of the sequence number in the duplicate messages MUST be identical to that of the original message. When the device receives such duplicate messages from its session partner, it MUST tries to answer them by sending the last outgoing message within a limited rate unless it has received another valid message with a larger sequence number from its session. Note that in these cases the outgoing sequence number will not be affected by the message retransmission.

6.4. Retransmission Policies

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

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

RT: Retransmission timeout from the previous (re)transmission

IRT: Base value for RT for the initial retransmission

MRC: Maximum retransmission count

MRT: Maximum retransmiting 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) * RTprev

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

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.

6.5. MTU Considerations

TBD

7. IANA Considerations

TBD

8. Security Considerations

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

However, before a transport key has been generated, the PA messages exchanged within a PA session have little cryptographic protection, and if there is no already established security channel between two session partners, these messages are subject to man-in-the-middle attacks and DOS attacks. For instance, the initial PA-Request and PA-Answer exchange is vulnerable to spoofing attacks as these messages are not authenticated and integrity protected. In order to prevent very basic DOS attacks, a PCP device SHOULD generate state information as little as possible in the initial PA-Request and PA-Answer exchanges. The choice of EAP method is also very important. The selected EAP method must be resilient to the attacks possibly

occurred in a insecure network environment, and the user-identity confidentiality, protection against dictionary attacks, and session-key establishment must be supported

9. Acknowledgements

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

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

10. Change Log

10.1. Changes from -00 to -01

- Editorial changes, added use cases to introduction.

11. References

11.1. Normative References

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

11.2. 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-16 (work in progress), October 2011.

[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.

[RFC5191] Forsberg, D., Ohba, Y., Patil, B., Tschofenig, H., and A. Yegin, "Protocol for Carrying Authentication for Network Access (PANA)", RFC 5191, May 2008.

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:

Network Working Group
Internet-Draft
Intended status: Experimental
Expires: September 12, 2012

M. Wasserman
S. Hartman
Painless Security
D. Zhang
Huawei
March 11, 2012

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

Abstract

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

Status of this Memo

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

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

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

This Internet-Draft will expire on September 12, 2012.

Copyright Notice

Table of Contents

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 un-authorized users from accessing such devices. This document proposes a PCP security extension which enables PCP servers to authenticate the clients that they are communicating with using Extensible Authentication Protocol (EAP). The following issues are considered in the design of this extension:

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

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

- On security infrastructure equipment, such as corporate firewalls, that does not create implicit mappings.
- On equipment (such as CGNs or service provider firewalls) that serve multiple administrative domains and do not have a mechanism to securely partition traffic from those domains.
- For any implementation that wants to be more permissive in authorizing explicit mappings than it is in authorizing implicit mappings.
- For implementations that support the THIRD_PARTY Option (unless they can meet the constraints outlined in Section 14.1.2.2).
- For implementations that wish to support any deployment scenario that does not meet the constraints described in Section 14.1.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

Most of the terms used in this document are introduced in [I-D.ietf-pcp-base].

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

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

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

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

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

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

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

3. Protocol Details

3.1. Session Initiation

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

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

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

In the scenario where a PCS receives a non-PA PCP message from a PCC which needs to be authenticated, the PCS can reply with a PA-Request to initiate a PA session; the result code field of the PA-Request is set as AUTHENTICATION-REQUIRED. In addition, the PCS MUST assign a session ID for the session and transfer it within the PA-Request. In the PA messages exchanged afterwards in this session, the session ID MUST be appended. Therefore, in the subsequent communication, the PCC can distinguish the messages in this session from those in other sessions through the PCS IP address and the session ID. When the PCC receives the initial PA-Request message from the PCS, it can reply with a PA-Answer message to continue the session or silently discards the request message according to its local policies.

In a PA session, PA-Request messages are sent from PCSs to PCCs while PA-Answer messages are only sent from PCCs to PCSs. Correspondently, an EAP request messages MUST be transported within a PA-Request message, and an EAP answer messages MUST be transported within a PA-Answer message. Particularly, when a PCP device receives a PA-Request or a PA-Answer message from its partner, the PCP device needs to reply with a PA-Acknowledge message to indicate that the message has been received. This solution is used to deal with the conditions where the device cannot generate a response within a pre-specified period due to certain reasons (e.g., waiting for human input to construct a EAP message). Therefore, the partner does not have to 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, and so a successful EAP authentication process will result in a Master Session Key (MSK). If the PCC and the PCS want to generate a traffic key using the MSK, they need to agree upon a Pseudo-Random Function (PRF) for the transport key derivation and a MAC algorithm to provide data origin authentication for subsequent PCP packets. On this occasion, the PCS needs to append the initial PA-Request message with a set of PRF Options and MAC Algorithm Options. Each PRF Option (MAC Algorithm Option) contains a PRF (MAC (Message Authentication Code) algorithm) that the PCS supports. After receiving the request, the PCC selects a PRF and a MAC algorithm which it intends to support, and sends back a PA-Answer with a PRF Option and a MAC Algorithm Option for the selected algorithms.

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

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

If the EAP authentication fails, the result code of the last PA-Request is AUTHENTICATION-FAILED. If the EAP authentication 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.

3.2. Session Termination

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

3.3. Result Codes

Following result codes are defined in the solution:

XXX AUTHENTICATION-REQUIRED

XXX AUTHENTICATION-FAILED

XXX AUTHENTICATION-SUCCESS

XXX AUTHORIZATION-FAILED

4. PA Security Association

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

- IP address and UDP port number of the PCC
- IP address and UDP port number of the PCS
- Session Identifier
- Sequence number for the next outgoing PCP message
- Sequence number for the next incoming PCP message
- Last outgoing message payload
- Retransmission interval
- MSK
- MAC algorithm: The algorithm that the transport key should use to generate digests for PCP messages.

- Pseudo-random function: The pseudo random function negotiated in the initial PA-Request and PA-Answer exchange for the transport key derivation

- 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:

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

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

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

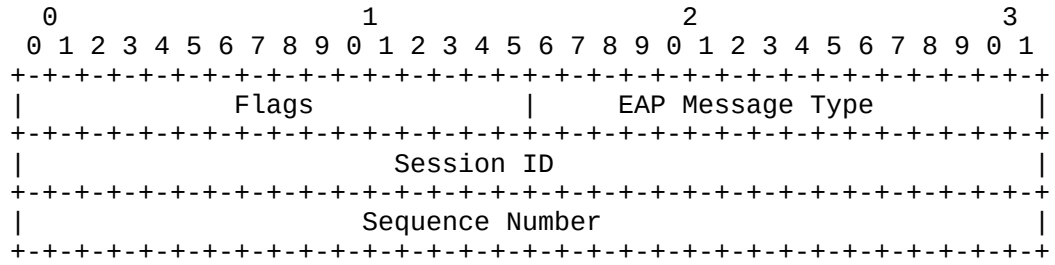
- Session_ID: The ID of the session which the MSK is derived from

- Key ID: The ID assigned for the traffic key

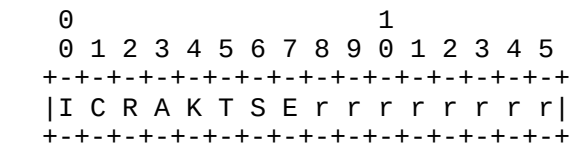
5. Packet Format

5.1. Authentication OpCode Format

The following figure illustrates the format of an authentication Opcode:



Flags: The Flags field is two octets. The following bits are assigned:



* I (Initiation): This bit is set in a PCC-Initiation message.

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

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

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

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

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

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

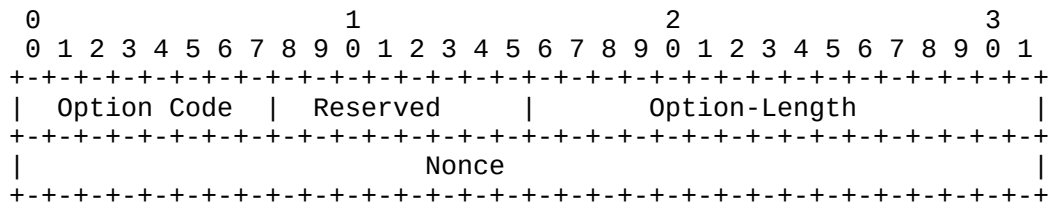
* E (Fragmentation start): This bit is set in a PA message which contain the last fragment of a EAP message.

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

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

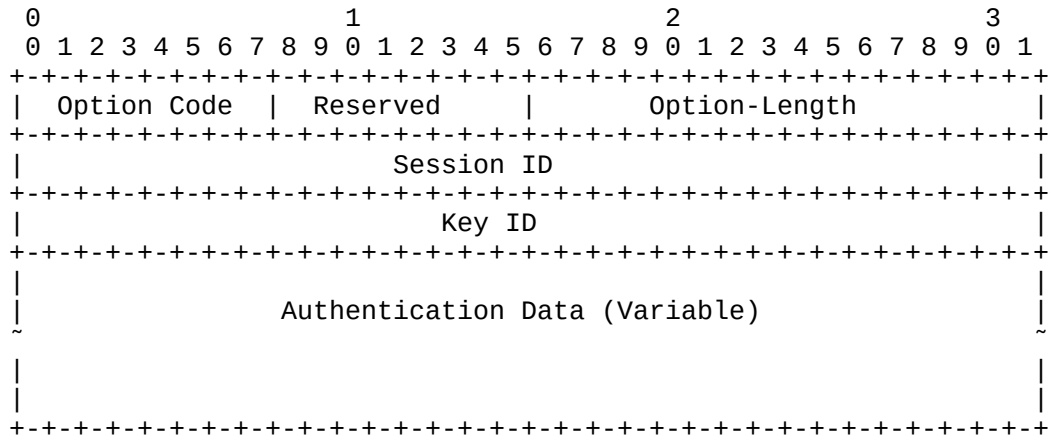
Sequence Number: This field contains a 32-bit sequence number. In this solution, a sequence number needs to be incremented on every new (non-retransmission) outgoing packet in order to provide ordering guarantee for PCP.

5.2. Nonce Option



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

5.3. Authentication Tag Option



Option-Length: The length of the Authentication Tag Option (in octet), including the 8 octet fixed header and the variable length of the authentication data.

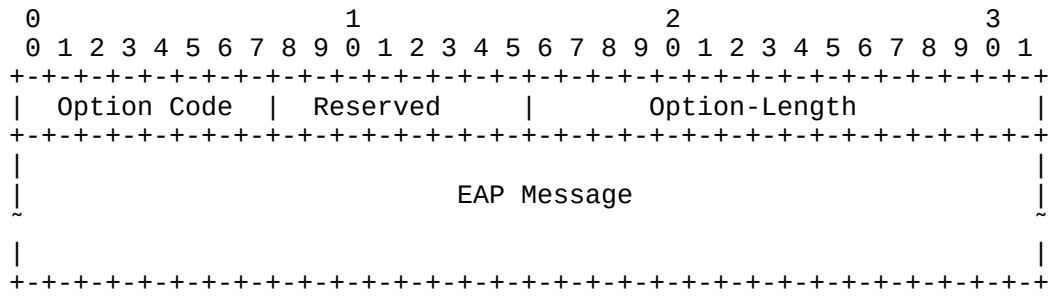
Session ID: A 32-bit field used to indicates the identifier of the session that the message belongs to and identifies the secret key used to create the message digest appended to the PCP message.

Key ID: The ID associated with the traffic key used to generate authentication data. This field is filled with zero if MSK is directly used to secure the message.

Authentication Data: A Variable length field that carries the Message Authentication Code for the PCP packet. The generation of the digest can be various according to the algorithms specified in

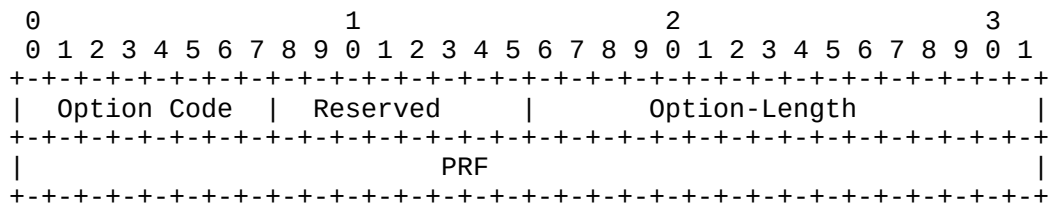
different PCP SAs.

5.4. EAP Payload Option



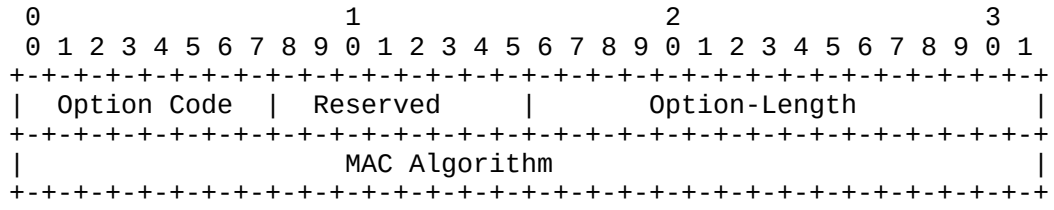
EAP Message: The EAP message transferred. Note this field MUST end on a 32-bit boundary, padded with 0's when necessary.

5.5. PRF Option



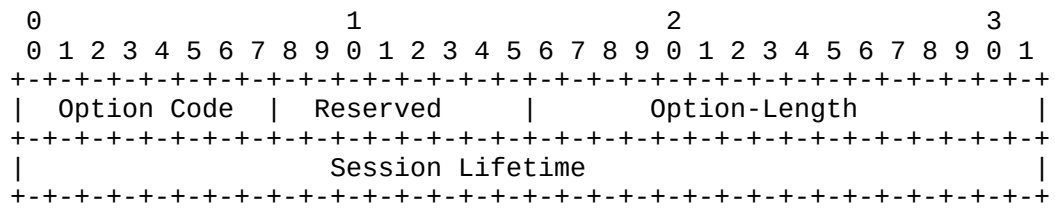
PRF: The pseudo-random Function which the sender supports to generate a MSK.

5.6. Hash Algorithm Option



MAC Algorithm: The MAC algorithm which the sender supports to generate authentication data.

5.7. Session Lifetime Option



Session Lifetime: The life period of the PA Session, which is decided by the authorization result.

6. Processing Rules

6.1. Authentication Data Generation

If a PCP SA is generated as the result of an successful EAP authentication process, every subsequent PCP message within the session needs carry an Authentication Tag Option which contains the digest of the PCP message for data orgin authentication and integrity protection.

Before generating a digest for a PCP message, a device needs to first select a traffic key in the session and append the Authentication Tag Option at the end of the protected PCP message. The length of the Authentication Data field is decided by the MAC algorithm adopted in the session. The device then fills the Session ID field and the PCP SA ID field, and sets the Authentication Data field as 0. After this, the device generates a digest for the PCP message with the MAC algorithm and the selected traffic key, and input the generated digest into the Authentication Data field.

6.2. Authentication Data Validation

When a device receives a PCP pacekt with an Authentication Tag Option, it needs to use the session ID transported in the option to locate the proper SA, and then find out the associated transport key and the MAC algorithm. After storing the value of the Authentication field of the Authentication Tag Option, the device fills the the Authentication field with zeros. Then, the device generates a digest for the packet with the transport key and the MAC algorithm found in the first step. If the value of the newly generated digest is identical to the stored one, the device can ensure that the packet has not been tampered during the transportation. The validation successes. Otherwise, the packet MUST be discarded.

6.3. Sequence Number

PCP adopts UDP to transport signaling messages. As an un-reliable transporting protocol, UDP does not guarantee the ordered packet delivery and does not provide any protection from packet loss. In order to ensure the EAP messages are exchanged in a reliable way, every PCP packet exchanged during EAP authentication must 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 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 matche the incoming sequence number the device maintains.

After confirming that the received packet is valid, the device increments the incoming sequence number by 1. However, the above rules are not applied to PA-Acknowledgement messages. When receiving or sending out a PA-Acknowledgement message, the device does not inicrease the correspondent sequence number. Another exception is message retransmission. When a device does not receive any response message from its session partner in a certain period, it needs to retransmit the last sent message with a limited rate. The duplicate messages and the original message MUST use the identical sequence number. When the device receives such duplicate messages from its session partner, it MUST tries to answer them by sending the last outgoing message with a limited rate unless it has received another valid message with a larger sequence number from its session. Note that in these cases the incoming and outgoing sequence number will not be affected by the message retransmission.

6.4. Retransmission Policies

This work provides a retransmission mechanism for reliable PA message delivery. The timer, the variables, and the rules used in this mechanism is mostly brought from PANA[RFC5191].

The retransmission behavior is controlled and described by the following variables:

RT: Retransmission timeout from the previous (re)transmission

IRT: Base value for RT for the initial retransmission

MRC: Maximum retransmission count

MRT: Maximum retransmiting 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) * RTprev

MRT specifies an upper bound on the value of RT (disregarding the randomization added by the use of RAND). If MRT has a value of 0, there is no upper limit on the value of RT. Otherwise:

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.

6.5. MTU Considerations

The fragmentation and reassembly of EAP messages must be provided in order to ensure the length of a PA message is not larger than the MTU of the link that it will be transported through. Therefore, a PA message may only transport a fragment of an EAP message. Becasue any loss or tamper of a EAP frgament will be detected and sequencing information is provided, fragmentation support can be added in a simple manner. Particularly, the S bit is set in a PA message which contain the first fragment of a EAP message, and the The E bit is set

in a PA message which contain the last fragment of a EAP message.

7. IANA Considerations

TBD

8. Security Considerations

In this work, a successful EAP authentication process performed between two PCP devices will result in the generation of a MSK which can be used to derive the transport keys to generate MAC digests for subsequent PCP message exchanges. This work does not exclude the possibility of using the MSK to generate keys for different security protocols to enable per-packet cryptographic protection. The methods of deriving the transport key for the security protocols is out of scope of this document.

However, before a transport key has been generated, the PA messages exchanged within a PA session have little cryptographic protection, and if there is no already established security channel between two session partners, these messages are subject to man-in-the-middle attacks and DOS attacks. For instance, the initial PA-Request and PA-Answer exchange is vulnerable to spoofing attacks as these messages are not authenticated and integrity protected. In order to prevent very basic DOS attacks, a PCP device SHOULD generate state information as little as possible in the initial PA-Request and PA-Answer exchanges. The choice of EAP method is also very important. The selected EAP method must be resilient to the attacks possibly occurred in a insecure network environment, and the user-identity confidentiality, protection against dictionary attacks, and session-key establishment must be supported.

9. Acknowledgements

This document was written using the xml2rfc tool described in RFC 2629 [RFC2629].

Some of the ideas in this document were adopted from PANA[RFC5191].

10. Change Log

10.1. Changes from -00 to -01

- Editorial changes, added use cases to introduction.

10.2. Changes from -01 to -02

- 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.
- Add the key ID field into the authentication tag option so that a MSK can generate multiple traffic keys.
- 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.
- Add the support of fragmenting EAP messages.

11. References

11.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

11.2. Informative 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.

[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.

[RFC5191] Forsberg, D., Ohba, Y., Patil, B., Tschofenig, H., and A. Yegin, "Protocol for Carrying Authentication for Network Access (PANA)", RFC 5191, May 2008.

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: