

PCP Working Group
Internet-Draft
Intended status: Standards Track
Expires: August 19, 2013

M. Boucadair
France Telecom
R. Penno
Cisco
February 15, 2013

Port Control Protocol (PCP) Failure Scenarios
draft-boucadair-pcp-failure-05

Abstract

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

A procedure to retrieve the explicit dynamic mapping(s) from the PCP Server is proposed. This procedure relies upon the use of a new PCP OpCode and Option: GET/NEXT.

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 19, 2013.

Copyright Notice

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

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

Table of Contents

1. Introduction	3
2. PCP Client Failure Scenarios	3
2.1. Change of the IP Address of The PCP Server	3
2.2. Application Crash	3
2.3. PCP Client Crash	4
2.4. Change of the Internal IP Address	5
2.5. Change of the CPE WAN IP Address	5
2.6. UPnP IGD/PCP IWF	6
3. Restart or Failure of the PCP Server	6
3.1. Basic Rule	6
3.2. Clear PCP Mappings	7
3.3. State Redundancy is Enabled	7
3.4. Cold-Standby without State Redundancy	7
3.5. Anycast Redundancy Mode	7
3.6. Mapping Repair Procedure	7
3.6.1. PCP Client Behaviour	7
3.6.2. PCP Proxy Behaviour	8
4. Security Considerations	8
5. IANA Considerations	8
6. Acknowledgements	9
7. References	9
7.1. Normative References	9
7.2. Informative References	9
Appendix A. PCP State Synchronization: Overview	10
Appendix B. GET/NEXT Operation	10
B.1. OpCode Format	10
B.2. OpCode-Specific Result Code	12
B.3. Ordering and Equality	12
B.4. NEXT Option	12
B.5. GET/NEXT PCP Client Theory of Operation	16
B.6. GET/NEXT PCP Server Theory of Operation	16
B.7. Flow Examples	17
Authors' Addresses	21

1. Introduction

This document discusses several failure scenarios that may occur when deploying PCP [I-D.ietf-pcp-base].

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 [I-D.ietf-pcp-base]); 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.

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]). 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) induced the same security threatened as those discussed in [I-D.ietf-pcp-base].

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.

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 are implementation-specific and are out of scope of this document.

[I-D.ietf-pcp-base] does not allow anymore 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. 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. 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.

If the PCP Client (or the application) is crashing, it should be allocating short PCP lifetimes until it is debugged and running properly. If it is never debugged and never running properly, it should continue to request short PCP lifetimes.

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.

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 (but also some side effect as discussed above). Note PCP does not support such functionality anymore.

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.

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:

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

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.

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. This issue can be softened 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.

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.

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.

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.

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 [I-D.ietf-pcp-base]) may help to detect the loss of state and therefore to re-create missing explicit dynamic mappings.

3.6. Mapping Repair Procedure

3.6.1. PCP Client Behaviour

[I-D.ietf-pcp-base] defines a procedure for the PCP Server to notify PCP Clients about changes related to the mappings it maintains. Indeed, the PCP Server can send unsolicited ANNOUNCE OpCode or unsolicited MAP/PEER responses. When unsolicited ANNOUNCE is

received, the PCP Client proceeds to re-installing its mappings. Unsolicited PCP MAP/PEER responses received from a PCP Server are handled as any normal MAP/PEER response.

3.6.2. PCP Proxy Behaviour

Upon receipt of an unsolicited ANNOUNCE response from a PCP Server, the PCP Proxy proceeds to renewing the mappings and checks whether there are changes compared to a local cache if it is maintained by the PCP Proxy. If no change is detected, no unsolicited ANNOUNCE is generated towards PCP Clients. If a change is detected, the PCP Proxy MUST generate unsolicited ANNOUNCE message(s) to appropriate PCP Clients. If the PCP Proxy does not maintain a local cache for the mappings, unsolicited ANNOUNCE messages are relayed to PCP Clients.

Unsolicited PCP MAP/PEER responses received from a PCP Server are handled as any normal MAP/PEER response. To handle unsolicited PCP MAP/PEER responses, the PCP Proxy is required to maintain a local cache of instantiated mappings in the PCP Server. When this service is supported the state SHOULD be recovered in case of failures using the procedure defined in Appendix B.

Upon change of its external IP address, the PCP Proxy SHOULD renew the mappings it maintained. This can be achieved only if a full state table is maintained by the PCP Proxy.

4. Security Considerations

TBC.

5. IANA Considerations

The following OpCode is requested:

- o GET

The following Option code is requested:

- o NEXT

The following error codes are requested:

- o NONEXIST_MAP

- o AMBIGUOUS

6. Acknowledgements

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

7. References

7.1. Normative References

- [I-D.ietf-pcp-base]
Wing, D., Cheshire, S., Boucadair, M., Penno, R., and P. Selkirk, "Port Control Protocol (PCP)", draft-ietf-pcp-base-29 (work in progress), November 2012.
- [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-06 (work in progress), December 2012.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

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:

- o One element (i.e., PCP Client/host/application, PCP Server, PCP Proxy, PCP IWF) of the chain is REQUIRED to use stable storage
- o If the PCP Client (resp., the PCP Server) crashes and restarts it just have to synchronize with the PCP Server (resp., the PCP Client);
- o 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);
- o 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.
- o 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.

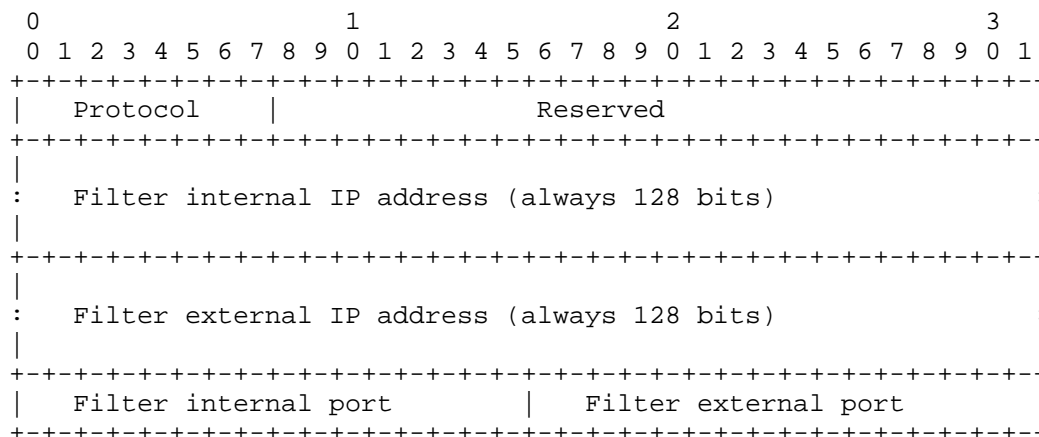


Figure 1: GET: OpCode format

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 [I-D.ietf-pcp-base].

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 [I-D.ietf-pcp-base].

Filter external IP address: Conveys the external IP address (including an unspecified IPv4IPv6 address). The encoding of this field follows Section 5 of [I-D.ietf-pcp-base].

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:

- o same protocol and;
- o same internal address and;
- o same external address and;
- o same internal port and;
- o 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 [I-D.ietf-pcp-base].

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

Version=1

0										1										2										3									
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1								
Protocol										Reserved										MORE/END																			
Mapping internal IP address (always 128 bits)																																							
Mapping external IP address (always 128 bits)																																							
Mapping remaining lifetime																																							
Mapping internal port																Mapping external port																							
Mapping Options																																							

Version=2

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
+-----+																					

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 [I-D.ietf-pcp-base]).

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 [I-D.ietf-pcp-base].

Mapping external IP address: the external address of the returned mapping. The encoding of this field follows Section 5 of [I-D.ietf-pcp-base].

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 ask 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:

- o 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.
- o 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.

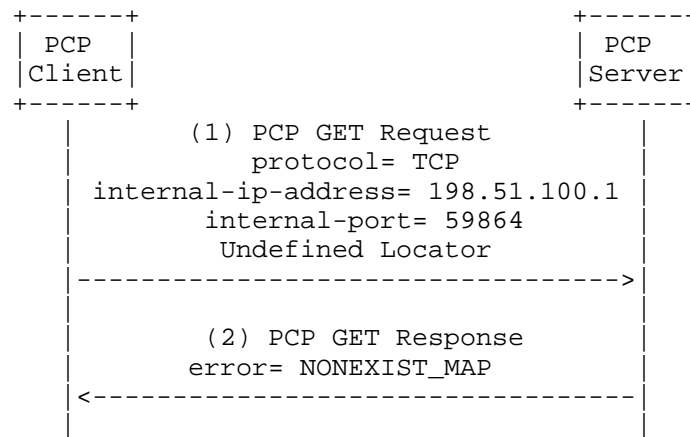


Figure 3: Example of a failed GET operation

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

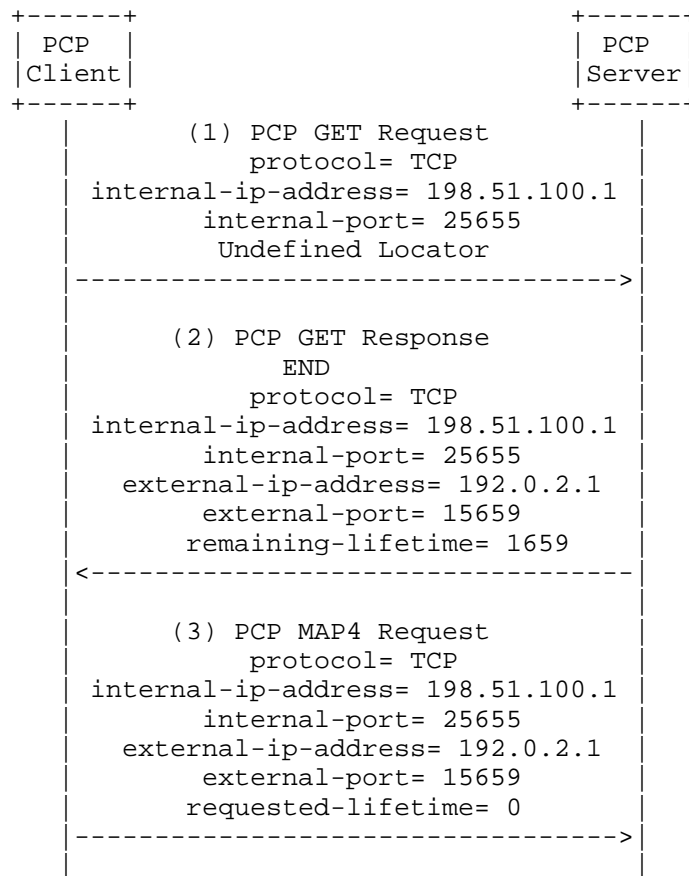


Figure 4: Example of a successful GET operation

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.

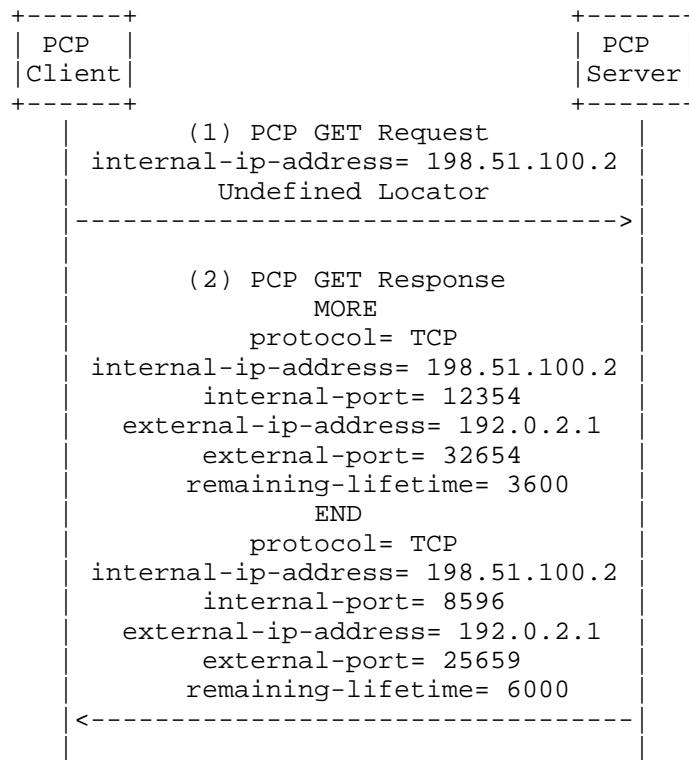


Figure 5: Flow example of GET/NEXT

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.

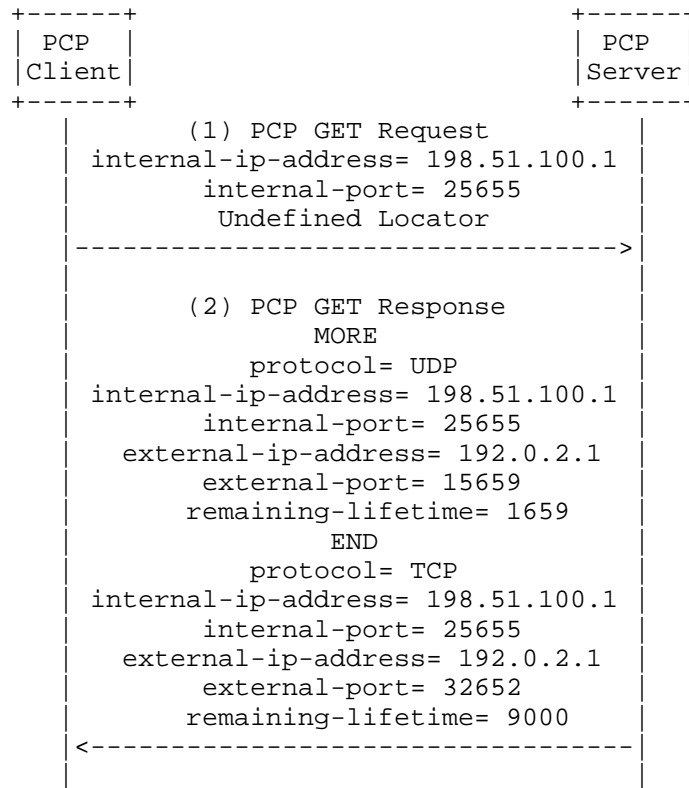


Figure 6: Flow example of GET/NEXT: same internal port number

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: Informational
Expires: September 24, 2015

M. Boucadair
France Telecom
March 23, 2015

Port Control Protocol (PCP) Flow Examples
draft-boucadair-pcp-flow-examples-04

Abstract

This document provides a set of examples to illustrate Port Control Protocol (PCP) operations. It is a companion document to the base PCP specification.

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 24, 2015.

Copyright Notice

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

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

Table of Contents

1. Introduction	3
2. Basic MAP Operations	3
2.1. Suggested External Port Honored by the PCP Server	3
2.2. IPv6-enabled PCP Client	4
2.3. Remove an Existing Mapping	5
2.4. Suggested External Port Not Honored by the PCP Server	6
2.5. Suggested External IP Address	7
2.6. Create Mapping with Distinct External IP Addresses	8
2.7. Mapping Nonce Doesn't Match: Base PCP Specification	11
2.8. Mapping Nonce Doesn't Match: Updated Specification	11
2.9. PREFER_FAILURE Option: Requested Port is Honored	12
2.10. PREFER_FAILURE Option: Requested Port is not Honored	13
2.11. Negative Impact of PREFER_FAILURE Option	14
2.12. Existing Implicit Mapping	15
2.13. Shortening a Mapping Lifetime in the Presence of Client- Originated Traffic	17
2.14. Create a Mapping for All Incoming Traffic of a Given Protocol	17
2.15. Create a Mapping for All Protocols	18
2.16. Malformed Request	18
2.17. Exceeded Port Quota	19
2.18. Unsupported Address Family	20
2.19. Unsupported Protocol	20
2.20. Unsolicited MAP Response	21
2.21. Mapping Repair	22
3. NAT Detect Example	23
4. Retrieve the External IP Address	24
5. THIRD_PARTY Examples	25
5.1. THIRD_PARTY Enabled at the Server Side	25
5.2. THIRD_PARTY Disabled at the Server Side	26
5.3. Malformed Request	26
6. MAP with FILTER Examples	27
6.1. Basic Filter Usage	27
6.2. Remove All Filters	28
6.3. Change an Existing Filter	29
7. Assess the Reachability of the PCP Server	30
8. PEER Operations	31
8.1. No Mapping Exists for the Internal Port Number	31
8.2. A Mapping Exists for the External Port Number	32
8.3. External IP Address Cannot be Honored	33
8.4. Extend the Lifetime	34
8.5. Learn the Lifetime of a Mapping	35
9. Version Negotiation	36
10. Security Considerations	37
11. IANA Considerations	37
12. Acknowledgements	37

13. References	37
13.1. Normative References	37
13.2. Informative References	37
Author's Address	37

1. Introduction

As a companion document to [RFC6887], this document provides examples to help understanding the PCP machinery and exchanged PCP messages in various usage contexts.

For more details about PCP protocol specification, the reader is invited to refer to [RFC6887].

Examples included in this document make use of the IPv4 and IPv6 address blocks for documentation purposes defined in [RFC5737] and [RFC3849].

2. Basic MAP Operations

The following figure illustrates the messages which are exchanged to create a mapping in a PCP-controlled device with MAP opcode.

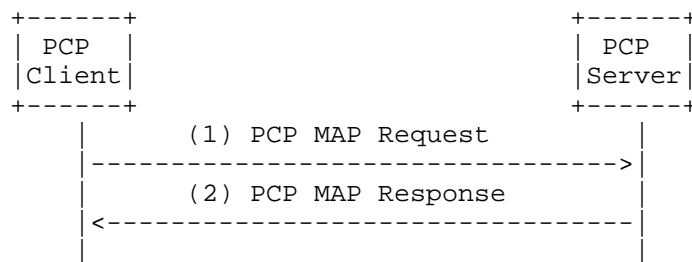


Figure 1: Example of creating a mapping

The following sub-sections provide several examples depending on the content of the MAP request and the decision of the PCP server.

2.1. Suggested External Port Honored by the PCP Server

This example illustrates the content of exchanged PCP messages when the PCP client does not include any PCP Option in its request. In this example, the PCP server assigns the suggested port number. In reference to Figure 1, the content of exchanged PCP messages is as follows:

```
Version: 2
R bit: Request (0)
opcode: MAP (0x01)
Requested Lifetime: 36000 sec
PCP client's IP Address: ::ffff:198.51.100.1
MAP Request:
  Mapping Nonce: 15685
  Protocol: UDP (17)
  Internal Port: 3938
  Suggested External Port: 3938
  Suggested External IP Address: ::ffff:0.0.0.0
```

Figure 2: MAP request (suggested External Port Honored by the PCP Server)

```
Version: 2
R bit: Response (1)
opcode: MAP (0x01)
Result Code: 0
Lifetime: 20000 sec
Epoch Time: 1250
MAP Response:
  Mapping Nonce: 15685
  Protocol: UDP (17)
  Internal Port: 3938
  Assigned External Port: 3938
  Assigned External IP Address: ::ffff:192.0.2.1
```

Figure 3: MAP response (suggested External Port Honored by the PCP Server)

2.2. IPv6-enabled PCP Client

This example illustrates the content of exchanged PCP messages when the PCP client is assigned with an IPv6 address but the remote server controls a NAT44 device. In reference to Figure 1, the content of exchanged PCP messages is as follows:

```
Version: 2
R bit: Request (0)
opcode: MAP (0x01)
Requested Lifetime: 36000 sec
PCP client's IP Address: 2001:db8:0:0:1::1
MAP Request:
  Mapping Nonce: 15685
  Protocol: UDP (17)
  Internal Port: 3938
  Suggested External Port: 3938
  Suggested External IP Address: ::ffff:0.0.0.0
```

Figure 4: MAP request (suggested External Port Honored by the PCP Server)

```
Version: 2
R bit: Response (1)
opcode: MAP (0x01)
Result Code: 0
Lifetime: 20000 sec
Epoch Time: 1250
MAP Response:
  Mapping Nonce: 15685
  Protocol: UDP (17)
  Internal Port: 3938
  Assigned External Port: 3938
  Assigned External IP Address: ::ffff:192.0.2.1
```

Figure 5: MAP response (suggested External Port Honored by the PCP Server)

2.3. Remove an Existing Mapping

This example illustrates the content of exchanged PCP messages when the PCP client request the removal of an existing mapping.

```
Version: 2
R bit: Request (0)
opcode: MAP (0x01)
Requested Lifetime: 0 sec
PCP client's IP Address: ::ffff:198.51.100.1
MAP Request:
  Mapping Nonce: 15685
  Protocol: UDP (17)
  Internal Port: 3938
  Suggested External Port: 3938
  Assigned External IP Address: ::ffff:192.0.2.1
```

Figure 6: MAP request (Remove an Existing Mapping)

```
Version: 2
R bit: Response (1)
opcode: MAP (0x01)
Result Code: 0
Lifetime: 0 sec
Epoch Time: 1250
MAP Response:
  Mapping Nonce: 15685
  Protocol: UDP (17)
  Internal Port: 3938
  Assigned External Port: 3938
  Assigned External IP Address: ::ffff:192.0.2.1
```

Figure 7: MAP response (Remove an Existing Mapping)

2.4. Suggested External Port Not Honored by the PCP Server

This example illustrates the content of exchanged PCP messages when the PCP client does not include any PCP Option in its request. In this example, the PCP server does not assign the suggested external port number. In reference to Figure 1, the content of exchanged PCP messages is as follows:

```
Version: 2
R bit: Request (0)
opcode: MAP (0x01)
Requested Lifetime: 36000 sec
PCP client's IP Address: ::ffff:198.51.100.1
MAP Request:
  Mapping Nonce: 15685
  Protocol: UDP (17)
  Internal Port: 11000
  Suggested External Port: 11000
  Suggested External IP Address: ::ffff:0.0.0.0
```

Figure 8: MAP request (Suggested External Port Not Honored by the PCP Server)

```
Version: 2
R bit: Response (1)
opcode: MAP (0x01)
Result Code: 0
Lifetime: 20000 sec
Epoch Time: 1250
MAP Response:
  Mapping Nonce: 15685
  Protocol: UDP (17)
  Internal Port: 11000
  Assigned External Port: 15200
  Assigned External IP Address: ::ffff:192.0.2.1
```

Figure 9: MAP response (Suggested External Port Not Honored by the PCP Server)

2.5. Suggested External IP Address

This example illustrates the content of exchanged PCP messages when the PCP client does not include any PCP Option in its request. In this example, the PCP client indicates a hinted external IP address honored by the PCP server. In reference to Figure 1, the content of exchanged PCP messages is as follows:

```
Version: 2
R bit: Request (0)
opcode: MAP (0x01)
Requested Lifetime: 36000 sec
PCP client's IP Address: ::ffff:198.51.100.1
MAP Request:
  Mapping Nonce: 15685
  Protocol: UDP (17)
  Internal Port: 3938
  Suggested External Port: 3938
  Suggested External IP Address: ::ffff:192.0.2.1
```

Figure 10: MAP request (Suggested External IP Address)

```
Version: 2
R bit: Response (1)
opcode: MAP (0x01)
Result Code: 0
Lifetime: 20000 sec
Epoch Time: 1250
MAP Response:
  Mapping Nonce: 15685
  Protocol: UDP (17)
  Internal Port: 3938
  Assigned External Port: 15200
  Assigned External IP Address: ::ffff:192.0.2.1
```

Figure 11: MAP response (Suggested External IP Address)

2.6. Create Mapping with Distinct External IP Addresses

Figure 12 shows a PCP server with a pool of public IPv4 addresses (192.0.2/24) and two PCP clients associated with different subscribers. The PCP clients each make a port mapping request to the PCP server which creates the mapping from its 192.0.2/24 pool.

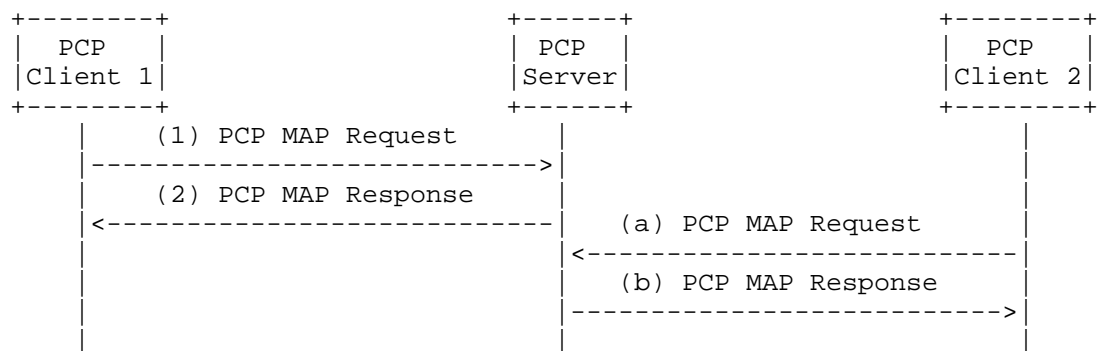


Figure 12: Example of creating mappings with distinct external IP addresses

In this example, the PCP clients were mapped to different public addresses as illustrated in the content of the PCP messages listed below.

The content of PCP messages exchanged between PCP client 1 and the PCP server is as follows:

```

Version: 2
R bit: Request (0)
opcode: MAP (0x01)
Requested Lifetime: 36000 sec
PCP client's IP Address: ::ffff:198.51.100.1
MAP Request:
  Mapping Nonce: 15685
  Protocol: TCP (6)
  Internal Port: 15333
  Suggested External Port: 15333
  Suggested External IP Address: ::ffff:0.0.0.0
  
```

Figure 13: MAP request (PCP Client 1)

```
Version: 2
R bit: Response (1)
opcode: MAP (0x01)
Result Code: 0
Lifetime: 20000 sec
Epoch Time: 1250
MAP Response:
  Mapping Nonce: 15685
  Protocol: TCP (6)
  Internal Port: 15333
  Assigned External Port: 12000
  Assigned External IP Address: ::ffff:192.0.2.1
```

Figure 14: MAP response (PCP Client 1)

The content of PCP messages exchanged between PCP client 2 and the PCP server is as follows:

```
Version: 2
R bit: Request (0)
opcode: MAP (0x01)
Requested Lifetime: 36000 sec
PCP client's IP Address: ::ffff:198.51.100.2
MAP Request:
  Mapping Nonce: 59869
  Protocol: UDP (17)
  Internal Port: 12000
  Suggested External Port: 12000
  Suggested External IP Address: ::ffff:0.0.0.0
```

Figure 15: MAP request (PCP Client 2)

```
Version: 2
R bit: Response (1)
opcode: MAP (0x01)
Result Code: 0
Lifetime: 20000 sec
Epoch Time: 1250
MAP Response:
  Mapping Nonce: 59869
  Protocol: UDP (17)
  Internal Port: 12000
  Assigned External Port: 6000
  Assigned External IP Address: ::ffff:192.0.2.2
```

Figure 16: MAP response (PCP Client 2)

2.7. Mapping Nonce Doesn't Match: Base PCP Specification

CAUTION: The behavior described in this section is obsoleted by [I-D.cheshire-pcp-unsupp-family]. This section records the behavior as initially specified the base PCP specification [RFC6887].

This example illustrates the content of exchanged PCP messages when the PCP client does not include any PCP Option in its request. In this example, the PCP client indicates a distinct Mapping Nonce than the one stored by the PCP server. In reference to Figure 1, the content of exchanged PCP messages is as follows:

```
Version: 2
R bit: Request (0)
opcode: MAP (0x01)
Requested Lifetime: 36000 sec
PCP client's IP Address: ::ffff:198.51.100.1
MAP Request:
  Mapping Nonce: 45687
  Protocol: UDP (17)
  Internal Port: 3938
  Suggested External Port: 3938
  Suggested External IP Address: ::ffff:192.0.2.1
```

Figure 17: MAP request (Mapping Nonce Doesn't Match)

```
Version: 2
R bit: Response (1)
opcode: MAP (0x01)
Result Code: NOT_AUTHORIZED (0x02)
Lifetime: 35550 sec
Epoch Time: 1300
```

Figure 18: MAP response (Mapping Nonce Doesn't Match)

2.8. Mapping Nonce Doesn't Match: Updated Specification

Nonce validation checks are problematic in various scenarios as discussed in [I-D.cheshire-pcp-unsupp-family]. As a consequence, the nonce validation checks are relaxed as follows: If operating in the Simple Threat Model (Section 18.1 of the PCP specification [RFC6887]), and the internal port, protocol, internal address, and external address family match an existing explicit dynamic mapping, but the mapping nonce does not match, then the existing mapping is not modified in any way, and a valid PCP reply is returned to the client, using the client-specified nonce, reporting the external address, port, and remaining lifetime of the existing mapping. An example is shown in Figure 19 and Figure 3.

The request shown in Figure 19 matches an existing mapping (see Figure 3). Even if the nonce of the exiting mapping does not match the one indicated in the request, a positive answer is returned to the requesting PCP client without any change to the existing mapping. The nonce of the existing mapping (i.e., 15685) is not returned in the response.

```
Version: 2
R bit: Request (0)
opcode: MAP (0x01)
Requested Lifetime: 36000 sec
PCP client's IP Address: ::ffff:198.51.100.1
MAP Request:
  Mapping Nonce: 45687
  Protocol: UDP (17)
  Internal Port: 3938
  Suggested External Port: 3938
  Suggested External IP Address: ::ffff:192.0.2.1
```

Figure 19: MAP request

```
Version: 2
R bit: Response (1)
opcode: MAP (0x01)
Result Code: 0
Lifetime: 10000 sec
Epoch Time: 3500
MAP Response:
  Mapping Nonce: 45687
  Protocol: UDP (17)
  Internal Port: 3938
  Assigned External Port: 3938
  Assigned External IP Address: ::ffff:192.0.2.1
```

Figure 20: MAP response

2.9. PREFER_FAILURE Option: Requested Port is Honored

This flow shows an example of the content of PCP messages that will be exchanged to create a mapping in a PCP-controlled device. In this example, the PCP client indicates a requested external UDP port number and also a PREFER_FAILURE Option. In this example, we suppose the requested port can be honored by the PCP server. In reference to Figure 1, the content of exchanged PCP messages is as follows:

```
Version: 2
R bit: Request (0)
opcode: MAP (0x01)
Requested Lifetime: 36000 sec
PCP client's IP Address: ::ffff:198.51.100.1
MAP Request:
  Mapping Nonce: 15685
  Protocol: UDP (17)
  Internal Port: 1234
  Suggested External Port: 12536
  Suggested External IP Address: ::ffff:0.0.0.0
Option Code: PREFER_FAILURE (0x02) Option Length: 0 bytes Data: (NULL)
```

Figure 21: MAP request (PREFER_FAILURE Option: Requested Port is Honored)

```
Version: 2
R bit: Response (1)
opcode: MAP (0x01)
Result Code: 0
Lifetime: 36000 sec
Epoch Time: 1250
MAP Response:
  Mapping Nonce: 15685
  Protocol: UDP (17)
  Internal Port: 1234
  Assigned External Port: 12536
  Assigned External IP Address: ::ffff:192.0.2.1
```

Figure 22: MAP response (PREFER_FAILURE Option: Requested Port is Honored)

2.10. PREFER_FAILURE Option: Requested Port is not Honored

This flow shows an example of the content of PCP messages that will be exchanged to create a mapping in a PCP-controlled device. In this example, the PCP client indicates a requested external UDP port number and also a PREFER_FAILURE Option. In this example, we suppose the requested port cannot be honored by the PCP server. In reference to Figure 1, the content of exchanged PCP messages is as follows:

```
Version: 2
R bit: Request (0)
opcode: MAP (0x01)
Requested Lifetime: 36000 sec
PCP client's IP Address: ::ffff:198.51.100.1
MAP Request:
  Mapping Nonce: 15685
  Protocol: UDP (17)
  Internal Port: 1234
  Suggested External Port: 1234
  Suggested External IP Address: ::ffff:0.0.0.0
Option Code: PREFER_FAILURE (0x02) Option Length: 0 bytes Data: (NULL)
```

Figure 23: MAP request (PREFER_FAILURE Option: Requested Port is not Honored)

```
Version: 2
R bit: Response (1)
opcode: MAP (0x01)
Result Code: CANNOT_PROVIDE_EXTERNAL (0x11)
Lifetime: 1560 sec
Epoch Time: 1300
```

Figure 24: MAP response (PREFER_FAILURE Option: Requested Port is not Honored)

2.11. Negative Impact of PREFER_FAILURE Option

The presence of PREFER_FAILURE option in a request may have negative impact on an application which does not require it. Figure 25 shows two examples:

1. With PREFER_FAILURE option: several round trips are needed for the client to retrieve the requested mapping.
2. Without PREFER_FAILURE option: the client retrieves a mapping without any extra delay.

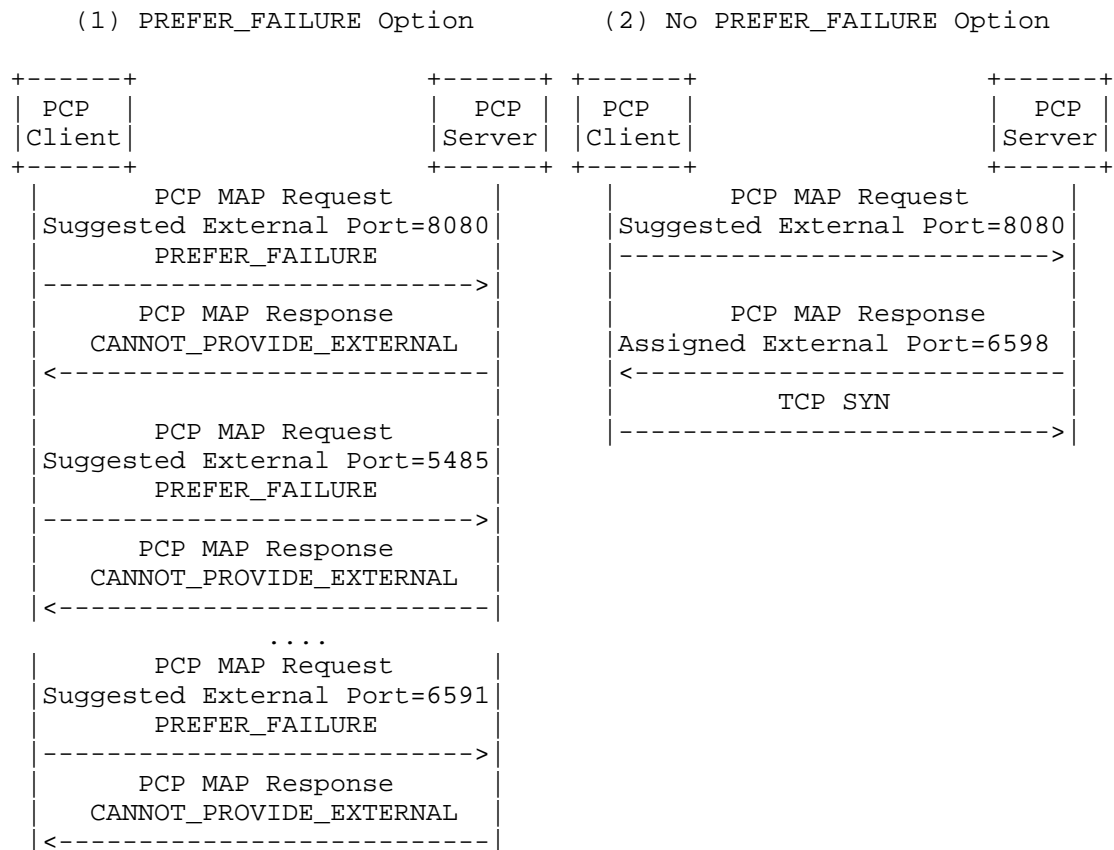


Figure 25: Negative Impact of the mis-usage of PREFER_FAILURE option

2.12. Existing Implicit Mapping

This example illustrates the content of exchanged PCP messages when the PCP client requests a mapping which matches an existing implicit dynamic mapping (see Figure 26). In this example, the PCP-Controlled device assigns 10000 as external port number when translating the packet from the client having with source port set to 1234.

This behavior is specified in Section 11.3 of [RFC6887].

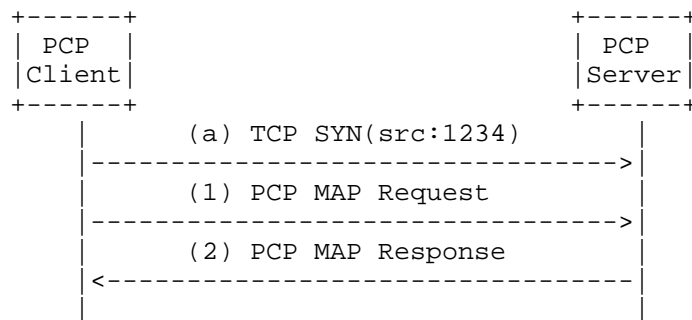


Figure 26: Example of creating a mapping

In reference to Figure 1, the content of exchanged PCP messages is as follows:

```

Version: 2
R bit: Request (0)
opcode: MAP (0x01)
Requested Lifetime: 36000 sec
PCP client's IP Address: ::ffff:198.51.100.1
MAP Request:
  Mapping Nonce: 15685
  Protocol: TCP (0x06)
  Internal Port: 1234
  Suggested External Port: 3938
  Suggested External IP Address: ::ffff:0.0.0.0
  
```

Figure 27: MAP request (Existing Implicit Mapping)

```

Version: 2
R bit: Response (1)
opcode: MAP (0x01)
Result Code: 0
Lifetime: 20000 sec
Epoch Time: 1250
MAP Response:
  Mapping Nonce: 15685
  Protocol: TCP (0x06)
  Internal Port: 1234
  Assigned External Port: 10000
  Assigned External IP Address: ::ffff:192.0.2.1
  
```

Figure 28: MAP response (Existing Implicit Mapping)

2.13. Shortening a Mapping Lifetime in the Presence of Client-Originated Traffic

Figure 29 shows an example illustrating the impact of requesting the deletion of a mapping in the presence of traffic originated from the client. In this example, the PCP server does not remove the requested mapping immediately; the returned lifetime is set to the remaining lifetime.

This behavior is specified in Section 15 of [RFC6887].

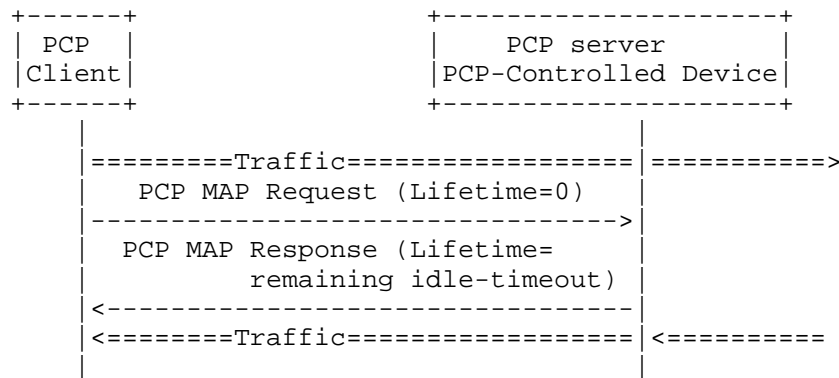


Figure 29: Shortening a Mapping Lifetime in the Presence of Client-Originated Traffic

2.14. Create a Mapping for All Incoming Traffic of a Given Protocol

This example illustrates the content of the PCP MAP request to create a mapping for all incoming traffic of a given protocol (UDP is used in this example).

```
Version: 2
R bit: Request (0)
opcode: MAP (0x01)
Requested Lifetime: 36000 sec
PCP client's IP Address: ::ffff:198.51.100.1
MAP Request:
  Mapping Nonce: 15685
  Protocol: UDP (17)
  Internal Port: 0
  Suggested External Port: 0
  Suggested External IP Address: ::ffff:0.0.0.0
```

Figure 30: MAP request (Create a mapping for all incoming traffic of a given protocol)

The PCP server may honor the request or reject it by sending UNSUPP_PROTOCOL (0x09) error.

2.15. Create a Mapping for All Protocols

This example illustrates the content of the PCP MAP request to create a mapping for the traffic of all protocols.

```
Version: 2
R bit: Request (0)
opcode: MAP (0x01)
Requested Lifetime: 36000 sec
PCP client's IP Address: ::ffff:198.51.100.1
MAP Request:
  Mapping Nonce: 15685
  Protocol: ANY (0)
  Internal Port: 0
  Suggested External Port: 0
  Suggested External IP Address: ::ffff:0.0.0.0
```

Figure 31: MAP request (Create a mapping for all protocols)

The PCP server may honor the request or reject it by sending UNSUPP_PROTOCOL (0x09) error.

2.16. Malformed Request

This flow shows an example of the content of PCP messages that will be exchanged when a malformed request is received by the PCP server. In this example, the Protocol field is set to null.


```
Version: 2
R bit: Request (0)
opcode: MAP (0x01)
Requested Lifetime: 36000 sec
PCP client's IP Address: ::ffff:198.51.100.1
MAP Request:
  Mapping Nonce: 45698
  Protocol: ANY (0)
  Internal Port: 5698
  Suggested External Port: 3938
  Suggested External IP Address: ::ffff:0.0.0.0
Option Code: PREFER_FAILURE (0x02) Option Length: 0 bytes Data: (NULL)
```

Figure 32: MAP request (Malformed Request)

```
Version: 2
R bit: Response (1)
opcode: MAP (0x01)
Result Code: MALFORMED_REQUEST (0x02)
Lifetime: 0 sec
Epoch Time: 1300
```

Figure 33: MAP response (Malformed Request)

2.17. Exceeded Port Quota

This flow shows an example of the content of PCP messages that will be exchanged when a per-user quota is reached. A short lifetime is returned so that the client may retry and see if the request can be honored because another state has been removed.

```
Version: 2
R bit: Request (0)
opcode: MAP (0x01)
Requested Lifetime: 36000 sec
PCP client's IP Address: ::ffff:198.51.100.1
MAP Request:
  Mapping Nonce: 45698
  Protocol: UDP (17)
  Internal Port: 8695
  Suggested External Port: 3938
  Suggested External IP Address: ::ffff:0.0.0.0
Option Code: PREFER_FAILURE (0x02) Option Length: 0 bytes Data: (NULL)
```

Figure 34: MAP request (Exceeded Port Quota)

```
Version: 2
R bit: Response (1)
opcode: MAP (0x01)
Result Code: USER_EX_QUOTA (10)
Lifetime: 300 sec
Epoch Time: 1300
```

Figure 35: MAP response (Exceeded Port Quota)

2.18. Unsupported Address Family

This flow shows an example of the content of PCP messages that will be exchanged when the requested external address family is not supported by the PCP server. In this example, IPv6 is indicated as the requested AF. The PCP server answers with an UNSUPP_FAMILY (14) error as defined in [I-D.cheshire-pcp-unsupp-family].

```
Version: 2
R bit: Request (0)
opcode: MAP (0x01)
Requested Lifetime: 36000 sec
PCP client's IP Address: ::ffff:198.51.100.1
MAP Request:
  Mapping Nonce: 45698
  Protocol: UDP (17)
  Internal Port: 8695
  Suggested External Port: 3938
  Suggested External IP Address: ::
```

Figure 36: MAP request (Unsupported Address Family)

```
Version: 2
R bit: Response (1)
opcode: MAP (0x01)
Result Code: UNSUPP_FAMILY (14)
Lifetime: 0 sec
Epoch Time: 1300
```

Figure 37: MAP response (Unsupported Address Family)

2.19. Unsupported Protocol

This flow shows an example of the content of PCP messages that will be exchanged when the requested port is not supported by the PCP server. In this example, SCTP is indicated as the requested protocol.

```
Version: 2
R bit: Request (0)
opcode: MAP (0x01)
Requested Lifetime: 36000 sec
PCP client's IP Address: ::ffff:198.51.100.1
MAP Request:
  Mapping Nonce: 45698
  Protocol: SCTP (132)
  Internal Port: 8695
  Suggested External Port: 3938
  Suggested External IP Address: ::ffff:0.0.0.0
```

Figure 38: MAP request (Unsupported Protocol)

```
Version: 2
R bit: Response (1)
opcode: MAP (0x01)
Result Code: UNSUPP_PROTOCOL (9)
Lifetime: 0 sec
Epoch Time: 1300
```

Figure 39: MAP response (Unsupported Protocol)

2.20. Unsolicited MAP Response

Suppose the client has instructed a UDP mapping for port 3938 (assigned external port is 15000 and assigned external IPv4 address is: 192.0.2.1). Upon a change of a state: e.g., change of the external IP Address, the PCP server issues an unsolicited MAP response. The content of the MAP response sent by the PCP server is shown below. The PCP client is now aware of the new assigned external IP address.

```
Version: 2
R bit: Response (1)
opcode: MAP (0x01)
Result Code: 0
Lifetime: 20000 sec
Epoch Time: 1250
MAP Response:
  Mapping Nonce: 15685
  Protocol: TCP (0x06)
  Internal Port: 1234
  Assigned External Port: 10000
  Assigned External IP Address: ::ffff:192.0.2.2
```

Figure 40: Unsolicited MAP Response

2.21. Mapping Repair

An example of mapping repair is shown in Figure 41.

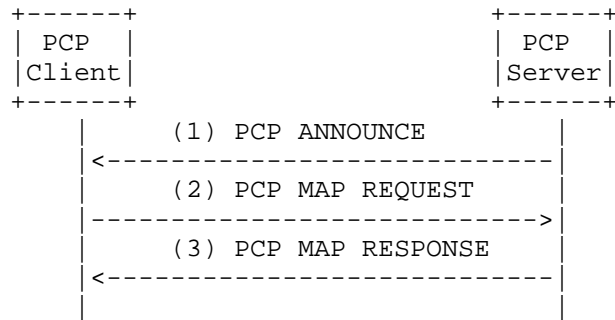


Figure 41: Flow Example of a PING/PONG exchange: Check the availability of the PCP Server

```

Version: 2
R bit: Response (1)
opcode: ANNOUNCE (0x00)
Result Code: 0
Lifetime: 0 sec
Epoch Time: 0
  
```

Figure 42: Unsolicited ANNOUNCE

```

Version: 2
R bit: Request (0)
opcode: MAP (0x01)
Requested Lifetime: 36000 sec
PCP client's IP Address: ::ffff:198.51.100.1
MAP Request:
  Mapping Nonce: 15685
  Protocol: UDP (17)
  Internal Port: 11000
  Assigned External Port: 15200
  Assigned External IP Address: ::ffff:192.0.2.1
  
```

Figure 43: MAP request (Mapping Repair)

```
Version: 2
R bit: Response (1)
opcode: MAP (0x01)
Result Code: 0
Lifetime: 20000 sec
Epoch Time: 10
MAP Response:
  Mapping Nonce: 15685
  Protocol: UDP (17)
  Internal Port: 11000
  Assigned External Port: 15200
  Assigned External IP Address: ::ffff:192.0.2.1
```

Figure 44: MAP response (Mapping Repair)

3. NAT Detect Example

Let us suppose a PCP-unaware NAT is located between the PCP server and the PCP client. An example of PCP MAP request issued by the PCP client is shown below.

```
Version: 2
R bit: Request (0)
opcode: MAP (0x01)
Requested Lifetime: 36000 sec
PCP client's IP Address: ::ffff:198.51.100.1
MAP Request:
  Mapping Nonce: 15685
  Protocol: UDP (17)
  Internal Port: 11000
  Assigned External Port: 15200
  Assigned External IP Address: ::ffff:0.0.0.0
```

Figure 45: MAP request (NAT Detect)

This message will be translated by the PCP-unaware NAT. The source IP address if the resulting message will be another address than 198.51.100.1. Upon receipt of this message, the PCP server compares the source IP address and the content of PCP client's IP Address field. Because the two addresses are not equal, the PCP server concludes there is PCP-unaware device in the path. As a result, the PCP server will issue the following error message:

```
Version: 2
R bit: Response (1)
opcode: MAP (0x01)
Result Code: ADDRESS_MISMATCH (12)
Lifetime: 0 sec
Epoch Time: 36000
```

Figure 46: MAP Response (NAT Detect)

This behavior is specified in Section 8.2 of [RFC6887].

4. Retrieve the External IP Address

In order to retrieve the IP address used on the external side of the PCP-controlled device, the PCP client sends a short-lived mapping (e.g., Discard service (TCP/9 or UDP/9) or other port). The returned IP address can be displayed by any application requiring such information.

```
Version: 2
R bit: Request (0)
opcode: MAP (0x01)
Requested Lifetime: 5 sec
PCP client's IP Address: ::ffff:198.51.100.1
MAP Request:
  Mapping Nonce: 15685
  Protocol: UDP (17)
  Internal Port: 9
  Suggested External Port: 9
  Suggested External IP Address: ::ffff:0.0.0.0
```

Figure 47: MAP request (Retrieve the External IP Address)

```
Version: 2
R bit: Response (1)
opcode: MAP (0x01)
Result Code: 0
Lifetime: 60 sec
Epoch Time: 1250
MAP Response:
  Mapping Nonce: 15685
  Protocol: UDP (17)
  Internal Port: 9
  Suggested External Port: 9
  Assigned External IP Address: ::ffff:192.0.2.1
```

Figure 48: MAP Response (Retrieve the External IP Address)

This behavior is specified in Section 11.6 of [RFC6887].

5. THIRD_PARTY Examples

These examples follow the behavior specified in Section 13.1 of [RFC6887].

5.1. THIRD_PARTY Enabled at the Server Side

The following messages are exchanged when the THIRD_PARTY option is enabled in the PCP server side. In this example the PCP client creates a mapping for the host assigned with 198.51.100.2.

```
Version: 2
R bit: Request (0)
opcode: MAP (0x01)
Requested Lifetime: 36000 sec
PCP client's IP Address: ::ffff:198.51.100.1
MAP Request:
  Mapping Nonce: 16584
  Protocol: UDP (17)
  Internal Port: 8080
  Suggested External Port: 8080
  Suggested External IP Address: ::ffff:0.0.0.0
Option Code: THIRD_PARTY (0x01) Option Length: 16 bytes Data:
  ::ffff:198.51.100.2
```

Figure 49: MAP request with THIRD_PARTY

```
Version: 2
R bit: Response (1)
opcode: MAP (0x01)
Result Code: 0
Lifetime: 20000 sec
Epoch Time: 1250
MAP Response:
  Mapping Nonce: 16584
  Protocol: UDP (17)
  Internal Port: 8080
  Assigned External Port: 15000
  Assigned External IP Address: ::ffff:161.105.194.14
Option Code: THIRD_PARTY (0x01) Option Length: 16 bytes Data:
  ::ffff:198.51.100.2
```

Figure 50: MAP Response with THIRD_PARTY

5.2. THIRD_PARTY Disabled at the Server Side

The following messages are exchanged when the THIRD_PARTY option is disabled in the PCP server side. In this example the PCP client tries to create a mapping for the host assigned with 198.51.100.2.

```
Version: 2
R bit: Request (0)
opcode: MAP (0x01)
Requested Lifetime: 36000 sec
PCP client's IP Address: ::ffff:198.51.100.1
MAP Request:
  Mapping Nonce: 16584
  Protocol: UDP (17)
  Internal Port: 8080
  Suggested External Port: 8080
  Suggested External IP Address: ::ffff:0.0.0.0
Option Code: THIRD_PARTY (0x01) Option Length: 16 bytes Data:
  ::ffff:198.51.100.2
```

Figure 51: MAP request with THIRD_PARTY

```
Version: 2
R bit: Response (1)
opcode: MAP (0x01)
Result Code: UNSUPP_OPTION (0x05)
Lifetime: 0 sec
Epoch Time: 1562
```

Figure 52: MAP Response with THIRD_PARTY

5.3. Malformed Request

In this example the PCP client inserts a THIRD_PARTY option which include the IP address of the PCP client.


```
Version: 2
R bit: Request (0)
opcode: MAP (0x01)
Requested Lifetime: 36000 sec
PCP client's IP Address: ::ffff:198.51.100.1
MAP Request:
  Mapping Nonce: 16584
  Protocol: UDP (17)
  Internal Port: 8080
  Suggested External Port: 8080
  Suggested External IP Address: ::ffff:0.0.0.0
Option Code: THIRD_PARTY (0x01) Option Length: 16 bytes Data:
  ::ffff:198.51.100.1
```

Figure 53: MAP request with THIRD_PARTY

```
Version: 2
R bit: Response (1)
opcode: MAP (0x01)
Result Code: MALFORMED_REQUEST (0x03)
Lifetime: 0 sec
Epoch Time: 1562
```

Figure 54: MAP Response with THIRD_PARTY

6. MAP with FILTER Examples

These examples follow the behavior specified in Section 13.3 of [RFC6887].

6.1. Basic Filter Usage

This example illustrates the content of exchanged PCP messages when the PCP client wants to receive traffic only from 192.0.2.200:5968. In reference to Figure 1, the content of exchanged PCP messages is as follows:

```
Version: 2
R bit: Request (0)
opcode: MAP (0x01)
Requested Lifetime: 36000 sec
PCP client's IP Address: ::ffff:198.51.100.1
MAP Request:
  Mapping Nonce: 15685
  Protocol: UDP (17)
  Internal Port: 3938
  Suggested External Port: 3938
  Suggested External IP Address: ::ffff:0.0.0.0
Option Code: FILTER (0x03) Option Length: 20 bytes Data:
  Prefix Length: 128
  Remote Peer Port: 5968
  Remote Peer IP Address: ::ffff:192.0.2.200
```

Figure 55: MAP request (Basic Filter Usage)

```
Version: 2
R bit: Response (1)
opcode: MAP (0x01)
Result Code: 0
Lifetime: 20000 sec
Epoch Time: 1250
MAP Response:
  Mapping Nonce: 15685
  Protocol: UDP (17)
  Internal Port: 3938
  Assigned External Port: 3938
  Assigned External IP Address: ::ffff:192.0.2.1
Option Code: FILTER (0x03) Option Length: 20 bytes Data:
  Prefix Length: 128
  Remote Peer Port: 5968
  Remote Peer IP Address: ::ffff:192.0.2.200
```

Figure 56: MAP Response (Basic Filter Usage)

6.2. Remove All Filters

This example illustrates the content of exchanged PCP messages when the PCP client wants to remove all filters. In reference to Figure 1, the content of exchanged PCP messages is as follows:

```
Version: 2
R bit: Request (0)
opcode: MAP (0x01)
Requested Lifetime: 36000 sec
PCP client's IP Address: ::ffff:198.51.100.1
MAP Request:
  Mapping Nonce: 15685
  Protocol: UDP (17)
  Internal Port: 3938
  Suggested External Port: 3938
  Suggested External IP Address: ::ffff:0.0.0.0
Option Code: FILTER (0x03) Option Length: 20 bytes Data:
  Prefix Length: 0
  Remote Peer Port: 0
  Remote Peer IP Address: ::ffff:0:0
```

Figure 57: MAP request (Remove All Filters)

```
Version: 2
R bit: Response (1)
opcode: MAP (0x01)
Result Code: 0
Lifetime: 20000 sec
Epoch Time: 1250
MAP Response:
  Mapping Nonce: 15685
  Protocol: UDP (17)
  Internal Port: 3938
  Assigned External Port: 3938
  Assigned External IP Address: ::ffff:192.0.2.1
Option Code: FILTER (0x03) Option Length: 20 bytes Data:
  Prefix Length: 0
  Remote Peer Port: 0
  Remote Peer IP Address: ::ffff:0:0
```

Figure 58: MAP response (Remove All Filters)

6.3. Change an Existing Filter

This example illustrates the content of exchanged PCP messages when the PCP client wants to change an existing filter. In reference to Figure 1, the content of exchanged PCP messages is as follows:

```

Version: 2
R bit: Request (0)
opcode: MAP (0x01)
Requested Lifetime: 36000 sec
PCP client's IP Address: ::ffff:198.51.100.1
MAP Request:
  Mapping Nonce: 15685
  Protocol: UDP (17)
  Internal Port: 3938
  Suggested External Port: 3938
  Suggested External IP Address: ::ffff:0.0.0.0
Option Code: FILTER (0x03) Option Length: 20 bytes Data:
  Prefix Length: 0
  Remote Peer Port: 0
  Remote Peer IP Address: ::ffff:0:0
Option Code: FILTER (0x03) Option Length: 20 bytes Data:
  Prefix Length: 128
  Remote Peer Port: 5968
  Remote Peer IP Address: ::ffff:192.0.2.201

```

Figure 59: MAP request (Change an Existing Filter)

7. Assess the Reachability of the PCP Server

In this example, the PCP client issues a PCP ANNOUNCE request to a PCP server. Once received by the PCP server, since it is configured to reply to such request, it sends back a PCP ANNOUNCE response. This procedure can be used to retrieve the Epoch time.

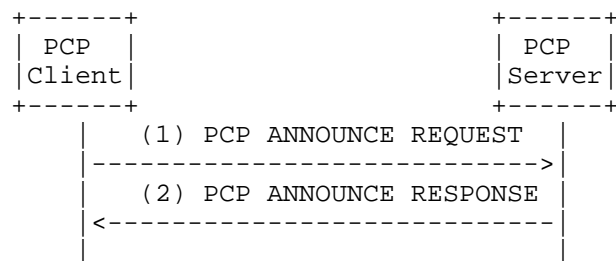


Figure 60: Flow example of a PING/PONG exchange: Check the availability of the PCP server

```

Version: 2
R bit: Request (0)
opcode: ANNOUNCE (0x00)
Requested Lifetime: 0 sec
PCP client's IP Address: ::ffff:198.51.100.1

```

Figure 61: ANNOUNCE request (Assess the Reachability of the PCP Server)

```

Version: 2
R bit: Response (1)
opcode: ANNOUNCE (0x00)
Result Code: 0
Lifetime: 0 sec
Epoch Time: 3600

```

Figure 62: ANNOUNCE response (Assess the Reachability of the PCP Server)

8. PEER Operations

The following figure illustrates the messages which are exchanged when PEER opcode is used:

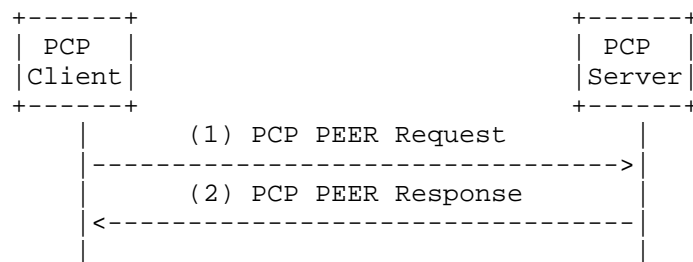


Figure 63: Typical usage of PEER message

Examples listed below follow the behavior specified in Section 12.2 and Section 12.3 of [RFC6887].

8.1. No Mapping Exists for the Internal Port Number

In reference to Figure 63, the content of exchanged PEER messages when no mapping is maintained by the PCP server for the indicated external port number:

```
Version: 2
R bit: Request (0)
opcode: PEER (0x02)
Requested Lifetime: 36000 sec
PCP client's IP Address: ::ffff:198.51.100.1
PEER Request:
  Mapping Nonce: 15685
  Protocol: UDP (17)
  Internal Port: 3938
  Suggested External Port: 3938
  Suggested External IP Address: ::ffff:0.0.0.0
  Remote Peer Port: 12456
  Remote IP Address: ::ffff:198.51.100.2
```

Figure 64: PEER request (No Mapping Exists for the Internal Port Number)

```
Version: 2
R bit: Response (1)
opcode: PEER (0x02)
Result Code: 0
Lifetime: 20000 sec
Epoch Time: 1250
PEER Response:
  Mapping Nonce: 15685
  Protocol: UDP (17)
  Internal Port: 3938
  Assigned External Port: 3938
  Assigned External IP Address: ::ffff:192.0.2.1
  Remote Peer Port: 12456
  Remote IP Address: ::ffff:198.51.100.2
```

Figure 65: PEER response (No Mapping Exists for the Internal Port Number)

8.2. A Mapping Exists for the External Port Number

In reference to Figure 63, the content of exchanged PEER messages when a mapping is maintained by the PCP server for the indicated external port number:

```
Version: 2
R bit: Request (0)
opcode: PEER (0x02)
Requested Lifetime: 36000 sec
PCP client's IP Address: ::ffff:198.51.100.1
PEER Request:
  Mapping Nonce: 15685
  Protocol: UDP (17)
  Internal Port: 3938
  Suggested External Port: 3938
  Suggested External IP Address: ::ffff:0.0.0.0
  Remote Peer Port: 12456
  Remote IP Address: ::ffff:198.51.100.2
```

Figure 66: PEER request (A Mapping Exists for the External Port Number)

```
Version: 2
R bit: Response (1)
opcode: PEER (0x02)
Result Code: CANNOT_PROVIDE_EXTERNAL
Lifetime: 0 sec
Epoch Time: 36000
```

Figure 67: PEER response (A Mapping Exists for the External Port Number)

8.3. External IP Address Cannot be Honored

In reference to Figure 63, the content of exchanged PEER messages when the suggested external IP address does not match an existing mapping is shown below:

```
Version: 2
R bit: Request (0)
opcode: PEER (0x02)
Requested Lifetime: 36000 sec
PCP client's IP Address: ::ffff:198.51.100.1
PEER Request:
  Mapping Nonce: 15685
  Protocol: UDP (17)
  Internal Port: 3938
  Suggested External Port: 3938
  Suggested External IP Address: ::ffff:192.0.2.5
  Remote Peer Port: 12456
  Remote IP Address: ::ffff:198.51.100.2
```

Figure 68: PEER request (External IP Address Cannot be Honored)

```

Version: 2
R bit: Response (1)
opcode: PEER (0x02)
Result Code: CANNOT_PROVIDE_EXTERNAL
Lifetime: 0 sec
Epoch Time: 36000

```

Figure 69: PEER response (External IP Address Cannot be Honored)

8.4. Extend the Lifetime

In reference to Figure 70, the content of exchanged PEER messages to extend the lifetime of a mapping.

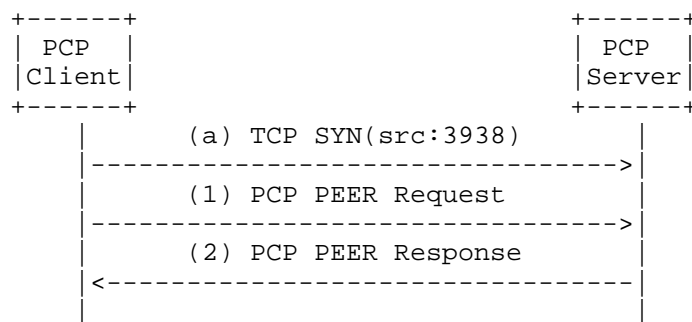


Figure 70: Example of creating a mapping

```

Version: 2
R bit: Request (0)
opcode: PEER (0x02)
Requested Lifetime: 36000 sec
PCP client's IP Address: ::ffff:198.51.100.1
PEER Request:
  Mapping Nonce: 15685
  Protocol: TCP (6)
  Internal Port: 3938
  Suggested External Port: 0
  Suggested External IP Address: ::ffff:0.0.0.0
  Remote Peer Port: 12456
  Remote IP Address: ::ffff:198.51.100.2

```

Figure 71: PEER request (Extend the Lifetime)


```
Version: 2
R bit: Response (1)
opcode: PEER (0x02)
Result Code: 0
Lifetime: 20000 sec
Epoch Time: 1250
PEER Response:
  Mapping Nonce: 15685
  Protocol: TCP (6)
  Internal Port: 3938
  Assigned External Port: 11000
  Assigned External IP Address: ::ffff:192.0.2.1
  Remote Peer Port: 12456
  Remote IP Address: ::ffff:198.51.100.2
```

Figure 72: PEER response (Extend the Lifetime)

8.5. Learn the Lifetime of a Mapping

In reference to Figure 70, the content of exchanged PEER messages to learn the lifetime of a mapping is shown below:

```
Version: 2
R bit: Request (0)
opcode: PEER (0x02)
Requested Lifetime: 5 sec
PCP client's IP Address: ::ffff:198.51.100.1
PEER Request:
  Mapping Nonce: 15685
  Protocol: TCP (6)
  Internal Port: 3938
  Suggested External Port: 0
  Suggested External IP Address: ::ffff:0.0.0.0
  Remote Peer Port: 12456
  Remote IP Address: ::ffff:198.51.100.2
```

Figure 73: PEER request (Learn the Lifetime of a Mapping)

```
Version: 2
R bit: Response (1)
opcode: PEER (0x02)
Result Code: 0
Lifetime: 20000 sec
Epoch Time: 1250
PEER Response:
  Mapping Nonce: 15685
  Protocol: TCP (6)
  Internal Port: 3938
  Assigned External Port: 11000
  Assigned External IP Address: ::ffff:192.0.2.1
  Remote Peer Port: 12456
  Remote IP Address: ::ffff:198.51.100.2
```

Figure 74: PEER response (Learn the Lifetime of a Mapping)

9. Version Negotiation

The following exchange occurs between a PCP client that supports PCP version 1 and the PCP server that supports PCP version 2.

```
Version: 1
R bit: Request (0)
opcode: MAP (0x01)
Requested Lifetime: 36000 sec
PCP client's IP Address: ::ffff:198.51.100.1
MAP Request:
  Protocol: UDP (17)
  Internal Port: 3938
  Suggested External Port: 3938
  Suggested External IP Address: ::ffff:0.0.0.0
```

Figure 75: MAP request with Version 1

```
Version: 2
R bit: Response (1)
opcode: MAP (0x01)
Result Code: UNSUPP_VERSION (1)
Lifetime: 0 sec
Epoch Time: 3600
```

Figure 76: MAP response (Unsupported Version)

Version negotiation is specified in Section 9 of [RFC6887].

10. Security Considerations

PCP security considerations are discussed in [RFC6887].

11. IANA Considerations

This document has no IANA actions.

12. Acknowledgements

Many thanks to C. Jacquenet and D. Wing for the comments.

13. References

13.1. Normative References

- [RFC3849] Huston, G., Lord, A., and P. Smith, "IPv6 Address Prefix Reserved for Documentation", RFC 3849, July 2004.
- [RFC5737] Arkko, J., Cotton, M., and L. Vegoda, "IPv4 Address Blocks Reserved for Documentation", RFC 5737, January 2010.
- [RFC6887] Wing, D., Cheshire, S., Boucadair, M., Penno, R., and P. Selkirk, "Port Control Protocol (PCP)", RFC 6887, April 2013.
- [RFC7220] Boucadair, M., Penno, R., and D. Wing, "Description Option for the Port Control Protocol (PCP)", RFC 7220, May 2014.
- [RFC7225] Boucadair, M., "Discovering NAT64 IPv6 Prefixes Using the Port Control Protocol (PCP)", RFC 7225, May 2014.

13.2. Informative References

- [I-D.cheshire-pcp-unsupp-family]
Cheshire, S. and S. Perreault, "Updates to the PCP Specification", draft-cheshire-pcp-unsupp-family-06 (work in progress), October 2013.

Author's Address

Mohamed Boucadair
France Telecom
Rennes 35000
France

Email: mohamed.boucadair@orange.com

PCP Working Group
Internet-Draft
Intended status: Standards Track
Expires: June 1, 2013

M. Boucadair
France Telecom
T. Reddy
P. Patil
D. Wing
Cisco
November 28, 2012

Using PCP to Reveal a Host behind NAT
draft-boucadair-pcp-nat-reveal-00

Abstract

This document describes how to use PCP to retrieve the identify of a host behind a NAT. Two use cases are discussed and the PCP applicability is analyzed. This document extends PCP with a new OpCode: QUERY.

The proposed mechanism is valid for all NAT flavors including NAT44, NAT64 or NPTv6.

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 June 1, 2013.

Copyright Notice

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

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

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

Table of Contents

1. Introduction	3
2. Requirements Language and Terminology	3
3. Problem Space	3
3.1. Policy and Charging Control Architecture	3
3.2. NAT between the PCEF and AF	4
3.3. NAT before PCEF	6
4. PCP Applicability	7
4.1. NAT between the PCEF and AF	7
4.2. NAT before PCEF	8
5. QUERY OpCode	9
5.1. QUERY Request Format	10
5.2. QUERY Response Format	11
5.3. Generating a QUERY Request	12
5.4. Processing a QUERY Request	12
5.5. Processing a QUERY Response	13
6. Applicability Scope of QUERY OpCode	13
7. IANA Considerations	14
8. Security Considerations	14
9. References	14
9.1. Normative References	14
9.2. Informative References	15
Authors' Addresses	16

1. Introduction

As reported in [RFC6269], several issues are encountered when an IP address is shared among several subscribers. These issues are encountered in various deployment contexts: e.g., Carrier Grade NAT (CGN), application proxies or A+P [RFC6346].

This document extends Port Control Protocol [I-D.ietf-pcp-base] to identify a host among those sharing the same IP address in certain scenarios.

The proposed technique can be used independently or combined with the host identifier, denoted as HOST_ID defined in [I-D.ietf-intarea-nat-reveal-analysis].

Additional scenarios requiring host identification are listed in [I-D.boucadair-intarea-host-identifier-scenarios].

2. Requirements Language and 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].

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

3. Problem Space

3.1. Policy and Charging Control Architecture

Figure 1 depicts a reference architecture of a mobile network [RFC6342].

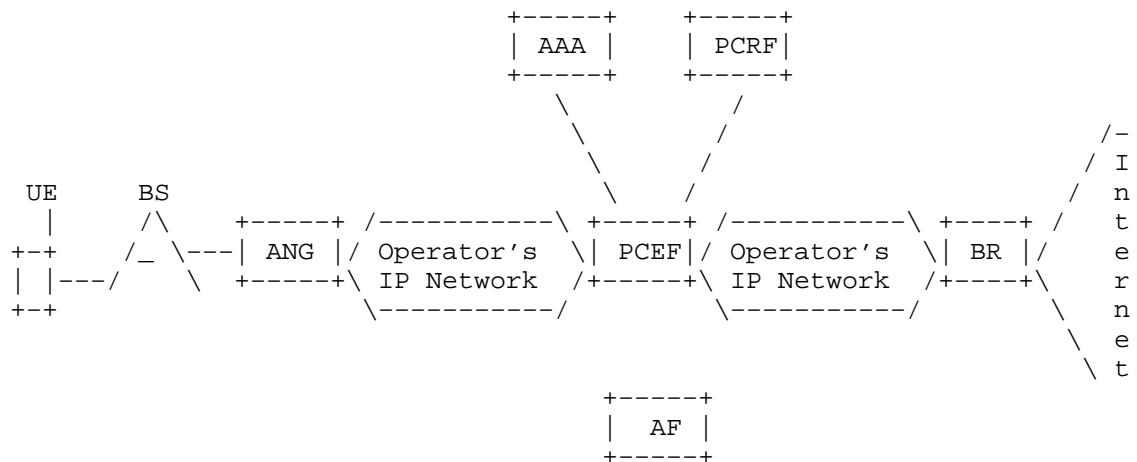


Figure 1: Mobile Network Architecture

The main involved functional elements are:

- o UE (User Equipment) is a mobile node.
- o Policy and Charging Rule Function (PCRF) which is responsible for determining which policy and charging control rules are to be applied [TS.23203].
- o Policy and Charging Enforcement Function (PCEF) which performs policy enforcement (e.g., Quality of Service (QoS)) and flow-based charging [TS.23203].
- o Application Function (AF) is an element offering applications that require dynamic policy and/or charging control [TS.23203].
- o Access Network Gateway (ANG), Base Station (BS) and Border Router (BR) are defined in [RFC6342].

Section 3.2 and Section 3.3 explain the encountered problems to identify individual UEs when a NAT is involved in the data path. The use of PCP to solve those problems is analyzed in Section 4.

3.2. NAT between the PCEF and AF

Figure 2 shows a scenario where a NAT function is located between the PCEF and AF.

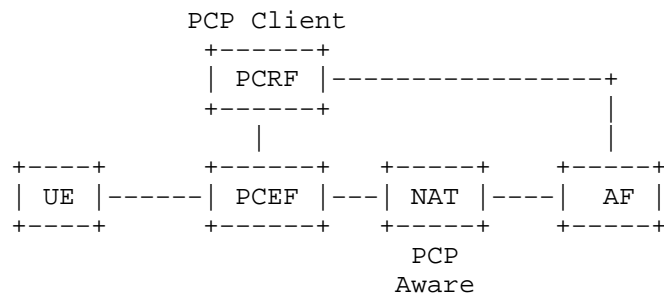


Figure 2: NAT between PCEF and AF

The main issue in this case is that PCEF, PCRF and AF all receive information bound to the same UE but cannot correlate between the piece of data visible for each entity. Concretely,

- o PCEF is aware of the IMSI (International Mobile Subscriber Identity) and an internal IP address assigned to the UE.
- o AF receives an external IP address and port as assigned by the NAT function.
- o PCRF is not able to correlate between the external IP address/port assigned by the NAT and the internal IP address and IMSI of the UE. For instance, the offered QoS on internal IPv4 address and the (shared) external IPv4 address may need to be correlated for accounting purposes.
- o The IP address seen by the AF is shared among multiple UEs using NAT, the PCRF needs to be able to inspect the NAT binding to disambiguate among the individual UEs. AF will not be able to treat UE traffic based on policy provided by PCRF.

This scenario can be generalized as follows (Figure 3):

- o Policy Enforcement Point (PEP, [RFC2753])
- o Policy Decision Point (PDP, [RFC2753])

Figure 3: NAT between PEP and Server

Figure 4: NAT before PCEF

This scenario can be generalized as follows (Figure 5):

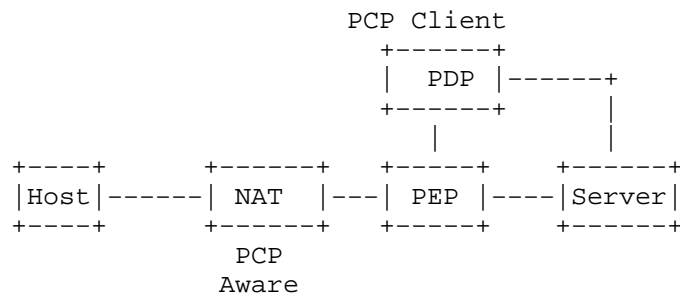


Figure 5: NAT before PEP

4. PCP Applicability

This section discusses how PCP can be used to solve the problems described in Section 3.2 and Section 3.3.

4.1. NAT between the PCEF and AF

A solution to solve the problem discussed in Section 3.2 is to enable a PCP Server to control the NAT and enable a PCP Client in the PCRF. The updated interaction between PCRF, PCEF and AF is detailed below.

- o The PCP server controlling the NAT is configured to accept PCP requests with THIRD_PARTY Option from authorized PCP clients (i.e., PCRF).
- o PCRF is configured with the IP address of the PCP Server.
- o The PCRF is aware of the following 5-tuple of each flow {Internal IP address of UE, Internal Port, Protocol, Remote Peer IP address, Remote Port number} learnt from PCEF. PCRF is also aware of the following 5-tuple of each flow {External IP address, External Port, Protocol, Remote Peer IP address, Remote Port number} learnt from AF.
- o The PCRF generates PCP PEER request with THIRD_PARTY option which has Internal IP Address set to the UE's Internal IP address provided by the PCEF.
 - * The Internal Port, Protocol, Remote Peer Port, Remote Peer IP Address fields of the PEER request are set by the PCRF according to the 5-tuple flow information provided by PCEF.

- * Suggested External Port and Suggested External IP Address are set to zero.
 - * Requested Lifetime field is set to a very low value (see Section 12.3 of [I-D.ietf-pcp-base]).
 - o Upon the receipt of the PEER response, the PCRF compares the External IP Address and Port learnt with the 5-tuple flow information provided by the AF to correlate external IP address/port associated with the mapping and the internal IP address/port of the flow.
 - o PCRF notifies PCEF/AF to enforce relevant policies for the UE.
- 4.2. NAT before PCEF

A solution to solve the problem discussed in Section 3.3 is to extend PCP with a new OpCode called QUERY (see Section 5).

The updated interaction between PCRF, PCEF and AF is detailed below:

- o The PCP server controlling the NAT is configured to accept QUERY requests Section 5 from authorized PCP clients such as PCRF. Query requests must not be received in the Internet-facing interface but from an internal interface (e.g., dedicated management interface).
- o PCRF generates a PCP QUERY request with External IP Address, External Port, Remote Peer IP address, Remote Peer Port and Protocol fields for the flow learnt from PCEF or AF.
- o PCRF learns the internal IP address and internal port number in the QUERY response. This correlation is used by the PCRF to retrieve the UE's policy to be passed to the PCEF.

Figure 6 shows an example of the use of QUERY OpCode. In this example, an HTTP connection is initiated by the UA (192.0.2.1:33041) to an HTTP server (198.51.100.2:80). The NAT assigns 198.51.100.1/23432 as external IP Address/Port. PCRF learns Internal IP Address and Port associated with the NAT mapping using PCP QUERY request/response.

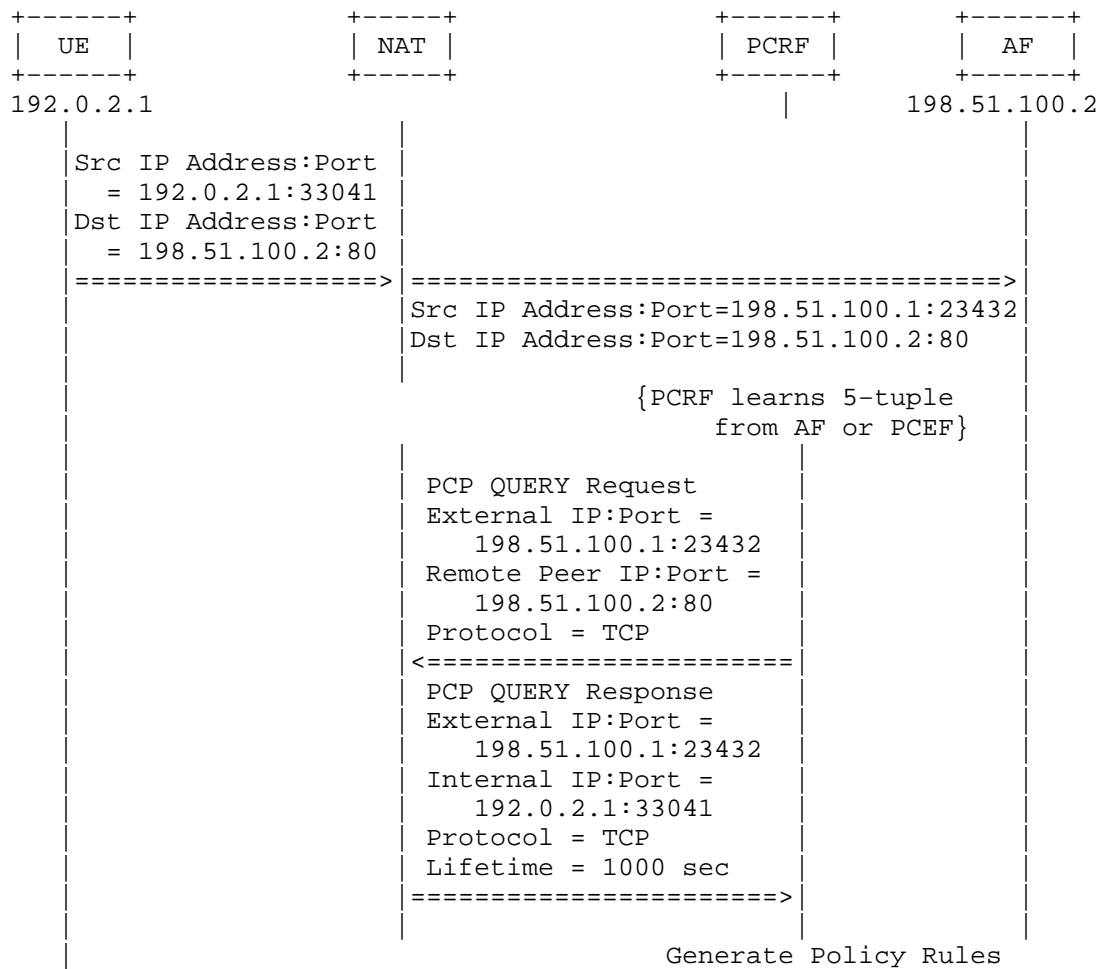


Figure 6: Usage Example

5. QUERY OpCode

This section defines a new PCP OpCode which can be used to query PCP-aware NAT to retrieve the Internal IP Address and Internal Port of a given mapping.

The PCP Server MUST provide a configuration option to allow administrators to enable/disable QUERY OpCode.

5.1. QUERY Request Format

The following diagram shows the format of the OpCode-specific information in a request for the QUERY OpCode.

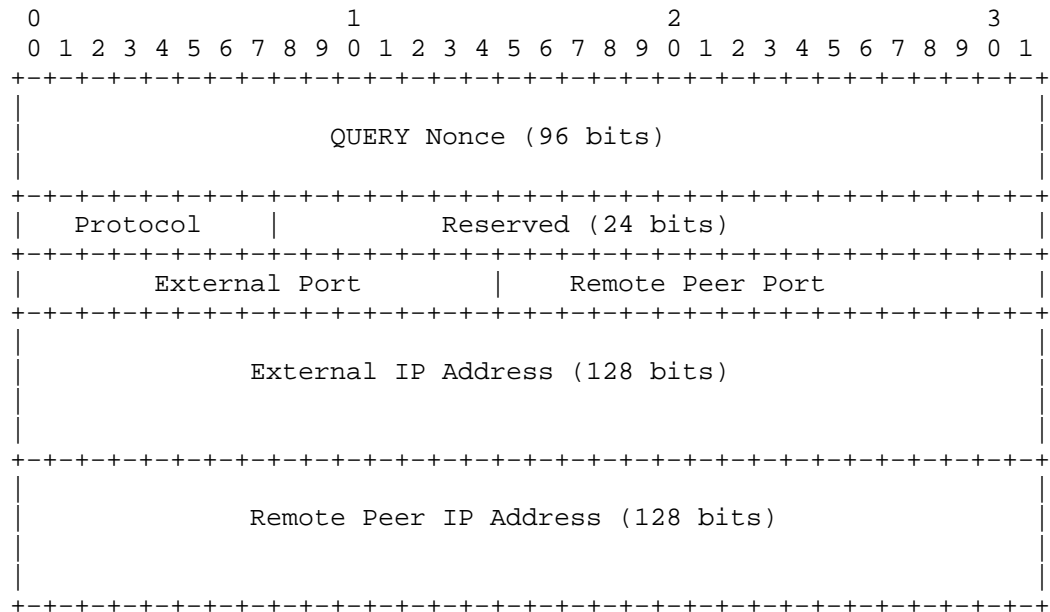


Figure 7: Query Opcode Request

These fields are described below:

Requested Lifetime (in common header): This field is positioned to 0.

Mapping Nonce: Random value chosen by the PCP client. See Section 12.2 of [I-D.ietf-pcp-base]

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.

External Port: External port allocated by NAT for the flow.
External Port MUST NOT be zero

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

External IP address: External IP address allocated by NAT for the
flow. External IP address MUST NOT be zero

Remote Peer IP address: Remote peer IP address for the flow. Remote
Peer IP address MUST NOT be zero.

5.2. QUERY Response Format

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

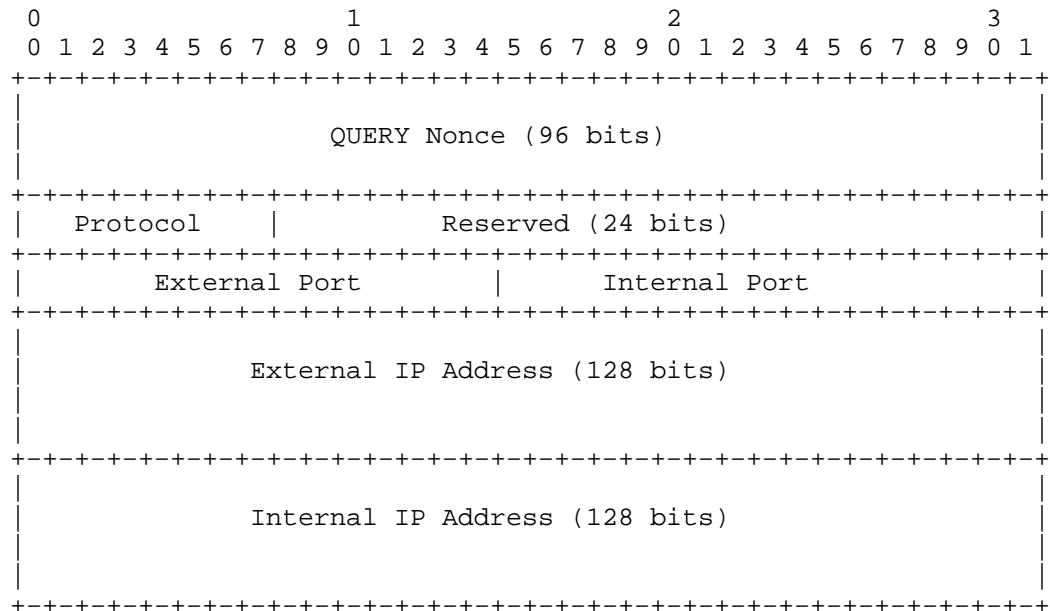


Figure 8: Query Opcode Response

These fields are described below:

Lifetime (in common header): On a success response, this indicates the lifetime for this mapping, in seconds. On an error response, this indicates that mapping does not exist.

Mapping Nonce: Copied from the request.

Protocol: Copied from the request.

Reserved: 24 reserved bits, MUST be set to 0.

External Port: Copied from the request.

External IP address: Copied from the request.

Internal Port: Internal Port as assigned by the PCP-controlled device.

Internal IP address: Internal IP address as assigned by the PCP-controlled device.

5.3. Generating a QUERY Request

This section describes the operation of a PCP client when sending requests with the QUERY OpCode.

PCP QUERY request is used by an authorized third party PCP client that is only aware of the 5-tuple {External IP address and Port, Protocol, Remote Peer IP address and Port} and needs to learn the Internal IP address and Port associated with the NAT mapping. The request MUST contain non-zero values of Protocol, External Port, Remote Peer Port, External IP address and Remote Peer IP address. The Requested Lifetime MUST be set to zero.

5.4. Processing a QUERY Request

This section describes the operation of a PCP server when processing a QUERY request.

For EIM/EIF port-mapping NAT, the processing of the QUERY request is as follows:

- o If any of the values Protocol, External Port and External IP address are equal to zero, the request is invalid and the PCP server MUST return a MALFORMED_REQUEST to the client.
- o If Protocol, External Port and External IP address do not match any existing implicit dynamic mapping, then the PCP server MUST return NONEXIST_MAP error response (also needed in

[I-D.boucadair-pcp-failure])).

- o If Protocol, External Port and External IP address match an existing implicit dynamic mapping, then the PCP server MUST build a QUERY response with the Internal IP address, Internal Port and the lifetime associated with the mapping.

For EDM port-mapping NAT, the processing of the QUERY request is as follows:

- o If any of the values Protocol, External Port, Remote Peer Port, External IP address and Remote Peer IP Address are zero, the request is invalid and PCP server MUST return a MALFORMED_REQUEST to the client.
- o If Protocol, External Port, Remote Peer Port, External IP address and Remote Peer IP address do not match any existing implicit dynamic mapping then the PCP server MUST return NONEXIST_MAP error response (also needed in [I-D.boucadair-pcp-failure])).
- o If Protocol, External Port, Remote Peer Port, External IP address and Remote Peer IP address matches an existing implicit dynamic mapping then the PCP server builds a QUERY response with the Internal IP address, Internal Port and the lifetime associated with the mapping.

PCP QUERY requests received on the Internet-facing interface MUST be silently dropped.

In DS-Lite context [RFC6333], the Internal IP address returned in the QUERY response MUST be the IPv6 address of the remote tunnel endpoint and not the internal private IPv4 address.

5.5. Processing a QUERY Response

After performing common PCP response processing by the PCP Client, the response is further matched with a previously-sent QUERY request by comparing the QUERY Nonce, External IP Address, External Port and Protocol. On a SUCCESS response, the PCP Client can use the Internal IP Address and Port in the QUERY response as needed.

6. Applicability Scope of QUERY OpCode

The PCP-Reveal solution is designed for needs within one single administrative domain (i.e., the PCP Client and PCP Server are managed by the same entity). Considerations related to the activation of the PCP-Reveal solution in an inter-domain context is

out of scope of this document.

7. IANA Considerations

Authors of this document request the following OpCode:

- o QUERY

The following error code is requested:

- o NONEXIST_MAP

8. Security Considerations

Security considerations discussed in [I-D.ietf-pcp-base] are to be taken into account. In particular, QUERY OpCode MUST NOT be implemented or used unless the network on which the PCP QUERY messages are to be sent is fully trusted. For example if Access Control Lists (ACLs) are installed on the PCP server, and the network between the PCP client and the PCP server, so those ACLs allow only communications from a trusted PCP client to the PCP server.

QUERY OpCode may be generated by non legitimate PCP Clients; the PCP Server MUST enforce some policies such as rate limit QUERY messages. QUERY requests received from non legitimate PCP Clients are silently dropped.

PCP authentication [I-D.ietf-pcp-authentication] MAY be used.

9. References

9.1. Normative References

- [I-D.ietf-pcp-base]
Wing, D., Cheshire, S., Boucadair, M., Penno, R., and P. Selkirk, "Port Control Protocol (PCP)",
draft-ietf-pcp-base-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.
- [proto_numbers]
IANA, "Protocol Numbers", 2010, <<http://www.iana.org/assignments/protocol-numbers/protocol-numbers.xml>>.

9.2. Informative References

- [I-D.boucadair-intarea-host-identifier-scenarios]
Boucadair, M., Binet, D., Durel, S., and T. Reddy,
"HOST_ID: Use Cases",
draft-boucadair-intarea-host-identifier-scenarios-01 (work
in progress), October 2012.
- [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.ietf-intarea-nat-reveal-analysis]
Boucadair, M., Touch, J., Levis, P., and R. Penno,
"Analysis of Solution Candidates to Reveal a Host
Identifier (HOST_ID) in Shared Address Deployments",
draft-ietf-intarea-nat-reveal-analysis-04 (work in
progress), August 2012.
- [I-D.ietf-pcp-authentication]
Wasserman, M., Hartman, S., and D. Zhang, "Port Control
Protocol (PCP) Authentication Mechanism",
draft-ietf-pcp-authentication-01 (work in progress),
October 2012.
- [RFC2753] Yavatkar, R., Pendarakis, D., and R. Guerin, "A Framework
for Policy-based Admission Control", RFC 2753,
January 2000.
- [RFC6269] Ford, M., Boucadair, M., Durand, A., Levis, P., and P.
Roberts, "Issues with IP Address Sharing", RFC 6269,
June 2011.
- [RFC6333] Durand, A., Droms, R., Woodyatt, J., and Y. Lee, "Dual-
Stack Lite Broadband Deployments Following IPv4
Exhaustion", RFC 6333, August 2011.
- [RFC6342] Koodli, R., "Mobile Networks Considerations for IPv6
Deployment", RFC 6342, August 2011.
- [RFC6346] Bush, R., "The Address plus Port (A+P) Approach to the
IPv4 Address Shortage", RFC 6346, August 2011.
- [TS.23203]
3GPP, "Policy and charging control architecture",
September 2012.

Authors' Addresses

Mohamed Boucadair
France Telecom
Rennes, 35000
France

Email: mohamed.boucadair@orange.com

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

Email: tiredddy@cisco.com

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

Email: praspati@cisco.com

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

Email: dwing@cisco.com

PCP working group
Internet-Draft
Intended status: Standards Track
Expires: January 15, 2014

S. Cheshire
Apple
July 14, 2013

PCP Anycast Address
draft-cheshire-pcp-anycast-02

Abstract

The Port Control Protocol Anycast Address enables PCP clients to transmit messages to their closest on-path NAT, Firewall, or other middlebox, without having to learn the IP address of that middlebox via some external channel.

Status of this Memo

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

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

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

This Internet-Draft will expire on January 15, 2014.

Copyright Notice

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

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

1. Introduction

The Port Control Protocol document [RFC6887] specifies the message formats used, but the address to which a client sends its request is either assumed to be the default router (which is appropriate in a typical single-link residential network) or has to be configured otherwise via some external mechanism, such as DHCP.

One drawback of relying on external configuration is that it creates an external dependency on another piece of network infrastructure which must be configured with the right address for PCP to work. In some environments the staff managing the DHCP servers may not be the same staff managing the NAT gateways, and in any case, constantly keeping the DHCP server address information up to date as NAT gateways are added, removed, or reconfigured, is burdensome.

Another drawback of relying on DHCP for configuration is that one of the target deployment environments for PCP -- 3GPP for mobile telephones -- does not use DHCP.

One design option that was considered for Apple's NAT gateways was to have the NAT gateway simply handle and respond to all packets addressed to UDP port 5351, regardless of the destination address in the packet. Since the device is a NAT gateway, it already examines every packet in order to rewrite port numbers, so also detecting packets addressed to UDP port 5351 is not a significant additional burden. Also, since this device is a NAT gateway which rewrites port numbers, any attempt by a client to talk *through* this first NAT gateway to create mappings in some second upstream NAT gateway is futile and pointless. Any mappings created in the second NAT gateway are useful to the client only if there are also corresponding mappings created in the first NAT gateway. Consequently, there is no case where it is useful for PCP requests to pass transparently through the first PCP-aware NAT gateway on their way to the second PCP-aware NAT gateway. In all cases, for useful connectivity to be established, the PCP request must be handled by the first NAT gateway, and then the first NAT gateway generates a corresponding new upstream request to establish a mapping in the second NAT gateway. (This process can be repeated recursively for as many times as necessary for the depth of nesting of NAT gateways; this is transparent to the client device [Recurs].)

These two issues result in the following related observations: the PCP client may not *know* what destination address to use in its PCP request packets; the PCP server doesn't *care* what destination address is in the PCP request packets.

Given that the devices neither need to know nor care what destination

address goes in the packet, all we need to do is pick one and use it. It's little more than a placeholder in the IP header. Any globally routable unicast address will do. Since this address is one that automatically routes its packet to the closest on-path device that implements the desired functionality, it is an anycast address.

In the simple case where the first-hop router is also the NAT gateway (as is common in a typical single-link residential network), sending to the PCP anycast address is equivalent to sending to the client's default router, as specified in the PCP base document [RFC6887].

In the case of a larger corporate network, where there may be several internal routed subnets and one or more border NAT gateway(s) connecting to the rest of the Internet, sending to the PCP anycast address has the interesting property that it magically finds the right border NAT gateway for that client. Since we posit that other network infrastructure does not need (and should not have) any special knowledge of PCP (or its anycast address) this means that to other non-NAT routers, the PCP anycast address will look like any other unicast destination address on the public Internet, and consequently the packet will be forwarded as for any other packet destined to the public Internet, until it reaches a NAT or firewall device that is aware of the PCP anycast address. This will result in the packet naturally arriving the NAT gateway that handles this client's outbound traffic destined to the public Internet, which is exactly the NAT gateway that the client wishes to communicate with when managing its port mappings.

2. Benefit of using a PCP Anycast Address

The benefit of using an anycast address is simplicity and reliability. In an example deployment scenario:

1. A network administrator installs a PCP-capable NAT.
2. An end user (who may be the same person) runs a PCP-enabled application. This application can implement PCP with purely user-level code -- no operating system support is required.
3. This PCP-enabled application sends its PCP request to the PCP anycast address. This packet travels through the network like any other, without any special support from DNS, DHCP, other routers, or anything else, until it reaches the PCP-capable NAT, which receives it, handles it, and sends back a reply.

Using the PCP anycast address, the only two things that need to be deployed in the network are the two things that actually use PCP: The

PCP-capable NAT, and the PCP-enabled application. Nothing else in the network needs to be changed or upgraded, and nothing needs to be configured, including the PCP client.

3. Historical Objections to Anycast

In March 2001 a draft document entitled "Analysis of DNS Server Discovery Mechanisms for IPv6" [DNSDisc] proposed using anycast to discover DNS servers, a proposal that was subsequently abandoned in later revisions of that draft document.

There are legitimate reasons why using anycast to discover DNS servers is not compelling, mainly because it requires explicit configuration of routing tables to direct those anycast packets to the desired DNS server. However, DNS server discovery is very different to NAT gateway discovery. A DNS server is something a client explicitly talks to, via IP address. The DNS server may be literally anywhere on the Internet. Various reasons make anycast an unconvincing technique for DNS server discovery:

- o DNS is a pure application-layer protocol, running over UDP.
- o On an operating system without appropriate support for configuring anycast addresses, a DNS server would have to use something like Berkeley Packet Filter (BPF) to snoop on received packets to intercept DNS requests, which is inelegant and inefficient.
- o Without appropriate routing changes elsewhere in the network, there's no reason to assume that packets sent to that anycast address would even make it to the desired DNS server machine. This places an additional configuration burden on the network administrators, to install appropriate routing table entries to direct packets to the desired DNS server machine.

In contrast, a NAT gateway is something a client's packets stumble across as they try to leave the local network and head out onto the public Internet. The NAT gateway has to be on the path those packets naturally take or it can't perform its NAT functions. As a result, the objections to using anycast for DNS server discovery do not apply to PCP:

- o No routing changes are needed (or desired) elsewhere in the local network, because the whole *point* of using anycast is that we want the client's PCP request packet to take the same forwarding path through the network as a TCP SYN to any other remote destination address, because we want the *same* NAT gateway that would have made a mapping in response to receiving an outbound TCP

SYN packet from the client to be the one that makes a mapping in response to receiving a PCP request packet from the client.

- o A NAT engine is already snooping on (and rewriting) every packet it forwards. As part of that snooping it could trivially look for packets addressed to the PCP UDP port and process them locally (just like the local processing it already does when it sees an outbound TCP SYN packet).

4. IANA Considerations

IANA should allocate an IPv4 and an IPv6 well-known PCP anycast address.

192.0.0.0/24 and 2001:0000::/23 are reserved for IETF Protocol Assignments, as listed at
<<http://www.iana.org/assignments/iana-ipv4-special-registry/>> and
<<http://www.iana.org/assignments/iana-ipv6-special-registry/>>

Suitable addresses in these ranges, such as 192.0.0.8, and a corresponding suitable IPv6 address, should be allocated.

5. Security Considerations

In a network without any border gateway, NAT or firewall that is aware of the PCP anycast address, outgoing PCP requests could leak out onto the external Internet, possibly revealing information about internal devices.

Using an IANA-assigned well-known PCP anycast address enables border gateways to block such outgoing packets. In the default-free zone, routers should be configured to drop such packets. Such configuration can occur naturally via BGP messages advertising that no route exists to said address.

Sensitive clients that do not wish to leak information about their presence can set an IP TTL on their PCP requests that limits how far they can travel into the public Internet.

6. References

6.1. Normative References

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

6.2. Informative References

[DNSDisc] Hagino, J. and D. Thaler, "Analysis of DNS Server Discovery Mechanisms for IPv6", draft-ietf-ipngwg-dns-discovery-01 (work in progress), November 2001.

[Recurs] Cheshire, S., "Recursive PCP", draft-cheshire-recursive-pcp-02 (work in progress), Mar 2013.

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: September 11, 2013

S. Cheshire
Apple
Mar 10, 2013

Recursive PCP
draft-cheshire-recursive-pcp-02

Abstract

The Port Control Protocol (PCP) allows clients to request explicit dynamic inbound and outbound port mappings in their closest on-path NAT, firewall, or other middlebox. However, in today's world, there may be more than one NAT on the path between a client and the public Internet. This document describes how the closest on-path middlebox generates a corresponding upstream PCP request to the next closest on-path middlebox, to request an appropriate explicit dynamic port mapping in that middlebox too. Applied recursively, this generates the necessary chain of port mappings in any number of middleboxes on the path between the client and the public Internet.

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 11, 2013.

Copyright Notice

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

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

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

1. Introduction

When NAT Port Mapping Protocol [NAT-PMP] was first created in 2004, a common network configuration was that a residential customer received a single public routable IPv4 address from their ISP, and had a single NAT gateway serving multiple computers in their home. Consequently, creating appropriate mappings in that single NAT gateway was sufficient to provide full Internet connectivity.

In today's world, with public routable IPv4 addresses becoming less readily available, it is increasingly common for customers to receive a private address from their ISP, and the ISP uses a NAT gateway of its own to translate those packets before sending them out onto the public Internet. This means that there is likely to be more than one NAT on the path between client machines and the public Internet:

- o If a residential customer receives a translated address from their ISP, and then installs their own residential NAT gateway to share that address between multiple client devices in their home, then there are at least two NAT gateways on the path between client devices and the public Internet.
- o If a mobile phone customer receives a translated address from their mobile phone carrier, and uses "Personal Hotspot" or "Internet Sharing" software on their mobile phone to make Wi-Fi Internet access available to other client devices, then there are at least two NAT gateways on the path between those client devices and the public Internet.
- o If a hotel guest connects a portable Wi-Fi gateway, such as an Apple AirPort Express, to their hotel room Ethernet port to share their room's Internet connection between their phone, their iPad, and their laptop computer, then packets from the client devices may traverse the hotel guest's portable NAT, the hotel network's NAT, and the ISP's NAT before reaching the public Internet.

While it is possible, in theory, that client devices could somehow discover all the NATs on the path, and communicate with each one separately using Port Control Protocol [PCP] (NAT-PMP's IETF Standards Track successor), in practice it's not clear how client devices would reliably learn this information. Since the NAT

gateways are installed and operated by different individuals and organizations, no single entity has knowledge of all the NATs on the path. Also, even if a client device could somehow know all the NATs on the path, requiring a client device to communicate separately with all of them imposes unreasonable complexity on PCP clients, many of which are expected to be simple low-cost devices.

In addition, this goes against the spirit of NAT gateways. The main purpose of a NAT gateway is to make multiple downstream client devices making outgoing TCP connections to appear, from the point of view of everything upstream of the NAT gateway, to be a single client device making outgoing TCP connections. In the same spirit, it makes sense for a PCP-capable NAT gateway to make multiple downstream client devices requesting port mappings to appear, from the point of view of everything upstream of the NAT gateway, to be a single client device requesting port mappings.

This document specifies how a PCP-capable NAT gateway uses Recursive PCP to create the appearance of being a single device, from the point of view of the upstream network.

1.1. Conventions and Terminology Used in this Document

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].

Where this document uses the terms "upstream" and "downstream", the term "upstream" refers to the direction outbound packets travel towards the public Internet, and the term "downstream" refers to the direction inbound packets travel from the public Internet towards client systems. Typically when a home user views a web site, their computer sends an outbound TCP SYN packet upstream towards the public Internet, and an inbound downstream TCP SYN ACK reply comes back from the public Internet.

1.2. Recursive Application

The protocol specified is described as "recursive" because of the following properties:

- o When the text refers to the upstream PCP server as if it were the final outermost NAT gateway, in fact that upstream PCP server could itself be another Recursive PCP server making requests to its own upstream PCP server, and relaying back the corresponding replies. That distinction is invisible to the PCP client making the request.

- o When the text refers to an incoming PCP request being received from a downstream PCP client, that downstream PCP client could itself be a Recursive PCP server relaying a request on behalf of one of its own downstream PCP clients (which could itself be another Recursive PCP server, and so on). The fact that the Recursive PCP server receiving the request does not need to be aware of this or take any special action, is an important simplifying property of the protocol. The purpose of a NAT gateway is to make many downstream client devices appear to be a single client device, and the purpose of a Recursive PCP server is to make many downstream client devices making PCP requests appear to be a single client device making PCP requests.

This recursive operation is an important simplifying property of the design.

When a PCP client talks to a PCP server, that PCP server behaves **exactly** as if it were the one and only NAT gateway on the path to the public Internet. If the PCP server is not in fact the final outermost NAT gateway, it is the PCP server's responsibility to hide that fact. The client should never have to be aware of the difference between talking to a single NAT gateway, and talking to a NAT gateway which is itself behind one or more other NAT gateways. This simplifying property applies both when the PCP client is a simple end-host client, and when the PCP client is itself the client face of a Recursive PCP server.

Similarly, when a PCP server receives a request from a PCP client, that PCP client behaves exactly as if it were a simple end-host PCP client requesting mappings for itself. If the client is not in fact a simple end-host PCP client, it is the PCP client's responsibility to hide that fact. The server should never have to be aware of the difference between talking to an end-host PCP client, and talking to the client face of a Recursive PCP server that is requesting mappings on behalf of its own downstream clients. If the PCP client is a firewall device, and it chooses to use the PCP THIRD_PARTY Option to make mappings on behalf of its downstream clients, then it should still behave like any other PCP client using the THIRD_PARTY Option.

2. Operation of Recursive PCP

Upon receipt of a PCP mapping-creation request from a downstream PCP client, a Recursive PCP server first examines its local mapping table to see if it already has a valid active mapping matching the Internal Address and Internal Port (and in the case of PEER requests, remote peer) given in the request.

If the Recursive PCP server does not already have a valid active mapping for this mapping-creation request, then it allocates an available port on its external interface. We assume for the sake of this description that the address of its external interface is itself a private address, subject to translation by an upstream NAT. The Recursive PCP server then constructs an appropriate corresponding PCP request of its own (described below), and sends it to its upstream NAT, and the newly-created local mapping is considered temporary until a confirming reply is received from the upstream PCP server.

If the Recursive PCP server does already have a valid active mapping for this mapping-creation request, and the lifetime remaining on the local mapping is at least 3/4 of the lifetime requested by the PCP client, then the Recursive PCP server SHOULD send an immediate reply giving the outermost External Address and Port (previously learned using Recursive PCP, as described below), and the actual lifetime remaining for this mapping. If the lifetime remaining on the local mapping is less than 3/4 of the lifetime requested by the PCP client, then the Recursive PCP server MUST generate an upstream request as described below.

For mapping-deletion requests (Lifetime = 0), the local mapping, if any, is deleted, and then (regardless of whether a local mapping existed) a corresponding upstream request is generated.

How the Recursive PCP server knows the destination IP address for its upstream PCP request is outside the scope of this document, but this may be achieved in a zero-configuration manner using PCP Anycast [Anycast]. In the upstream PCP request:

- o The PCP Client's IP Address and Internal Port are the Recursive PCP server's own external address and port just allocated for this mapping.
- o The Suggested External Address and Port in the upstream PCP request SHOULD be copied from the original PCP request.
- o The Requested Lifetime is as requested by the client if it falls within the acceptable range for this PCP server; otherwise it SHOULD be capped to appropriate minimum and maximum values configured for this PCP server.
- o The Mapping Nonce is copied from the original PCP request.
- o For PEER requests, the Remote Peer IP Address and Port are copied from the original PCP request.

- o Any options in the original PCP request are handled or rejected locally. No options are blindly copied from the original PCP request to the upstream PCP request. Options in the original PCP request pertain to the transaction between the client and its Recursive PCP server. In the new upstream PCP request PCP options may also be used if necessary to create the desired mapping, but they are best thought of as new options pertaining to the transaction between the Recursive PCP server and its upstream PCP server, rather than as pre-existing options that were "copied" from the original PCP request (even if, in some cases, the content of those new options may be similar or identical to the options in the original PCP request).

Upon receipt of a PCP reply giving the outermost (i.e. publicly routable) External Address, Port and Lifetime, the Recursive PCP server records this information in its own mapping table and relays the information to the requesting downstream PCP client in a PCP reply. The Recursive PCP server therefore records, among other things, the following information in its mapping table:

- o Client's Internal Address and Port.
- o External Address and Port allocated by this Recursive PCP server.
- o Outermost External Address and Port allocated by the upstream PCP server.
- o Mapping lifetime (also dictated by the upstream PCP server).
- o Mapping nonce.

In the downstream PCP reply:

- o The Lifetime is as granted by the upstream PCP server, or less, if the granted lifetime exceeds the maximum lifetime this PCP server is configured to grant. If the downstream Lifetime is more than the Lifetime granted by the upstream PCP server (which is NOT RECOMMENDED) then this Recursive PCP server MUST take responsibility for renewing the upstream mapping itself.
- o The Epoch Time is *this* Recursive PCP server's Epoch Time, not the Epoch Time of the upstream PCP server. Each PCP server has its own independent Epoch Time. However, if the Epoch Time received from the upstream PCP server indicates a loss of state in that PCP server, the Recursive PCP server can either recreate the lost mappings itself, or it can reset its own Epoch Time to cause its downstream clients to perform such state repairs themselves. A Recursive PCP server MUST NOT simply copy the upstream PCP

server's Epoch Time into its downstream PCP replies, since if it suffers its own state loss it needs the ability to communicate that state loss to clients. Thus each PCP server has its own independent Epoch Time. However, as a convenience, a downstream Recursive PCP server may simply choose to reset its own Epoch Time whenever it detects that its upstream PCP server has lost state. Thus, in this case, the Recursive PCP server's Epoch Time always resets whenever its upstream PCP server loses state; it may also reset at other times too.

- o The Mapping Nonce is copied from the reply received from the upstream PCP server.
- o The Assigned External Port and Assigned External IP Address are copied from the reply received from the upstream PCP server. (I.e. they are the outermost External IP Address and Port, not the locally-assigned external address and port.)
- o For PEER requests, the Remote Peer IP Address and Port are copied from the reply received from the upstream PCP server.
- o Any options in the reply received from the upstream PCP server are handled locally as appropriate to the options in question. No options are blindly copied from the upstream PCP reply to the downstream PCP reply. If the original PCP request contained options which necessitate a corresponding option in the reply, then appropriate reply options should be generated and inserted into the downstream PCP reply by the Recursive PCP server. These downstream reply options are best thought of as data pertaining to the transaction between the Recursive PCP server and its downstream client, rather than as pre-existing options that were "copied" from the upstream PCP reply into the downstream PCP reply (even if, in some cases, the content of those new options in the downstream PCP reply may be similar or identical to the options received in the reply from the upstream PCP server).

2.1. Optimized Hairpin Routing

A Recursive PCP server SHOULD implement Optimized Hairpin Routing. What this means is the following:

- o If a Recursive PCP server observes an outgoing packet arriving on its internal interface that is addressed to an External Address and Port appearing in the NAT gateway's own mapping table, then the NAT gateway SHOULD (after creating a new outbound mapping if one does not already exist) rewrite the packet appropriately and deliver it to the internal client currently allocated that External Address and Port.

- o If a Recursive PCP server observes an outgoing packet arriving on its internal interface which is addressed to an Outermost External Address and Port appearing in the NAT gateway's own mapping table, then the NAT gateway SHOULD do likewise: create a new outbound mapping if one does not already exist, and then rewrite the packet appropriately and deliver it to the internal client currently allocated that Outermost External Address and Port. This is not necessary for successful communication, but for efficiency. Without this Optimized Hairpin Routing, the packet will be delivered all the way to the outermost NAT gateway, which will then perform standard hairpin translation and send it back. Using knowledge of the Outermost External Address and Port, this rewriting can be anticipated and performed locally, which will typically offer higher throughput and lower latency than sending it all the way to the outermost NAT gateway and back.

2.2. Termination of Recursion

Any recursive algorithm needs a mechanism to terminate the recursion at the appropriate point. This termination of recursion can be achieved in a variety of ways:

- o An ISP's NAT gateway could be configured to know that it is the outermost NAT gateway, and consequently does not need to relay PCP requests upstream. In fact, it may be the case that many large-scale NATs of the kind used by ISPs may simply not implement Recursive PCP, thereby naturally terminating the recursion at that point.
- o A NAT gateway could determine automatically that if its external address is not one of the known private addresses [RFC1918][RFC6598] then its external address is a public routable IP address, and consequently it does not need to relay PCP requests upstream.
- o A NAT gateway could attempt sending PCP requests upstream, and upon failing to receive any positive reply (e.g. receiving ICMP host unreachable, ICMP port unreachable, or a timeout) conclude that it does not need to relay PCP requests upstream.

2.3. Recursive PCP with Firewalls

When a Recursive PCP server is a NAT gateway, it sends out upstream PCP requests using its own external IP address. When a Recursive PCP server is a firewall, it still needs to install upstream mappings on behalf of its downstream clients. It should do this either by using the downstream client's IP address as the source IP address in its upstream PCP request, or by using the PCP THIRD_PARTY Option in its

upstream PCP request.

3. IANA Considerations

No IANA actions are required by this document.

4. Security Considerations

No new security concerns are raised by use of Recursive PCP. Since the purpose of a NAT gateway is to enable multiple client devices to appear as a single client device to the upstream network, a NAT gateway implementing Recursive PCP maintains this property, appearing to the upstream network to be a single client device using PCP to request port mappings for itself. Whether those port mappings are for multiple processes running on multiple CPUs connected via an internal bus in a single computer, or multiple processes running on multiple CPUs connected via an IP network, is transparent to the external network.

5. References

5.1. Normative References

- [PCP] 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.
- [RFC1918] Rekhter, Y., Moskowitz, R., Karrenberg, D., Groot, G., and E. Lear, "Address Allocation for Private Internets", BCP 5, RFC 1918, February 1996.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC6598] Weil, J., Kuarsingh, V., Donley, C., Liljenstolpe, C., and M. Azinger, "IANA-Reserved IPv4 Prefix for Shared Address Space", BCP 153, RFC 6598, April 2012.

5.2. Informative References

- [Anycast] Cheshire, S., "PCP Anycast Address", draft-cheshire-pcp-anycast-00 (work in progress), February 2013.
- [NAT-PMP] Cheshire, S., "NAT Port Mapping Protocol (NAT-PMP)",

draft-cheshire-nat-pmp-07 (work in progress),
January 2013.

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: August 23, 2013

M. Boucadair
France Telecom
R. Penno
D. Wing
Cisco
February 19, 2013

DHCP Options for the Port Control Protocol (PCP)
draft-ietf-pcp-dhcp-06

Abstract

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

Requirements Language

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

Status of this Memo

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

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

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

This Internet-Draft will expire on August 23, 2013.

Copyright Notice

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

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

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

Table of Contents

1. Introduction	3
2. Terminology	3
3. DHCPv6 PCP Server Option	3
3.1. Format	3
3.2. Client Behavior	4
4. DHCPv4 PCP Option	5
4.1. Format	5
4.2. Client Behavior	6
5. Use of PCP Server Names	6
6. Dual-Stack Hosts	7
7. Security Considerations	7
8. IANA Considerations	7
8.1. DHCPv6 Option	7
8.2. DHCPv4 Option	7
9. Acknowledgements	7
10. References	8
10.1. Normative References	8
10.2. Informative References	8
Appendix A. Rationale	9
A.1. Dependency on Name Resolution	10
Authors' Addresses	11

1. Introduction

This document defines DHCPv4 [RFC2131] and DHCPv6 [RFC3315] options which can be used to provision PCP Server [I-D.ietf-pcp-base] names. Motivations for expressing the PCP option as a textual string rather than a 32 or 128-bit binary address are discussed in Appendix A.

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

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

2. Terminology

This document makes use of the following terms:

- o PCP Server denotes a functional element which receives and processes PCP requests from a PCP Client. A PCP Server can be co-located with or be separated from the function (e.g., NAT, Firewall) it controls. Refer to [I-D.ietf-pcp-base].
- o PCP Client denotes a PCP software instance responsible for issuing PCP requests to a PCP Server. Refer to [I-D.ietf-pcp-base].
- o DHCP refers to both DHCPv4 [RFC2131] and DHCPv6 [RFC3315].
- o DHCP client (or client) denotes a node that initiates requests to obtain configuration parameters from one or more DHCP servers.
- o DHCP server (or server) refers to a node that responds to requests from DHCP clients.
- o Name is a UTF-8 [RFC3629] string that can be passed to getaddrinfo (Section 6.1 of [RFC3493]), such as a DNS name, address literals, etc. The name MUST NOT contain spaces or nulls. A name may be a fully qualified domain name (e.g., "myservice.example.com."), IPv4 address in dotted-decimal form (e.g., 192.0.2.33) or textual representation of an IPv6 address (e.g., 2001:db8::1) [RFC4291][RFC5952].

3. DHCPv6 PCP Server Option

This DHCPv6 option conveys a 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.

3.1. Format

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

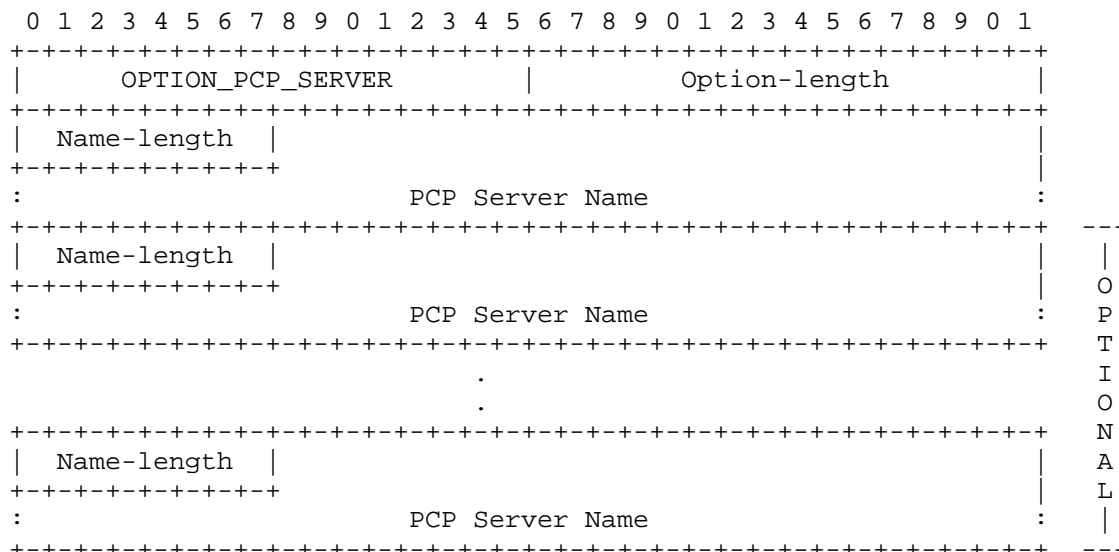


Figure 1: PCP Server Name DHCPv6 Option

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

- o Option-code: OPTION_PCP_SERVER (TBA, see Section 8.1)
- o Option-length: includes total length of all following option data in octets.
- o Name-length (one-octet field): Includes the length of the PCP Server Name, in octets.
- o PCP Server Name (variable): The name of the PCP Server to be used by the PCP Client. The name is encoded as a UTF-8 [RFC3629] string.

The OPTION_PCP_SERVER option can include multiple PCP Server names; each name is treated as a separate PCP Server. When several names are to be included, "Name-length" and "PCP Server Name" fields are repeated.

3.2. Client Behavior

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).

If the DHCPv6 client receives an OPTION_PCP_SERVER option from the DHCPv6 server, it extracts the name(s) conveyed in the OPTION_PCP_SERVER option. A name is considered as valid if it is a

legal UTF-8 string which does not contain any spaces or nulls. Below are listed some additional validation rules:

- o The trailing dot is optional when a domain name is conveyed in the option.
- o IPv6 addresses MUST NOT be enclosed in brackets.
- o A domain name is structured as one or more labels concatenated with dots. A label MUST have no more than 63 characters.

The DHCPv6 client MUST silently ignore invalid names.

Once each name conveyed in the OPTION_PCP_SERVER option is validated, the DHCPv6 client MUST follow the procedure specified in Section 5.

4. DHCPv4 PCP Option

4.1. Format

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

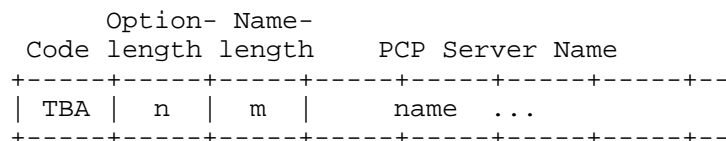


Figure 2: PCP Server Name DHCPv4 Option

The description of the fields is as follows:

- o Code: OPTION_PCP_SERVER (TBA, see Section 8.2);
- o Option-length: includes total length of all following option data in octets. The maximum length is 255 octets.
- o Name-length (one-octet field): Includes the length of the PCP Server Name, in octets.
- o PCP Server Name (variable): The name of the PCP Server to be used by the PCP Client when issuing PCP messages. The name is encoded as a UTF-8 [RFC3629] string.

The OPTION_PCP_SERVER option can include multiple PCP Server names; each name is treated as a separate PCP Server. When several names are to be included, "Name-length" and "PCP Server Name" fields are repeated.

The OPTION_PCP_SERVER DHCPv4 option is a concatenation-requiring option. As such, the mechanism specified in [RFC3396] MUST be used if the PCP Server Name option exceeds the maximum DHCPv4 option size of 255 octets.

4.2. Client Behavior

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

If the DHCPv4 client receives an OPTION_PCP_SERVER option from the DHCPv4 server, it extracts the name(s) conveyed in the option. A name is considered as valid if it is a legal UTF-8 string which does not contain any spaces or nulls. Below are listed some additional validation rules:

- o The trailing dot is optional when a domain name is conveyed in the option.
- o A domain name is structured as one or more labels concatenated with dots. A label MUST have no more than 63 characters.

The DHCPv4 client MUST silently discard non valid names.

Once each name conveyed in the OPTION_PCP_SERVER option is validated, the DHCPv4 client MUST follow the procedure specified in Section 5.

5. Use of PCP Server Names

Each configured PCP Server Name is passed to the name resolution library (e.g., Section 6.1.1 of [RFC1123] or [RFC6055]) to retrieve the corresponding IP address(es) (IPv4 or IPv6). It is out of scope of this document to specify how the PCP Client selects the PCP Server(s) to contact.

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

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

6. Dual-Stack Hosts

In some deployment contexts, the PCP Server may be reachable with an IPv4 address but DHCPv6 is used to provision the PCP Client. In such scenarios, a plain IPv4 address or an IPv4-mapped IPv6 address can be configured to reach the PCP Server.

A Dual-Stack host may receive `OPTION_PCP_SERVER` via both DHCPv4 and DHCPv6. The content of these `OPTION_PCP_SERVER` options may refer to the same or distinct PCP Servers. This is deployment-specific and as such it is out of scope of this document.

7. Security Considerations

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

8. IANA Considerations

8.1. DHCPv6 Option

Authors of this document request the following DHCPv6 option code:

Option Name	Value
<code>OPTION_PCP_SERVER</code>	TBA

8.2. DHCPv4 Option

Authors of this document request the following DHCPv4 option code:

Option Name	Value
<code>OPTION_PCP_SERVER</code>	TBA

9. Acknowledgements

Many thanks to B. Volz, C. Jacquenet, R. Maglione, D. Thaler, T. Mrugalski, T. Lemon, S. Cheshire and M. Wasserman for their review and 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-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.
- [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.
- [RFC3629] Yergeau, F., "UTF-8, a transformation format of ISO 10646", STD 63, RFC 3629, November 2003.
- [RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, February 2006.
- [RFC5952] Kawamura, S. and M. Kawashima, "A Recommendation for IPv6 Address Text Representation", RFC 5952, August 2010.

10.2. Informative References

- [I-D.ietf-behave-lsn-requirements]
Perreault, S., Yamagata, I., Miyakawa, S., Nakagawa, A., and H. Ashida, "Common requirements for Carrier Grade NATs (CGNs)", draft-ietf-behave-lsn-requirements-10 (work in progress), December 2012.
- [I-D.ietf-dhc-option-guidelines]
Hankins, D., Mrugalski, T., Siodelski, M., Jiang, S., and S. Krishnan, "Guidelines for Creating New DHCPv6 Options", draft-ietf-dhc-option-guidelines-09 (work in progress), December 2012.
- [RFC1123] Braden, R., "Requirements for Internet Hosts - Application

and Support", STD 3, RFC 1123, October 1989.

- [RFC2181] Elz, R. and R. Bush, "Clarifications to the DNS Specification", RFC 2181, July 1997.
- [RFC3493] Gilligan, R., Thomson, S., Bound, J., McCann, J., and W. Stevens, "Basic Socket Interface Extensions for IPv6", RFC 3493, February 2003.
- [RFC6055] Thaler, D., Klensin, J., and S. Cheshire, "IAB Thoughts on Encodings for Internationalized Domain Names", RFC 6055, February 2011.
- [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.
- [RFC6334] Hankins, D. and T. Mrugalski, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6) Option for Dual-Stack Lite", RFC 6334, August 2011.

Appendix A. Rationale

The pcp WG consensus is to define a DHCP option which contains a string that can be passed to APIs (e.g., `getaddrinfo()`). In particular, the option should be designed to include a name (e.g., domain name [RFC2181]) or IP address literal string. In such design, DHCP clients are expected to pass the conveyed string to any supported name resolution library (DNS is a name resolution service among others). The underlying name resolution library is responsible for validating the name.

Distinct IP-Address and Name DHCP options have been considered in early stages of this specification. This flexibility aims to let service providers make their own engineering choices and use the most convenient option according to their deployment context. Nevertheless, the DHC WG's position is this flexibility has some drawbacks such as inducing errors (See Section 7 of [I-D.ietf-dhc-option-guidelines]). Therefore, only the Name option is maintained within this document.

This choice is motivated by operational considerations: In particular, some Service Providers are considering two levels of

redirection:

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

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

In addition to the operational considerations:

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

A.1. Dependency on Name Resolution

The approach adopted in this document allows for an IP address or a Name to be returned in the specified DHCP option. In particular, a server can resolve first the name and return in the option the resolved IP address(es). For deployments where this is not possible, the server can return a name which will be resolved by the host embedding the client. This document does not have any requirement on the underlying name resolution library (in particular, DNS is not assumed as the only available name resolution service).

Returning a Name requires the host to embed a name resolution service. Some may present this as an argument against defining a Name option. Nevertheless, this argument may be objected as implementing a name resolution library (e.g., embed a DNS resolver) is cheap and devices which don't embed DNS resolver are uncommon.

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: Standards Track
Expires: August 16, 2013

M. Boucadair
France Telecom
February 12, 2013

Learn NAT64 PREFIX64s using PCP
draft-ietf-pcp-nat64-prefix64-00

Abstract

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

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 16, 2013.

Copyright Notice

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

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

Table of Contents

1. Introduction	3
2. Requirements Language	3
3. Problem Statement	3
3.1. Issues	3
3.2. Use Cases	3
3.2.1. AAAA Synthesis by Stub-resolver	3
3.2.2. Applications Referrals	4
3.3. Illustration Example	4
4. PREFIX64 Option	5
4.1. Format	5
4.2. Behaviour	6
5. Flow Example	6
6. IANA Considerations	8
7. Security Considerations	8
8. Acknowledgements	8
9. References	9
9.1. Normative References	9
9.2. Informative References	9
Author's Address	9

1. Introduction

This document defines a new PCP extension [I-D.ietf-pcp-base] to inform PCP Clients about the Pref64::

This extension is required to help establishing communications between IPv6-only hosts and remote IPv4-only hosts.

2. Requirements Language

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

3. Problem Statement

3.1. Issues

This document proposes a deterministic solution to solve the following issues:

- o Learn the Pref64:: - * distinguishing between IPv4-converted IPv6 addresses and native IPv6 addresses.
 - * implementing IPv6 address synthesis for applications not relying on DNS.
- o Avoid stale Pref64::- o Discover multiple Pref64::- o Use DNSSEC in the presence of NAT64.

Section 3.2 lists some applications which encounter the issues listed above.

3.2. Use Cases

3.2.1. AAAA Synthesis by Stub-resolver

The extension defined in this document can be used for hosts with DNS64 capability [RFC6147], added to the host's stub-resolver.

The stub resolver on the host will try to obtain (native) AAAA records and if they are not found, the DNS64 function on the host will query for A records and then synthesizes AAAA records. Using the PREFIX64 PCP extension, the host's stub-resolver can learn the prefix used for IPv6/IPv4 translator and synthesize AAAA records

accordingly.

Learning the Pref64::/n used to construct IPv4-converted IPv6 addresses [RFC6052] allows to make use of DNSSEC.

3.2.2. Applications Referrals

This PCP extension can be used by applications making use of address referrals.

As Peer-to-Peer (P2P) communications for real-time communication is becoming popular with RTCWEB (e.g., P2P for Media, data channels for file transfer etc), this extension can be used to help for NAT64 traversal. SIP [RFC3261] is only one example among those protocols.

3.3. Illustration Example

An illustration example is shown in Figure 1. In this example, NAT64 is co-located with a PCP server while IPv6-only SIP UA interacts with a PCP Client.

In Figure 1, the PCP Client issues a PCP MAP request with PORT_RESERVATION_OPTION to reserve a pair of ports preserving parity and contiguity [I-D.boucadair-pcp-rtp-rtcp]. A pair of ports and an external IPv4 address are then returned by the PCP server to the requesting PCP Client. This information is used by the IPv6-only SIP UA to build its SDP offer which contains exclusively IPv4 addresses (especially in the "c=" line, the port indicated for media port is the external port assigned by the PCP server). The INVITE request including the SDP offer is then forwarded by the NAT64 to the Proxy Server which will relay it to the called party (i.e., IPv4-only SIP UA) (Steps (1) to (3)). IPv4-only SIP UA accepts the offer and sends back its SDP answer in a "200 OK" message which is relayed by the SIP Proxy Server and NAT64 until being delivered to IPv6-only SIP UA (Steps (4) to (6)).

At the end of this process, IPv4-only SIP UA can send media streams to the IPv4 address/port as indicated in the SDP offer while IPv6-only SIP UA can not send media streams as only IPv4 addresses are present in the SDP answer.

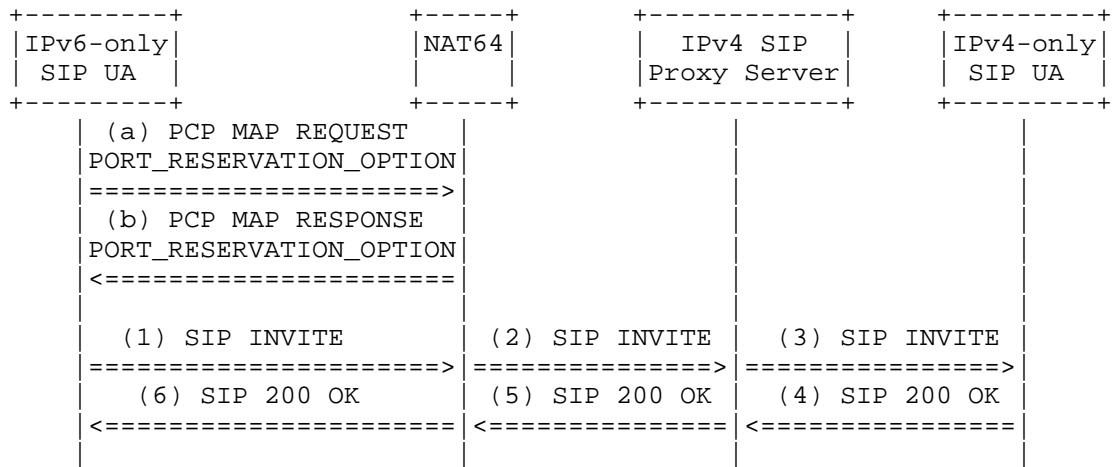


Figure 1

4. PREFIX64 Option

4.1. Format

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

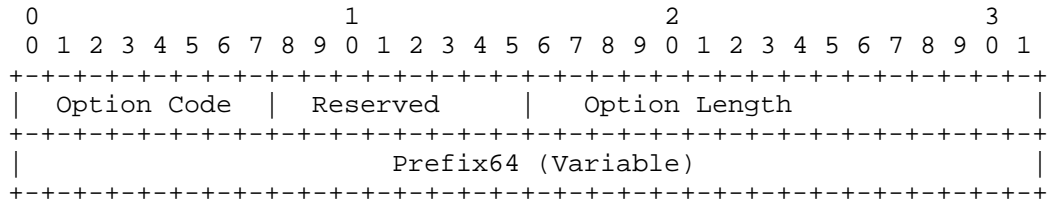


Figure 2: Prefix64 PCP Option

The description of the fields is as follows:

- o Option Code: To be assigned by IANA.
- o Option Length: Indicates in octets the length of the Pref64::/n. Allowed values are 4, 5, 6, 7, 8, or 12 [RFC6052].
- o Prefix64: This field identifies the IPv6 unicast prefix to be used for constructing an IPv4-embedded IPv6 address from an IPv4 address. The address synthesize MUST follow the guidelines documented in [RFC6052].

Option Name: PREFIX64
Number: To be assigned by IANA.
Purpose: Learn the prefix used by the NAT64 to build
IPv4-embedded IPv6 addresses. This is be used by a host
for local address synthesis (e.g., when IPv4 address is
present in referrals).
Valid for Opcodes: MAP
Length: Variable
May appear in: request, response.
Maximum occurrences: 1

4.2. Behaviour

A PCP Client MAY include a PREFIX64 PCP Option in a MAP request to learn the IPv6 prefix used by an upstream PCP-controlled NAT64 device. When enclosed in a MAP request, PREFIX64 MUST be set to `::/96`. PREFIX64 PCP Option can be inserted in a MAP request used to learn the external IP address as detailed in Section 11.6 of [I-D.ietf-pcp-base].

A PCP Server controlling a NAT64 SHOULD be configured to return to requesting PCP Clients the value of the `Pref64::/n` used to build IPv4-embedded IPv6 addresses. When enabled, PREFIX64 PCP Option conveys the value of `Pref64::/n`.

A PCP Server controlling a NAT64 MAY be configured to inject a PREFIX64 PCP Option in all MAP responses even if the option is not listed in the associated request.

Upon receipt of the PREFIX64 PCP Option, the host embedding the PCP Client uses `Pref64::/n` for local address synthesise [RFC6052]. How the content of PREFIX64 PCP Option is passed to the OS is implementation-specific.

A PCP Client SHOULD associate each received `Pref64::/n` with the PCP Server from which the `Pref64::/n` information was retrieved.

5. Flow Example

Figure 3 shows an example of the use of the option defined in Section 4.

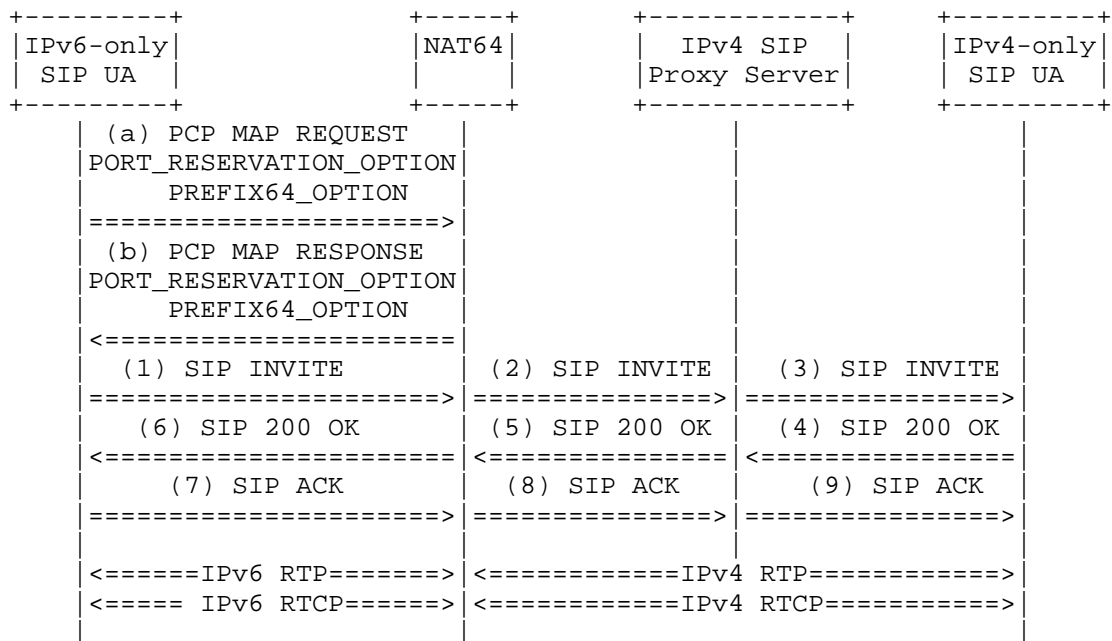


Figure 3: Example of IPv6 to IPv4 SIP initiated Session

In Steps (a) and (b), the IPv6-only SIP UA retrieves a pair of ports to be used for RTP/RTCP, the external IPv4 address and the Pref64::/n to be used to build IPv4-embedded IPv6 addresses. The retrieved IPv4 address and port numbers are used to build the SDP offer in Step (1) while Pref64::/n is used to construct a corresponding IPv6 address of the IPv4 address enclosed in the SDP answer made by the IPv4-only SIP UA (Step 6). RTP/RTCP flows are exchanged between an IPv6-only SIP UA and an IPv4-only UA without requiring any ALG at the NAT64 and no particular function to be supported by the IPv4-only SIP Proxy Server to help establishing the session (e.g., Hosted NAT traversal).

When the session is initiated from IPv4 SIP UA (see Figure 4): Steps (a) and (b), the IPv6-only SIP UA retrieves a pair of ports to be used for RTP/RTCP, the external IPv4 address and the Pref64::/n to be used to build IPv4-embedded IPv6 addresses. These two steps can be delayed until receiving the INVITE message (Step 3).

The retrieved IPv4 address and port numbers are used to build the SDP answer in Step (4) while Pref64::/n is used to construct a corresponding IPv6 address of the IPv4 address enclosed in the SDP offer made by the IPv4-only SIP UA (Step 3). RTP/RTCP flows are exchanged between an IPv6-only SIP UA and an IPv4-only UA without

requiring any ALG at the NAT64 and no particular function to be supported by the IPv4-only SIP Proxy Server to help establishing the session (e.g., Hosted NAT traversal).

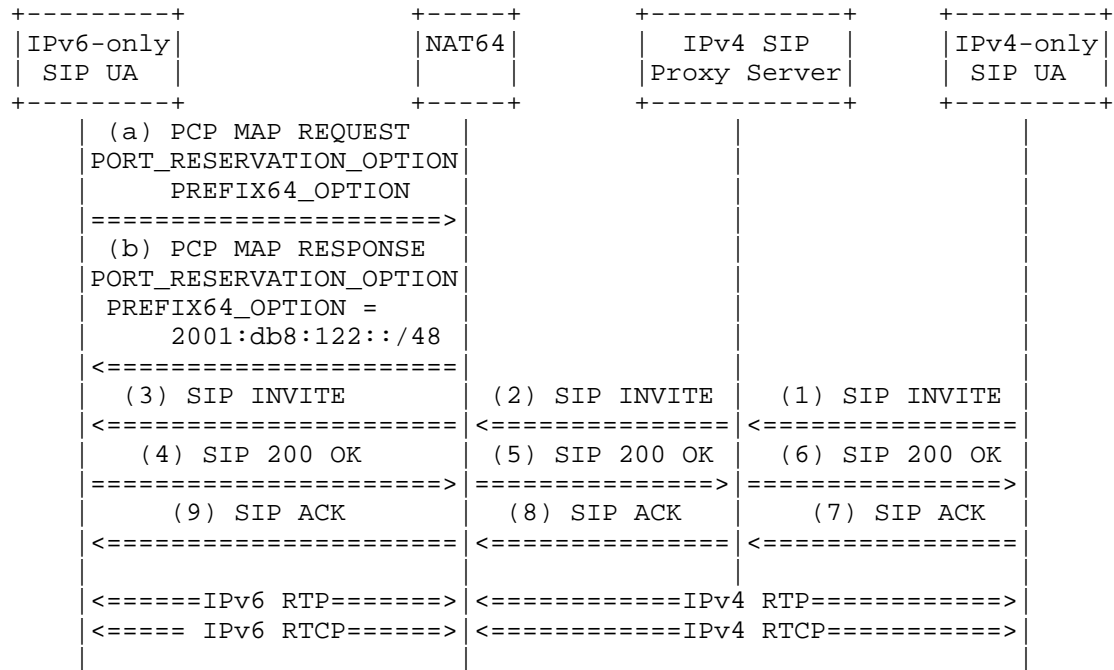


Figure 4: Example of IPv4 to IPv6 SIP initiated Session

6. IANA Considerations

This document requests a new PCP option:
PREFIX64

7. Security Considerations

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

8. Acknowledgements

Many thanks to S. Perreault , R. Tirumaleswar, T. Tsou, D. Wing, J.

Zhao and R. Penno for the comments and suggestions.

9. References

9.1. Normative References

- [I-D.ietf-pcp-base]
Wing, D., Cheshire, S., Boucadair, M., Penno, R., and P. Selkirk, "Port Control Protocol (PCP)", draft-ietf-pcp-base-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.
- [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.
- [RFC6052] Bao, C., Huitema, C., Bagnulo, M., Boucadair, M., and X. Li, "IPv6 Addressing of IPv4/IPv6 Translators", RFC 6052, October 2010.
- [RFC6146] Bagnulo, M., Matthews, P., and I. van Beijnum, "Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers", RFC 6146, April 2011.
- [RFC6147] Bagnulo, M., Sullivan, A., Matthews, P., and I. van Beijnum, "DNS64: DNS Extensions for Network Address Translation from IPv6 Clients to IPv4 Servers", RFC 6147, April 2011.

9.2. Informative References

- [I-D.boucadair-pcp-rtp-rtcp]
Boucadair, M. and S. Sivakumar, "Reserving N and N+1 Ports with PCP", draft-boucadair-pcp-rtp-rtcp-05 (work in progress), October 2012.

Author's Address

Mohamed Boucadair
France Telecom
Rennes, 35000
France

Email: mohamed.boucadair@orange.com

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

M. Boucadair
France Telecom
R. Penno
D. Wing
Cisco
February 13, 2013

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

Abstract

This document specifies a new PCP functional element denoted as PCP Proxy. The PCP Proxy relays PCP requests received from PCP Clients to upstream PCP Server(s). This function is mandatory when PCP Clients can not be configured with the address of the PCP Server located more than one hop.

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 17, 2013.

Copyright Notice

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

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of

the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
2. Requirements Language	3
3. PCP Server Discovery and Provisioning	4
4. PCP Proxy as a PCP Server	4
5. Control of the Firewall	4
6. No NAT is Co-located with the PCP Proxy	4
7. PCP Proxy Co-located with a NAT Function	5
8. MAP/PEER Handling	6
9. Mapping Repair	7
10. Advanced Functions	8
10.1. Multiple PCP Servers	8
10.2. Epoch Handling	8
10.3. Request/Response Caching	9
10.4. Retransmission Handling	9
10.5. Full State	9
11. IANA Considerations	9
12. Security Considerations	9
13. Acknowledgements	10
14. References	10
14.1. Normative References	10
14.2. Informative References	10
Authors' Addresses	11

1. Introduction

This document defines a new PCP [I-D.ietf-pcp-base] function element, called PCP Proxy, which is meant to facilitate the communication between a PCP Client and upstream PCP Server(s). The PCP Proxy 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 PCP Server. The PCP Server in turn sends PCP responses to the PCP Proxy external interface which are finally forwarded to PCP Clients. A reference architecture is depicted in Figure 1.

A PCP Proxy can be for instance embedded in a CP (Customer Premises) router while the PCP Server is located in a network operated by an ISP (Internet Service Provider). It is out of scope of this document to list all deployment scenarios requiring a PCP Proxy to be involved.

The PCP Proxy can be simple (i.e., a single-homed entity which implement as transparent/minimal processing as possible) or it can support advanced features (see Section 10). A Proxy can be co-located with UPnP IGD [I-D.ietf-pcp-upnp-igd-interworking] or/and NAT-PMP [I-D.bpw-pcp-nat-pmp-interworking] Interworking Function (IWF).

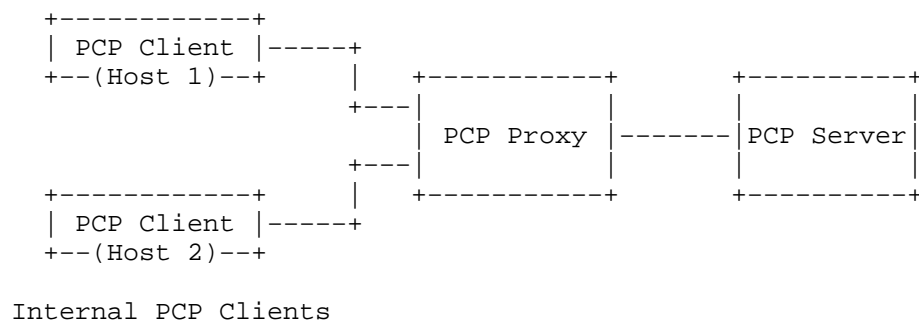


Figure 1: Reference Architecture

2. 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].

3. PCP Server Discovery and Provisioning

The PCP Proxy MUST follow the procedure defined in Section 8.1 of [I-D.ietf-pcp-base] to discover its PCP Server.

The address of the PCP Proxy is provisioned to internal PCP Clients (see Figure 1) as their default PCP Server: If the PCP DHCP option [I-D.ietf-pcp-dhcp] is supported by an internal PCP Client, it will retrieve the PCP Server IP address to use from its local DHCP server; otherwise internal PCP Clients will assume their default router being the PCP Server.

4. 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.

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

5. Control of the Firewall

A security policy to accept PCP messages from the provisioned PCP Server(s) is to be enabled on the device embedding the PCP Proxy. 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.

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

6. No NAT is Co-located with the PCP Proxy

When no NAT is co-located with the PCP Proxy, the port numbers included in received PCP messages (from the PCP Server or PCP

Client(s)) are not altered by the PCP Proxy. Nevertheless, the PCP Client IP Address MUST be changed to the address of the PCP Proxy and a THIRD_PARTY Option inserted to carry the IP address of the source PCP Client.

Because no NAT is invoked, there is no reachability failure risk to relay to the PCP Server unknown Options and OpCodes which carry an IP address.

7. PCP Proxy Co-located with a NAT Function

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

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

Rejecting unknown Options and OpCodes has the drawback of preventing a PCP Client to make use of new capabilities offered by the PCP Server but not supported by the PCP Proxy even if no IP address and/or port is included in the Option/OpCode.

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.

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.

8. MAP/PEER Handling

A simple PCP Proxy performs minimal modifications to PCP requests and responses, in particular it does not change the Nonce value in requests and the Epoch value in responses. A simple PCP Proxy is assumed to handle only one PCP Server.

For handling THIRD_PARTY option, the PCP Proxy MUST follow the PCP Server behavior specified in Section 13.1 of [I-D.ietf-pcp-base].

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

- o Check if the source IP address and the PCP Client IP Address are the same.
- o Apply security controls (e.g., THIRD_PARTY filtering).
- o If the request is rejected, build a synthetic error response and send it back to the PCP Client.
- o If the request is accepted, adjust it (e.g., adding a THIRD_PARTY Option, updating the PCP Client IP Address and Internal Port to their translated values as specified in Section 7 and forward it on a fresh UDP socket connected to the PCP Server).
- o Wait for the response during a reasonable delay.
- o When the response is received from the PCP Server, adjust it back (e.g., removing the THIRD_PARTY Option added previously, updating the PCP Client IP Address and Internal Port to their initial values as specified in Section 7), forward it to the source PCP Client and close the socket to the PCP Server.

- o 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:

- o the request payload
- o the interface where the request was received
- o the source IP address of the request
- o the source UDP port of the request
- o the UDP socket connected to the PCP server
- o 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.

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.

9. Mapping Repair

ANNOUNCE requests received from PCP Clients are handled locally; as such these requests MUST NOT be relayed to the provisioned PCP Server.

Upon receipt of an unsolicited ANNOUNCE response from a PCP Server, the PCP Proxy proceeds to renewing the mappings and checks whether there are changes compared to a local cache if it is maintained by the PCP Proxy. If no change is detected, no unsolicited ANNOUNCE is generated towards PCP Clients. If a change is detected, the PCP Proxy MUST generate unsolicited ANNOUNCE message(s) to appropriate PCP Clients. If the PCP Proxy does not maintain a local cache for

the mappings, unsolicited ANNOUNCE messages are relayed to PCP Clients.

Unsolicited PCP MAP/PEER responses received from a PCP Server are handled as any normal MAP/PEER response. To handle unsolicited PCP MAP/PEER responses, the PCP Proxy is required to maintain a local cache of instantiated mappings in the PCP Server (Section 10.5).

Upon change of its external IP address, the PCP Proxy SHOULD renew the mappings it maintained. If the PCP Server assigns a different external port, the PCP Proxy SHOULD follow the mapping repair procedure defined in [I-D.ietf-pcp-base]. This can be achieved only if a full state table is maintained by the PCP Proxy.

10. Advanced Functions

Below are listed a set of advanced features which may be supported by the PCP Proxy.

10.1. Multiple PCP Servers

A PCP Proxy MAY handle multiple PCP Servers at the same time, each PCP Server is associated to each own handled Epoch value according to Section 10.2. PCP Clients are not aware of the presence of multiple PCP Servers.

According to [I-D.ietf-pcp-dhcp], if several PCP Names are configured to the PCP Proxy, it will contact in parallel all these PCP Servers.

In some contexts (e.g., PCP-controlled CGNs), the PCP Proxy MAY load balance the PCP Client among available PCP Servers. The PCP Proxy MUST ensure requests of a given PCP Client are relayed to the same PCP Server.

In other deployment scenarios (e.g., presence of multiple PCP-controlled firewalls), the PCP Proxy MUST relay PCP requests to all these PCP Servers.

The PCP Proxy MAY rely on some fields (e.g., Zone ID [I-D.penno-pcp-zones]) in the PCP request to redirect the request to a given PCP Server.

10.2. Epoch Handling

A PCP Proxy MAY use its own internal timers and not blindly copy them 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.:

- o at startup if the PCP Proxy can't recover the state.
- o when the WAN address is changed or any similar events which show any previous state is no longer valid.
- o when the Epoch value in a PCP response is too small (cf. Epoch value validation rules in [I-D.ietf-pcp-base]).
- o when the External IP 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.

10.3. 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.

10.4. Retransmission Handling

An extension of the previous service is to manage the retransmission of pending requests to the PCP 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.

10.5. Full State

A PCP Proxy MAY keep the full state, i.e., an image of all active explicit dynamic mappings is kept in memory. When this service is supported the state SHOULD be recovered in case of failures (e.g., according to [I-D.boucadair-pcp-failure]).

11. IANA Considerations

This document makes no request of IANA.

12. Security Considerations

The PCP Proxy MUST follow the security considerations elaborated in [I-D.ietf-pcp-base] for both the client and server side.

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.

The device embedding the PCP Proxy MAY block PCP requests directly sent to the PCP Server. This can be enforced using access control lists (ACLs).

13. Acknowledgements

Many thanks to C. Zhou and T. Reddy for their review and comments.

Special thanks to F. Dupont who contributed to this document.

14. References

14.1. 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.

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

[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-06 (work in progress), December 2012.

[I-D.penno-pcp-zones]

Penno, R., "PCP Support for Multi-Zone Environments", draft-penno-pcp-zones-01 (work in progress), October 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: August 20, 2013

S. Kiesel
University of Stuttgart
R. Penno
Cisco Systems
February 16, 2013

PCP Server Discovery based on well-known IP Address
draft-kiesel-pcp-ip-based-srv-disc-00

Abstract

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 keep alive traffic.

This document establishes a well-known IP address for the PCP Server and documents how PCP clients embedded in endpoints can use it during the discovery and regular operation phases.

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 20, 2013.

Copyright Notice

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

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

Table of Contents

1. Introduction	4
2. PCP Server Discovery based on well-known IP Address	5
2.1. Well-Known PCP Server IP Address (WkPsdIPa)	5
2.2. PCP Discovery Client behavior	5
2.3. PCP Discovery Server behavior	5
3. Deployment Considerations	6
3.1. Multiple PCP Servers, Symmetric Routing	6
3.2. Multiple PCP Servers, Assymetric Routing	6
4. IANA Considerations	8
4.1. Registration of IPv4 Special Purpose Address	8
4.2. Registration of IPv6 Special Purpose Address	9
4.3. PCP Option	10
5. Security Considerations	11
6. Acknowledgements	12
7. References	13
7.1. Normative References	13
7.2. Informative References	13
Appendix A. Problems with Other Discovery methods	15
A.1. DHCP PCP Options	15
A.2. Default Router	15
A.3. User Input	15
A.4. Domain Name System Based	16
Authors' Addresses	17

1. Introduction

The Port Control Protocol (PCP) [I-D.ietf-pcp-base] 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 keep alive traffic.

But before a PCP client can perform any of these tasks it needs to discover one or more PCP servers. Several algorithms have been specified that produce a suitable PCP Server address given PCP client (i.e., the address may vary for different clients or different points of network attachment, etc.). These approaches are based on user input, DHCP [I-D.ietf-pcp-dhcp] or default router, which is the one detailed in the PCP base document [I-D.ietf-pcp-base].

But unfortunately in many deployments, the first-hop router does not run a PCP server, or DHCP cannot be used. These and other problems are described in detail in the Appendix. Appendix A.

This document follows a different approach: it establishes a well-known address for the PCP Server (TBD: this approach could easily be generalized in order to discover other services as well. But this is for further study). PCP clients are expected to send requests to this address during the PCP Server discovery process. A PCP Server configured with the anycast address could optionally redirect or return a list of unicast PCP Servers to the client.

2. PCP Server Discovery based on well-known IP Address

2.1. Well-Known PCP Server IP Address (WkPsdIPa)

IANA is requested to register a single IPv4 address 192.0.0.X (TBD) and a single IPv6 address 2001::XXXX (TBD) within the respective Special Purpose Address Registries as the well-known IP anycast addresses for PCP Servers. These addresses are called WkPsdIPa (well-known PCP server discovery IP address(es)) in this document.

2.2. PCP Discovery Client behavior

PCP Clients that need to discover PCP servers should first send a PCP request to its default router. This is important because in the case of cascaded PCP Servers, all of them need to be discovered in order of hop distance from the client. The PCP client then SHOULD send a PCP request to the WkPsdIPa. PCP Clients must be prepared to receive an error and try other discovery methods.

2.3. PCP Discovery Server behavior

PCP Server can be configured to listen on the WkPsdIPa for incoming PCP requests.

PCP responses are sent from that same IANA-assigned address (see Page 5 of [RFC1546]).

3. Deployment Considerations

Network operators should install one or more PCP Servers as specified above. Depending on the network deployment scenario they may use IP routing tables, or other suitable mechanisms to direct PCP requests to one of these servers.

[TBD: explain in more detail] This works fine even with cascaded access routers with NATs. After each router hop the operator may decide whether to handle the discovery requests, e.g., using a static routing table entry, or whether let them flow "automatically" towards the Internet backbones using the default routing table entry.

3.1. Multiple PCP Servers, Symmetric Routing

In the case of symmetric routing all inbound and outbound packets from a PCP client traverse the same PCP Server or controlled device. Multiple PCP Servers sharing an anycast address in a symmetric routing scenario are used for two purposes: ease of network configuration and redundancy. In the case of redundancy, If there is a network or routing change a PCP client might start interacting with a different PCP Server sharing the same anycast address. From a PCP Client point of view this would be the same as a PCP Server reboot and a PCP Client could find out about it by examining the Epoch field during the next PCP request or ANNOUNCE message.

3.2. Multiple PCP Servers, Assymetric Routing

In the case of asymmetric routing inbound packets from a PCP client traverse a different PCP Server or controlled device than outbound packets. If these PCP Servers are firewalls, the PCP client would need to create mappings on both of them in order to properly communicate with other hosts. But if these PCP Servers share an anycast address the PCP Client will create mappings in only on, when in fact should create mapping on both of them.

Therefore in order to support this scenario we propose a new option for the ANNOUNCE opcode. This will allow a PCP Client to request from a PCP Server a list of unicast IP addresses associated with other PCP Servers. The client can then proceed to create mappings on these PCP Servers using their unicast addresses.

This Option:

Option Name: LIST_PCP_SRVS

Number: TBA (IANA)

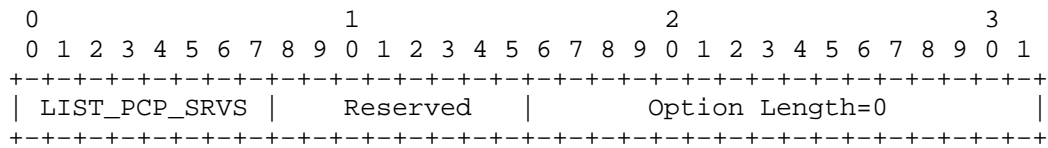
Purpose: Allows a PCP Client to request from a PCP Server a list of
all PCP Servers configured

Valid for Opcodes: ANNOUNCE

Length: 0x0

May appear in: request and reply

Maximum occurrences in request: 1



The Reply from the PCP Server would be a list of IP addresses

Length in reply: 128 bits * number of IP addresses

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

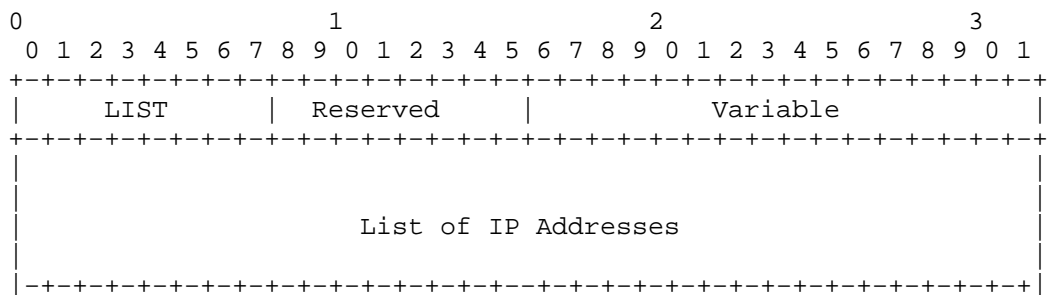


Figure 1: List of PCP Servers

4. IANA Considerations

4.1. Registration of IPv4 Special Purpose Address

IANA is requested to register a single IPv4 address in the IANA IPv4 Special Purpose Address Registry [RFC5736].

[RFC5736] itemizes some information to be recorded for all designations:

1. The designated address prefix.

Prefix: TBD by IANA. Prefix length: /32

2. The RFC that called for the IANA address designation.

This document.

3. The date the designation was made.

TBD.

4. The date the use designation is to be terminated (if specified as a limited-use designation).

Unlimited. No termination date.

5. The nature of the purpose of the designated address (e.g., unicast experiment or protocol service anycast).

protocol service anycast.

6. For experimental unicast applications and otherwise as appropriate, the registry will also identify the entity and related contact details to whom the address designation has been made.

N/A.

7. The registry will also note, for each designation, the intended routing scope of the address, indicating whether the address is intended to be routable only in scoped, local, or private contexts, or whether the address prefix is intended to be routed globally.

Typically used within a network operator's network domain, but in principle globally routable.

8. The date in the IANA registry is the date of the IANA action, i.e., the day IANA records the allocation.

TBD.

4.2. Registration of IPv6 Special Purpose Address

IANA is requested to register a single IPv6 address in the IANA IPv6 Special Purpose Address Block [RFC4773].

[RFC4773] itemizes some information to be recorded for all designations:

1. The designated address prefix.

Prefix: TBD by IANA. Prefix length: /128

2. The RFC that called for the IANA address designation.

This document.

3. The date the designation was made.

TBD.

4. The date the use designation is to be terminated (if specified as a limited-use designation).

Unlimited. No termination date.

5. The nature of the purpose of the designated address (e.g., unicast experiment or protocol service anycast).

protocol service anycast.

6. For experimental unicast applications and otherwise as appropriate, the registry will also identify the entity and related contact details to whom the address designation has been made.

N/A.

7. The registry will also note, for each designation, the intended routing scope of the address, indicating whether the address is intended to be routable only in scoped, local, or private contexts, or whether the address prefix is intended to be routed globally.

Typically used within a network operator's network domain, but in principle globally routable.

8. The date in the IANA registry is the date of the IANA action, i.e., the day IANA records the allocation.

TBD.

4.3. PCP Option

The following PCP Option should be allocated:

LIST_PCP_SRVS

5. Security Considerations

TBD

6. Acknowledgements

Ted Lemon for insightful DHCP discussions and Dave Thaler for pointing out the asymmetric routing case.

7. References

7.1. Normative References

- [RFC1546] Partridge, C., Mendez, T., and W. Milliken, "Host Anycasting Service", RFC 1546, November 1993.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2616] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999.
- [RFC2732] Hinden, R., Carpenter, B., and L. Masinter, "Format for Literal IPv6 Addresses in URL's", RFC 2732, December 1999.
- [RFC3958] Daigle, L. and A. Newton, "Domain-Based Application Service Location Using SRV RRs and the Dynamic Delegation Discovery Service (DDDS)", RFC 3958, January 2005.
- [RFC4773] Huston, G., "Administration of the IANA Special Purpose IPv6 Address Block", RFC 4773, December 2006.
- [RFC5736] Huston, G., Cotton, M., and L. Vegoda, "IANA IPv4 Special Purpose Address Registry", RFC 5736, January 2010.

7.2. Informative References

- [DhcpRequestParams]
OpenFlow, "OpenFlow Switch Specification", February 2011, <<http://msdn.microsoft.com/en-us/library/windows/desktop/aa363298%28v=vs.85%29.aspx>>.
- [I-D.chen-pcp-mobile-deployment]
Chen, G., Cao, Z., Boucadair, M., Ales, V., and L. Thiebaut, "Analysis of Port Control Protocol in Mobile Network", draft-chen-pcp-mobile-deployment-02 (work in progress), October 2012.
- [I-D.ietf-dhc-container-opt]
Droms, R. and R. Penno, "Container Option for Server Configuration", draft-ietf-dhc-container-opt-06 (work in progress), December 2012.
- [I-D.ietf-pcp-base]
Wing, D., Cheshire, S., Boucadair, M., Penno, R., and P. Selkirk, "Port Control Protocol (PCP)",

draft-ietf-pcp-base-29 (work in progress), November 2012.

[I-D.ietf-pcp-dhcp]

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

Appendix A. Problems with Other Discovery methods

Several algorithms have been specified that allows PCP Client to discover the PCP Servers on a network . However, each of this approaches has technical or operational issues that will hinder the fast deployment of PCP.

A.1. DHCP PCP Options

There are two problems with DHCP Options: DHCP Server on Home Gateways (HGW) and Operating Systems DHCP clients

Currently what the HGW does with the options it receives from the ISP is not standardized in any general way. As a matter of practice, the HGW is most likely to use its own customer-LAN-facing IP address for the DNS server address. As for other options, it's free to offer the same values to the client, offer no value at all, or offer its own IP address if that makes sense, as it does (sort of) for DNS.

In scenarios where PCP Server resides on ISP network and is intended to work with arbitrary home gateways that don't know they are being used in a PCP context, that won't work, because there's no reason to think that the HGW will even request the option from the DHCP server, much less offer the value it gets from the server on the customer-facing LAN. There is work on the DHC WG to overcome some of these limitations [I-D.ietf-dhc-container-opt] but in terms of deployment it also needs HGW to be upgraded.

The problems with Operating Systems is that even if DHCP PCP Option were made available to customer-facing LAN, host stack DHCP enhancements are required to process or request new DHCP PCP option. One exception is Windows [DhcpRequestParams]

Finally, in the case of IPv6 there are networks where there is DHCPv6 infrastructure at all or some hosts do not have a DHCPv6 client.

A.2. Default Router

If PCP server does not reside in first hop router, whether because subscriber has a existing home router or in the case of Wireless Networks (3G, LTE) [I-D.chen-pcp-mobile-deployment], trying to send a request to default router will not work.

A.3. User Input

A regular subscriber can not be expected to input IP address of PCP Server or network domain name. Moreover, user can be at a Wi-Fi hotspot, Hotel or related. Therefore relying on user input is not

reliable.

A.4. Domain Name System Based

There are three separate category of problems with NAPTR [RFC3958]

1. End Points: It relies on PCP client determining the domain name and supporting certain DNS queries
2. DNS Servers: DNS server need to be provisioned with the necessary records
3. CPEs: CPEs might interfere with DNS queries and the DHCP domain name option conveyed by ISP that could be used to bootstrap NAPTR might not be relayed to home network.

Authors' Addresses

Sebastian Kiesel
University of Stuttgart Computing Center
Allmandring 30
Stuttgart 70550
Germany

Email: ietf-alto@skiesel.de
URI: <http://www.rus.uni-stuttgart.de/nks/>

Reinaldo Penno
Cisco Systems
170 West Tasman Dr
San Jose CA
USA

Email: repenno@cisco.com

pcp
Internet-Draft
Intended status: Standards Track
Expires: April 19, 2013

R. Maglione
Telecom Italia
D. Cheng
Huawei Technologies
M. Boucadair
France Telecom
October 16, 2012

RADIUS Extensions for Port Control Protocol
draft-maglione-pcp-radius-ext-05

Abstract

This memo specifies a new Remote Authentication Dial In User Service (RADIUS) attribute to carry one or a list of Port Control Protocol (PCP) Server Names. This attribute can be configured on a RADIUS server so that the information can be conveyed to Network Access Server (NAS) via RADIUS protocol, and the co-located Dynamic Host Configuration Protocol (DHCP/DHCPv6) server can then populate the information to PCP client.

Status of this Memo

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

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

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

This Internet-Draft will expire on April 19, 2013.

Copyright Notice

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

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

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

Table of Contents

1. Introduction	3
2. Terminology	4
3. PCP Server Configuration using RADIUS and DHCPv4/DHCPv6	4
4. RADIUS Attribute	8
5. Table of attributes	9
6. Security Considerations	10
7. IANA Considerations	10
8. Acknowledgments	10
9. References	10
9.1. Normative References	10
9.2. Informative References	11
Authors' Addresses	11

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 Customer Premises Equipment (CPE), etc., which communicates with a PCP server that may reside anywhere in a network.

[I-D.ietf-pcp-base] defines a procedure for the PCP client to communicate with its PCP Server. The IP address of the PCP Server(s) can be configured to the PCP Client; if not the PCP Client assumes its default router as being its PCP Server.

[I-D.ietf-pcp-dhcp] defines DHCPv6 and DHCPv4 options which are meant to be used by a PCP client to discover a PCP server name. However, provisioning for name of the PCP server is required on a DHCPv4/DHCPv6 server before it can populate this information.

Auto-configuration on a DHCPv4/DHCPv6 is possible in a broadband network, where typically, user profile is maintained on a Remote Authentication Dial In User Service (RADIUS) server and RADIUS protocol [RFC2865] is used to convey user-related information to other network elements including a host and CPE.

[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 DHCPv4/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 one or a list of PCP server names. As defined in [I-D.ietf-pcp-dhcp], a PCP Server Name can be a DNS name, IP literals strings, etc. This document is designed to allow for configuring PCP Server name(s) which can be a DNS name, IP literals or any strings which may be passed to a local name resolution library on the PCP client side. Operational considerations related to the configuration of PCP Server names are similar to those discussed in Section 4 and Section 7 of [I-D.ietf-pcp-dhcp]. The proposed RADIUS option is designed to accommodate various deployment contexts (e.g., dedicated option per IP connectivity context, single option for dual-stack access, etc.).

The approach described above is already used for providing the FQDN of the AFTR in the DS-Lite scenario [RFC6333] and the equivalent RADIUS attribute for the DS-Lite Tunnel Name is defined [RFC6519].

2. Terminology

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

The following terms are defined in [I-D.ietf-pcp-base]:

- Port forwarding
- PCP
- PCP client
- PCP Server

The following term is defined in [I-D.ietf-pcp-dhcp]:

- PCP Server Name

3. PCP Server Configuration using RADIUS and DHCPv4/DHCPv6

Figure 1 illustrates an example of how RADIUS protocol works together with DHCPv6, to allow a host to learn automatically the name of a PCP server in case of a PPP session that carries IPv6 traffic.

The Network Access Server (NAS) operates as a client of RADIUS and co-locates with a DHCPv6 Server for DHCPv6. The NAS initially sends a RADIUS Access Request message to the RADIUS server, requesting authentication. Once the RADIUS server receives the request, it validates the sending client and if the request is approved, the RADIUS server replies with an Access Accept message including a list of attribute-value pairs that describe the parameters to be used for this session. This list MAY also contain the name of a PCP server. When the co-located DHCPv6 server receives a DHCPv6 message containing the PCP Server Option, it SHALL use the name returned in the RADIUS attribute as defined in this memo to populate the DHCPv6 PCP Server option defined in [I-D.ietf-pcp-dhcp]

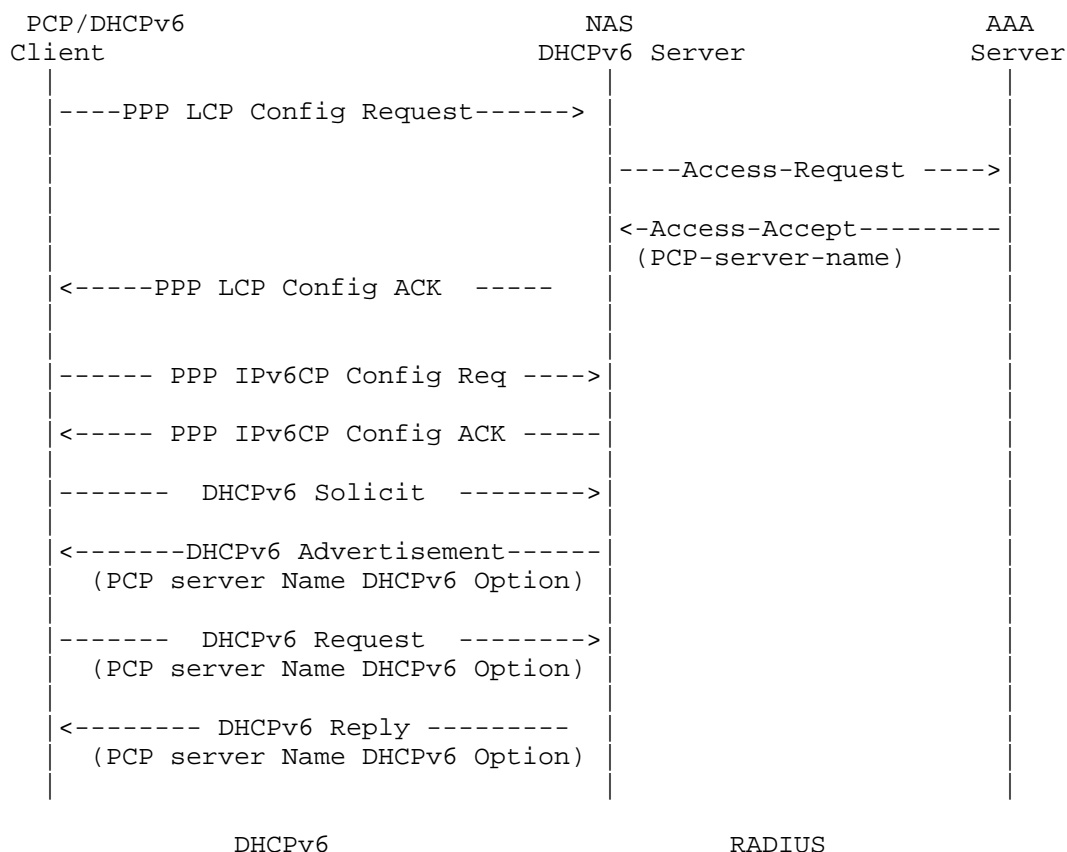


Figure 1: RADIUS and DHCPv6 Message Flow for a PPP Session

The Figure 2 illustrates how the RADIUS protocol and DHCPv6 work together to accomplish PCP client configuration when DHCPv6 is used to provide connectivity to a requesting host.

The difference between this message flow and previous one is that in this scenario the interaction between NAS and AAA/ RADIUS Server is triggered by the DHCPv6 Solicit message received by the NAS from the DHCPv6 client, while in case of a PPP Session the trigger is the PPP LCP Config Request message received by the NAS.

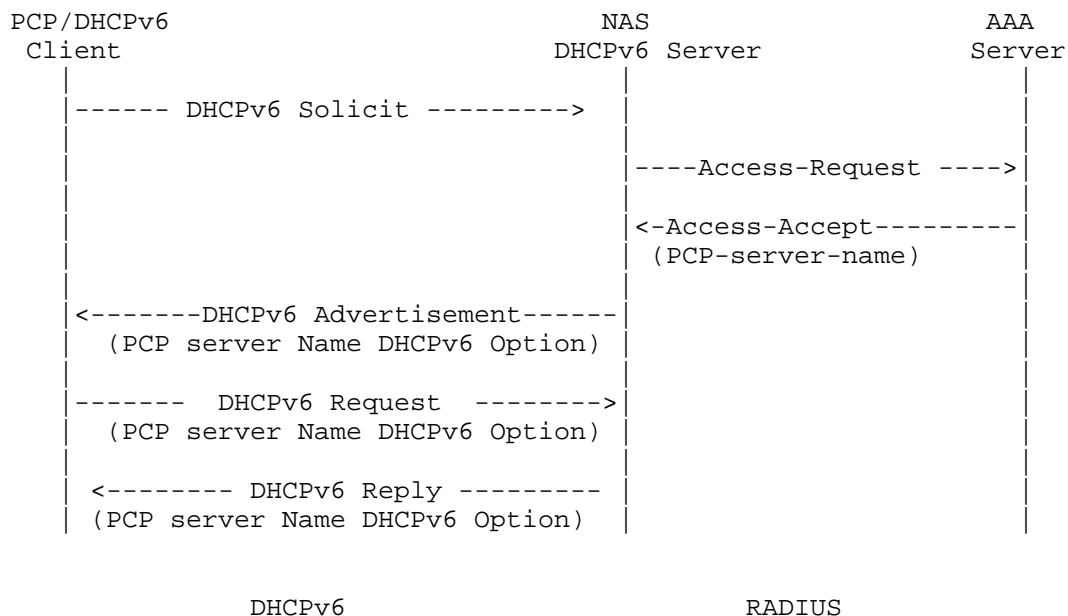


Figure 2: RADIUS and DHCPv6 Message Flow for an IP Session

In the scenario depicted in Figure 2 the Access-Request packet contains a Service-Type attribute with the value Authorize Only (17), thus according to [RFC5080] the Access-Request packet MUST contain a State attribute.

A similar message flow also applies to the IPv4 scenario when DHCPv4 is used to provide connectivity to the user (Figure 3).

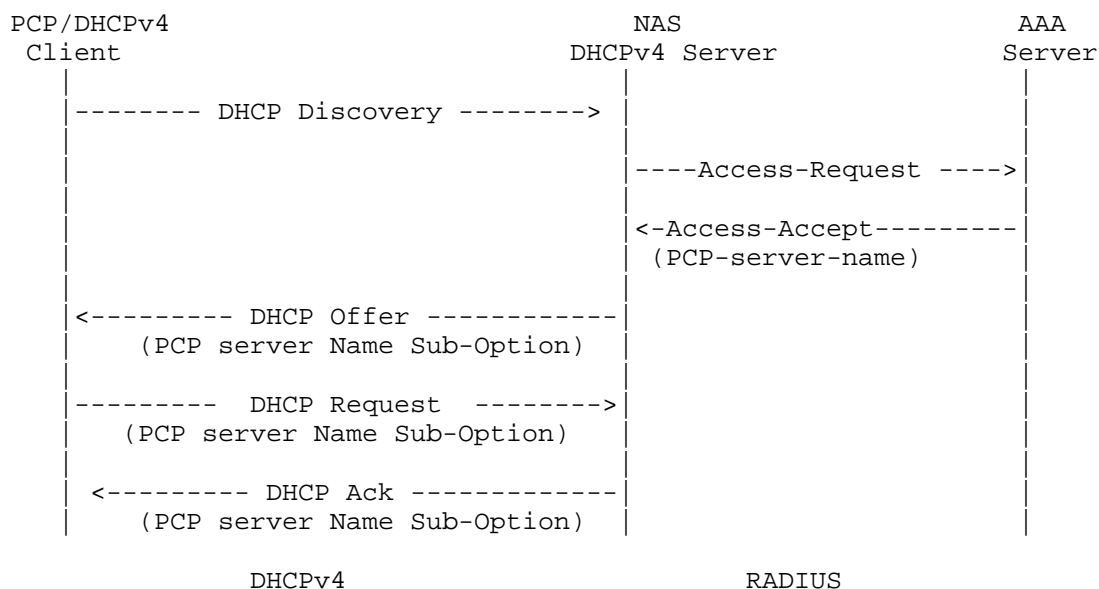


Figure 3: RADIUS and DHCPv4 Message Flow for an IP Session

After receiving the PCP server name in the initial Access-Accept the NAS MUST store the received PCP Server Name locally. When the PCP Client sends a DHCPv4 message to request an extension of the lifetimes for the assigned address or prefix, the NAS does not have to initiate a new Access-Request towards the AAA server to request the PCP server name. The NAS retrieves the previously stored PCP Server name and uses it in its reply.

If the DHCPv4 server to which the DHCP Renew message was sent at time T1 has not responded, the DHCPv4 client initiates a Rebind/Reply exchange with any available server. In this scenario the NAS MUST initiate a new Access-Request towards the AAA server, after the co-located DHCPv4 server receives the DHCP message. The NAS MAY include the PCP Server Name attribute in its Access-Request.

If the NAS does not receive the PCP server name attribute in the Access-Accept it MAY fallback to a pre-configured default tunnel name, if any. If the NAS does not have any pre-configured default tunnel name or if the NAS receives an Access-Reject, the PCP client can not be configured by the NAS.

The scenario with PPP Session and IPv4 only connectivity does not require DHCPv4: the whole configuration of the client is performed by PPP. This case is out of scope of this document because in order to complete the configuration of the PCP client a new PPP IPC option

would be required.

4. RADIUS Attribute

A new RADIUS attribute, called PCP-Server-Name, along with its format is defined below.

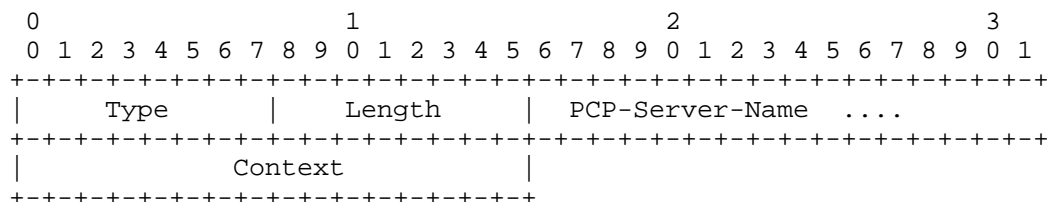
Description

The PCP-server-name attribute contains a name or a list of names 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 Name DHCP Sub-Option in IPv4 addressing context, or the PCP Server Name DHCPv6 Option in IPv6 addressing context, as determined by the DHCP server [I-D.ietf-pcp-dhcp]. The same or distinct PCP Server Names may be configured; it is out of scope of this document to elaborate on this point. Nevertheless, the PCP-server-name attribute conveys an indication for the deployment context.

The PCP-server-name attribute MAY appear in an Access-Accept packet. This attribute MAY be used in Access-Request packets as a hint to the RADIUS server; for example if the NAS is pre-configured with a default PCP server name, this name MAY be inserted in the attribute. The RADIUS server MAY ignore the hint sent by the NAS and it MAY assign a different PCP Server name. If the NAS includes the PCP Server Name attribute, but the AAA server does not recognize it, this attribute MUST be ignored by the AAA Server. If the NAS does not receive PCP Server Name attribute in the Access-Accept it MAY fallback to a pre-configured default PCP server name, if any. If the NAS is pre-provisioned with a default PCP server name and the PCP server name received in Access-Accept is different from the configured default, then the PCP server name received in the Access-Accept message MUST be used for the session.

The PCP server Name RADIUS attribute MAY be present in Accounting-Request records where the Acct-Status-Type is set to Start, Stop or Interim-Update. The PCP Server Name RADIUS attribute MUST NOT appear more than once in a message.

A summary of the PCP-Server-Name RADIUS attribute format is shown below. The fields are transmitted from left to right.



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:

One or a list of PCP Server Name(s). The domain name is encoded as specified in [I-D.ietf-pcp-dhcp]

Context:

This field indicates the IP connectivity context:

0: Dual-Stack. The same option is provided for both DHCPv4 and DHCPv6 requesting hosts

1: This option is provided for DHCPv4 requesting hosts

2: This option is provided for DHCPv6 requesting hosts

The data type of PCP Server Name is a string with opaque encapsulation, according to section 2.1 of [RFC6158]

5. Table of attributes

The following table provides a guide to which attributes may be found in which kinds of packets, and in what quantity.

Request	Accept	Reject	Challenge	Accounting	#	Attribute
				Request		
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. References

9.1. Normative References

- [I-D.ietf-pcp-base]
Wing, D., Cheshire, S., Boucadair, M., Penno, R., and P. Selkirk, "Port Control Protocol (PCP)", draft-ietf-pcp-base-28 (work in progress), October 2012.
- [I-D.ietf-pcp-dhcp]
Boucadair, M., Penno, R., and D. Wing, "DHCP Options for the Port Control Protocol (PCP)", draft-ietf-pcp-dhcp-05 (work in progress), September 2012.
- [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.

- [RFC5080] Nelson, D. and A. DeKok, "Common Remote Authentication Dial In User Service (RADIUS) Implementation Issues and Suggested Fixes", RFC 5080, December 2007.
- [RFC6158] DeKok, A. and G. Weber, "RADIUS Design Guidelines", BCP 158, RFC 6158, March 2011.
- [RFC6519] Maglione, R. and A. Durand, "RADIUS Extensions for Dual-Stack Lite", RFC 6519, February 2012.

9.2. Informative References

- [I-D.ietf-radext-ipv6-access]
Dec, W., Sarikaya, B., Zorn, G., Miles, D., and B. Lourdelet, "RADIUS attributes for IPv6 Access Networks", draft-ietf-radext-ipv6-access-11 (work in progress), August 2012.
- [RFC6333] Durand, A., Droms, R., Woodyatt, J., and Y. Lee, "Dual-Stack Lite Broadband Deployments Following IPv4 Exhaustion", RFC 6333, August 2011.

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:

Mohamed Boucadair
France Telecom
Rennes, 35000
France

Phone:
Fax:
Email: mohamed.boucadair@orange.com
URI:

Port Control Protocol
Internet-Draft
Intended status: Standards Track
Expires: July 25, 2013

R. Penno
D. Wing
Cisco
M. Boucadair
France Telecom
January 21, 2013

PCP Support for Nested NAT Environments
draft-penno-pcp-nested-nat-03

Abstract

Nested NATs or multi-layer NATs are already widely deployed. They are characterized by two or more NAT devices in the path of packets from the subscriber to the Internet. Moreover, NAT devices currently deployed are PCP unaware and it is assumed that NAT aware PCP devices will take a long time to be rolled out. Therefore in order to lower the adoption barrier of PCP and make it work for currently deployed networks, this document proposes a few mechanisms for PCP-enabled applications to work through NATs with varying levels of PCP protocol support.

Requirements Language

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

Status of this Memo

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

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

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

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

Copyright Notice

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

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

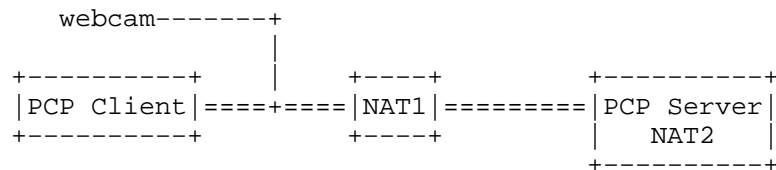
Table of Contents

1. Introduction	3
1.1. Terminology	3
1.2. Problem Statement	3
1.3. Scope	4
2. PCP MAP Nested NAT Methods	4
2.1. PCP and UPnP unaware Intermediate NATs	5
2.2. PCP Server intermediate NAT	7
2.3. UPnP enabled intermediate NAT	8
2.4. PCP Proxy Intermediate NAT	8
2.4.1. PCP Proxy Discovery	9
3. PCP PEER Nested NAT Methods	9
3.1. Send-then-connect	9
3.2. Connect-then-send	10
4. RECEIVED_SOURCE_IP_PORT Option	10
5. SCOPE Option	11
6. IANA Considerations	12
7. Security Considerations	13
8. Acknowledgements	13
9. References	13
9.1. Normative References	13
9.2. Informative References	13
Authors' Addresses	13

1. Introduction

Nested NATs are widely deployed and come in different topology flavors. It could be a home subscriber which has an ISP provided NAT CPE chained with another personal NAT router. It could be an ISP provided CPE chained with a CGN.

An example of the use of the proposed options is illustrated in the following figure where there is a NAT in the path between the PCP Client and the PCP Server.



An example of instructing mappings in the PCP Server is as follows:

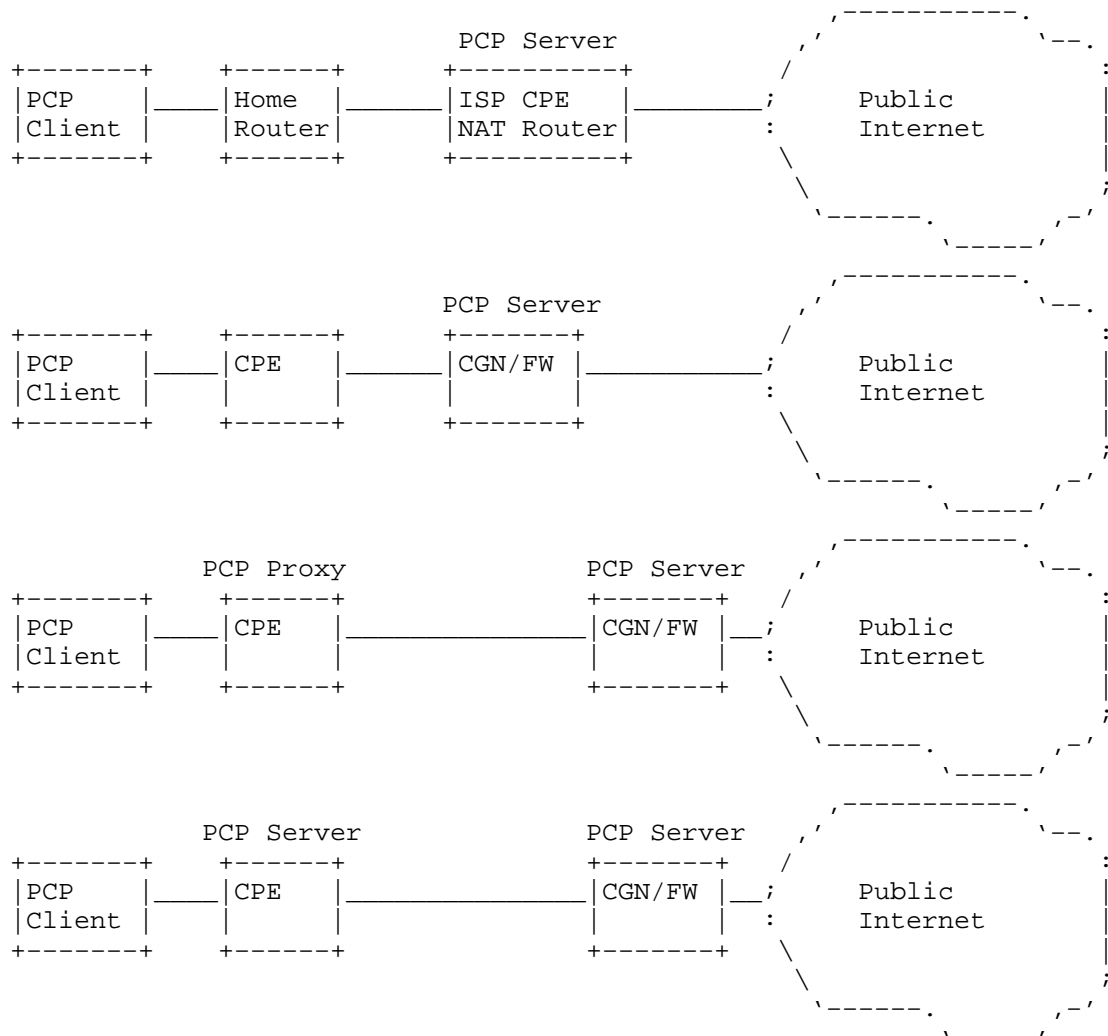
- o NAT1 is detected in the path between the PCP Client and the PCP Server owing to the use of the RECEIVED_SOURCE_IP_PORT Option and the returned IP address (IP Header) of PCP request in PCP response;
- o After learning about that NAT, the PCP Client uses UPnP IGD, NAT-PMP or manual configuration to interact with NAT1 and open the necessary port on NAT1 (e.g., IP address= IPx, port=X);
- o The PCP Client then sends PCP message to the PCP Server, indicating IPx and X as the internal IP address and port. The PCP Server opens pinhole towards IPx and X.

1.1. Terminology

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

1.2. Problem Statement

The current NAT deployed devices will take years to be replaced or upgraded to become PCP aware. Moreover, nested NATs are common and come in a variety of flavors (examples below). Therefore, as applications become PCP enabled, it is important that they can work through nested NAT networks as is, without requiring infrastructure changes. From the point of view of a PCP-enabled application running on an end host, the core problem is common across different nested NAT topologies: how to install PCP mappings in a nested NAT scenario where the different NATs in the path have varying level of PCP protocol support.



1.3. Scope

This proposal considers the discovery of the PCP Server out of scope. Nonetheless, it is a critical piece of PCP deployment in service provider networks.

2. PCP MAP Nested NAT Methods

There are a few methods to make PCP work through nested NATs. They differ mainly based on the level of support that can be expected from

intermediate NATs, which can be:

- o PCP and UPnP unaware or disabled
- o PCP Server
- o UPnP Server
- o PCP Proxy

The next sections discuss each scenario on the basis of protocol support on intermediate NATs.

2.1. PCP and UPnP unaware Intermediate NATs

This method will most likely be used by PCP clients in nested NAT environments while PCP Proxy support is not ubiquitous. It assumes no UPnP or PCP Proxy support on intermediate NATs. This proposal leverages the current behavior of PCP [I-D.ietf-pcp-base] which allows a PCP Client and Server to detect intervening nested NATs. The PCP Server uses the information on the outer IP and PCP headers to detect and install a proper NAT mapping and return the source IP:port from the IP header on the PCP response. It does not assume any change to current deployed NATs.

1. The PCP Client sends the MAP request as it normally would without any changes.
2. As the message goes through one (or more) PCP-unaware NAT, the source IP:port of the IP header will change accordingly
3. The PCP Server compares the PCP Client IP:port in the PCP header with the source IP:port of the IP header
4. If these are different, the server knows that the PCP message went through a PCP-unaware NAT. Therefore it installs a mapping directed to the source IP address found on the IP header and internal port of the PCP header.

s/dport: source/destination port
 s/dIP : source/destination IP
 PCP-C : PCP client
 iport : Internal port
 PCP-U : PCP Unaware NAT
 E-port : External port
 E-IP : External IP

PCP Client

PCP-U NAT

PCP Server

<pre> Map request Outer sIP:192.168.0.2 Outer sPort:19216 PCP-C Addr:192.168.0.2 PCP-C port:19216 iPort:40000 -----> Map response Outer dIP:192.168.0.2 Outer dport:19216 Assigned E-port:20001 Assigned E-IP:20.0.0.1 PCP-C Addr:10.0.0.2 PCP-C port:10002 <----- </pre>	<pre> Map request Outer sIP:10.0.0.2 Outer sPort:10002 PCP-C Addr:192.168.0.2 PCP-C port:19216 iPort:40000 -----> PCP client IP != Outer IP Allocate public IP and port Mapping: (10.0.0.2, 40000) <- (20.0.0.1, 20001) Map response Outer dIP:10.0.0.2 Outer dport:10002 Assigned E-port:20001 Assigned E-IP:20.0.0.1 PCP-C Addr:10.0.0.2 PCP-C port:10002 <----- </pre>	<pre> </pre>
--	--	--------------

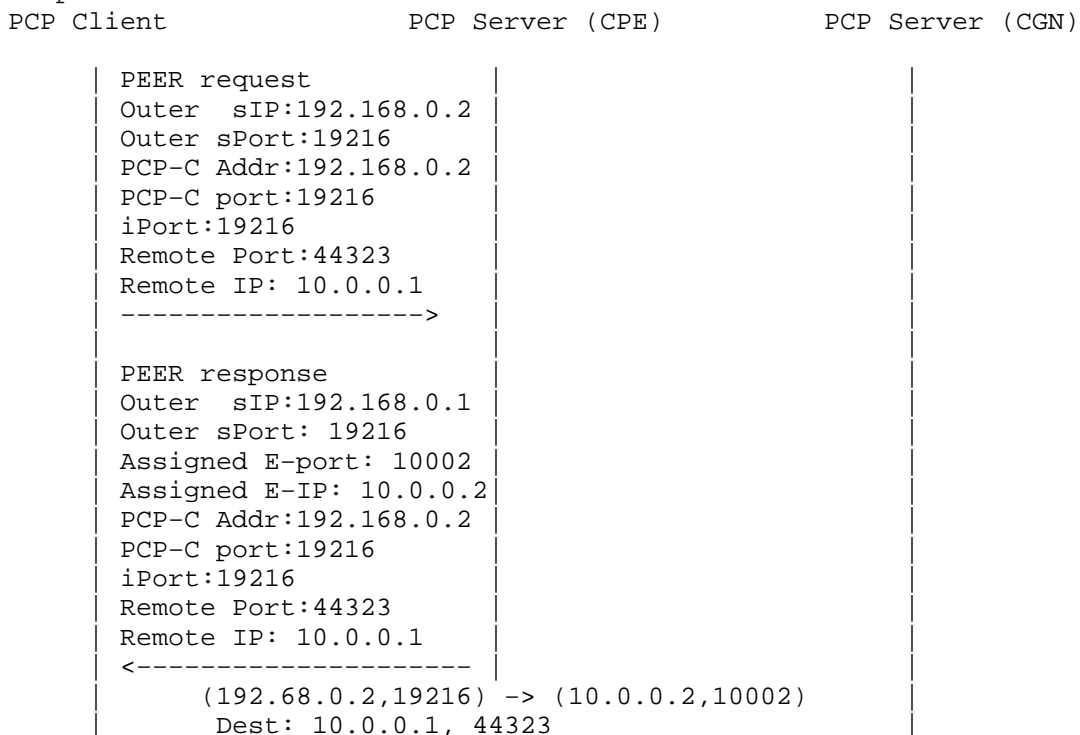
- Subscriber installs a port forwarding or DMZ entry on its home CPE (PCP U-NAT) through manual configuration. The entry would be (*, 40000) -> (10.0.0.1, 40000). Alternatively the application could use UPnP for the same purpose.

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 Address does not match the IP address of IP header, PCP Server (CGN) will reject request with ADDRESS_MISMATCH error. Therefore PCP Client first needs to know the IP address and port the CPE NAT will use for the actual PCP request to CGN.

If the PCP client relies on nested NAT detection the first step is not needed. It is assumed that before sending 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.



<pre> Map request Outer sIP:192.168.0.2 Outer sPort:19216 PCP-C Addr:10.0.0.2 PCP-C port:10002 iPort:40000 -----> </pre>	<pre> Map request Outer sIP:10.0.0.2 Outer sPort:10002 PCP-C Addr:10.0.0.2 PCP-C port: 10002 iPort:40000 -----> </pre>	<pre> (10.0.0.2, 40000) <- (20.0.0.1, 20001) </pre>
<pre> Map response Outer dIP:192.168.0.2 Outer dport:19216 Assigned E-port: 20001 Assigned E-IP: 20.0.0.1 PCP-C Addr: 10.0.0.2 PCP-C port: 10002 <----- </pre>	<pre> Map response Outer dIP:10.0.0.2 Outer dport: 10002 Assigned E-port: 20001 Assigned E-IP: 20.0.0.1 PCP-C Addr: 10.0.0.2 PCP-C port: 10002 <----- </pre>	

2.3. UPnP enabled intermediate NAT

This scenario is very similar to the PCP Server intermediate NAT, but the CPE implements a UPnP Server instead of PCP Server. The mechanics are the same with the difference that first PEER message to setup the PCP Control messages mapping is substituted by its UPnP equivalent.

2.4. PCP Proxy Intermediate NAT

This method assumed that the intermediate NATs implement a PCP Proxy function. There are two non-exclusive types of proxy functions: interception (ALG) and server-client based. In the interception case the PCP Proxy intercepts PCP messages destined to a PCP Server downstream, modifies IP, UDP and PCP headers, allocates a mapping and send them to the downstream PCP Server. Ideally if the interception PCP Proxy also implements a PCP server it would let the PCP Client

know of its existence in a PCP response through an option (TBD) and henceforth the PCP Client would start directing messages to it.

In the server-client scenario the PCP Client sends PCP messages to the proxy which acts as both PCP Server and Client. This proxy in turn will terminate the PCP request and generate a new one acting as a PCP Client to its own PCP Server. Therefore mappings are installed in all NAT devices in a recursive manner. This is the recommended method since it does not need a special discovery procedure and works with any number of NATs. More information about this method can be found in [I-D.bpw-pcp-proxy].

2.4.1. PCP Proxy Discovery

TBD

3. PCP PEER Nested NAT Methods

All techniques discussed for PCP MAP methods do not work for PCP PEER messages. PCP PEER is a different beast and another set of techniques need to be used to overcome intervening NATs. The critical issue related to PEER is that the client needs to know the external source port NAT1 will use to translate packets for the actual data session. There are two scenarios to consider: send-then-connect and connect-then-send.

3.1. Send-then-connect

In this scenario the client sends a PEER message to install a mapping which later will be used by a regular UDP or TCP data session. In order for this to work reliably, the following procedure needs to be followed:

1. PCP Client needs to allocate a binding on the intervening NAT through STUN, UPnP or other method. Let's suppose this binding is (192.168.0.2, 19216 <-> 10.0.0.2, 10002).
2. PCP Client constructs a PCP PEER request like the following
 - * Internal port: 10002
 - * Remote Peer Port: 20002 (upcoming data connection destination port)
 - * Remote Peer address: 20.0.0.2 (upcoming data connection destination IP address)

- * PCP Client Address: 10.0.0.2

- * Protocol: TCP

3. Application will connect to remote peer using the same source IP address and port of the existing mapping on the intervening NAT. If the intervening NAT supports Protocol Independent Endpoint Independent Mapping (PI-EIM) [I-D.penno-behave-rfc4787-5382-5508-bis] , it will allocate the same external IP:port of step 1 to the new connection. Therefore 5-tuple of the new connection will match those of the previously installed PEER map.

3.2. Connect-then-send

If the data connection to the remote peer is established before the PEER message, the challenge for the PCP client is to find out which source IP:port the intervening NAT is using to translate the data packets. If the PCP Client has the necessary permissions to reuse the socket used by the data connection and the intervening NAT support EIM, two solutions are possible:

1. PCP Client sends a request from the same source IP:port as the data connection. Since the intervening NAT supports PI-EIM, it should allocate the same external IP:port of the data connection, which would be returned in the `RECEIVED_PORT_OPTION`. The PCP Client then can send an appropriate PEER message to take over the data connection. The advantage of this solution is that is built around a single protocol, PCP, and the disadvantage is that it requires a PCP extension.
2. If the PCP Client is also a STUN client it can send a binding request from the same source IP:port as the data connection and since the intervening NAT supports EIM the client will find out the external IP:port that is used to translate data packets. The PCP Client then can send an appropriate PEER message to take over the data connection. The advantage of this solution is that no extensions to PCP are needed. The disadvantage is that STUN client and server are needed, specially the fact that in case of nested NATs the STUN server needs to be located between NAT1 and NAT2.

4. `RECEIVED_SOURCE_IP_PORT` Option

This option (Code TBA, Figure 1) is used by a PCP Server to indicate in a PCP response the source IP and 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 IP:port of the received PCP request to set the Received Port.

This Option:

Option Name: PCP Received Port Option (RECEIVED_SOURCE_IP_PORT)

Number: TBA (IANA)

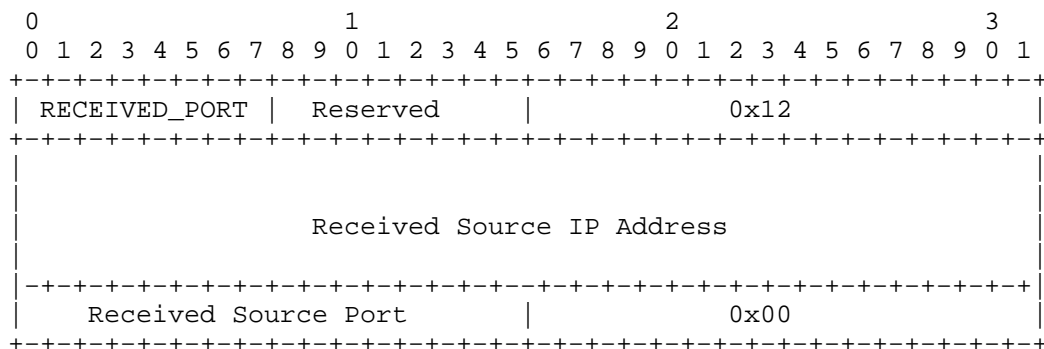
Purpose: Detect the presence of a NAT in the path and discover externally allocate IP:port

Valid for Opcodes: MAP and PEER

Length: 0x12

May appear in: both request and response

Maximum occurrences: 1



Received Source Port and IP: The source IP:port number of the received PCP request as seen by the PCP Server.

Figure 1: Received IP address/port PCP option

5. 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

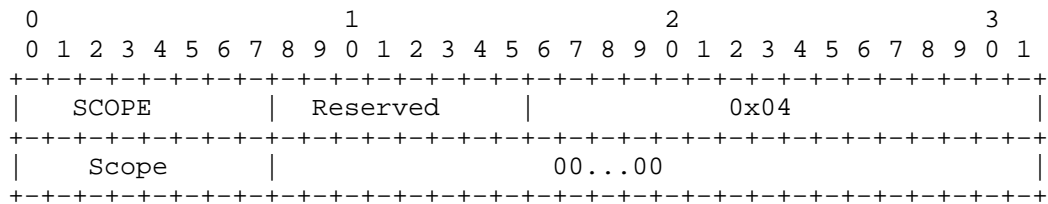


Figure 2: Scope Option

6. IANA Considerations

The following PCP Option Codes are to be allocated:

RECEIVED_PORT

SCOPE

7. Security Considerations

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

8. Acknowledgements

Thanks to Linda (wang.cuil@zte.com.cn) for her review.

9. References

9.1. Normative References

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

9.2. 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.penno-behave-rfc4787-5382-5508-bis]
Penno, R., Perreault, S., Kamiset, S., Boucadair, M., and K. Naito, "Network Address Translation (NAT) Behavioral Requirements Updates",
draft-penno-behave-rfc4787-5382-5508-bis-04 (work in progress), January 2013.

Authors' Addresses

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

Email: repenno@cisco.com

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

Email: dwing@cisco.com

Mohamed Boucadair
France Telecom
Rennes, 35000
France

Email: mohamed.boucadair@orange.com

PCP
Internet-Draft
Intended status: Standards Track
Expires: July 25, 2013

T. Reddy
Cisco
M. Isomaki
Nokia
D. Wing
P. Patil
Cisco
January 21, 2013

Optimizing NAT and Firewall Keepalives Using Port Control Protocol (PCP)
draft-reddy-pcp-optimize-keepalives-01

Abstract

This document describes how Port Control Protocol is useful to reduce NAT and firewall keepalive messages for a variety of applications.

Status of this Memo

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

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

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

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

Copyright Notice

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

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

described in the Simplified BSD License.

Table of Contents

1. Introduction	3
2. Notational Conventions	3
3. Overview of Operation	3
3.1. Application Scenarios	3
3.2. NAT and Firewall Topologies and Detection	5
3.3. Detect PCP Unaware Firewalls	7
3.4. Keepalive Optimization	7
4. Keepalive Interval Determination Procedure when PCP unaware Firewall or NAT is detected	7
5. Application-Specific Operation	9
5.1. SIP	9
5.2. HTTP	9
5.3. Media and data channels with ICE	10
5.4. Detecting Flow Failure	11
5.5. Firewalls	11
5.5.1. IPv6 Network with Firewalls	11
5.5.2. Mobile Network with Firewalls	12
6. IANA Considerations	12
7. Security Considerations	12
8. Acknowledgements	12
9. Change History	12
9.1. Changes from draft-reddy-pcp-optimize-keepalives-00	12
10. References	13
10.1. Normative References	13
10.2. Informative References	13
Appendix A. Example PHP script	14
Authors' Addresses	14

1. Introduction

Many types of applications need to keep their Network Address Translator (NAT) and Firewall (FW) mappings alive for long periods of time, even when they are otherwise not sending or receiving any traffic. This is typically done by sending periodic keep-alive messages just to prevent the mappings from expiring. As NAT/FW mapping timers may be short and unknown to the endpoint, the frequency of these keep-alives may be high. An IPv4 or IPv6 host can use the Port Control Protocol (PCP)[I-D.ietf-pcp-base] to flexibly manage the IP address and port mapping information on NATs and FWs to facilitate communications with remote hosts. This document describes how PCP can be used to reduce keep-alive messages for both client-server and peer-to-peer type of communication.

The mechanism described in this document is especially useful in cellular mobile networks, where frequent keep-alive messages make the radio transition between active and power-save states causing signaling congestion. The excessive time spent on the active state due to keep-alives also greatly reduces the battery life of the cellular connected devices such as smartphones or tablets. Requirement #14 in [I-D.binet-v6ops-cellular-host-reqs-rfc3316update] explains that cellular host SHOULD support of PCP as a driver to save battery consumption exacerbated by keepalive messages.

2. Notational Conventions

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

This note uses terminology defined in [RFC5245] and [I-D.ietf-pcp-base] .

3. Overview of Operation

3.1. Application Scenarios

PCP can help both client-server and peer-to-peer applications to reduce their keep-alive rate. The relevant applications are the ones that need to keep their NAT/FW mappings alive for long periods of time, for instance to be able to send or receive application messages in both directions at any time.

A typical client-server scenario is depicted in Figure 1. A client, who may reside behind one or multiple layers of NATs/FWs, opens a

connection to a globally reachable server, and keeps it open to be able to receive messages from the server at any time. The connection may be a connection-oriented transport protocol such as TCP or SCTP or connection-less transport protocol such as UDP. Protocols operating in this manner include Session Initiation Protocol (SIP) [RFC3261], Extensible Messaging and Presence Protocol (XMPP) [RFC3921], Internet Mail Application Protocol (IMAP) [RFC2177] with its IDLE command, the WebSocket protocol and the various HTTP long-polling protocols. There are also a number of proprietary instant messaging, Voice over IP, e-mail and notification delivery protocols that belong in this category. All of these protocols aim to keep the client-server connection alive for as long as the application is running. When the application has otherwise no traffic to send, specific keep-alive messages are sent periodically to ensure that the NAT/FW state in the middle does not expire. The client can use PCP to keep the required mapping at the NAT/FW and use application keep-alives to keep the state on the Application Server/Peer as mentioned in Section 3.4.

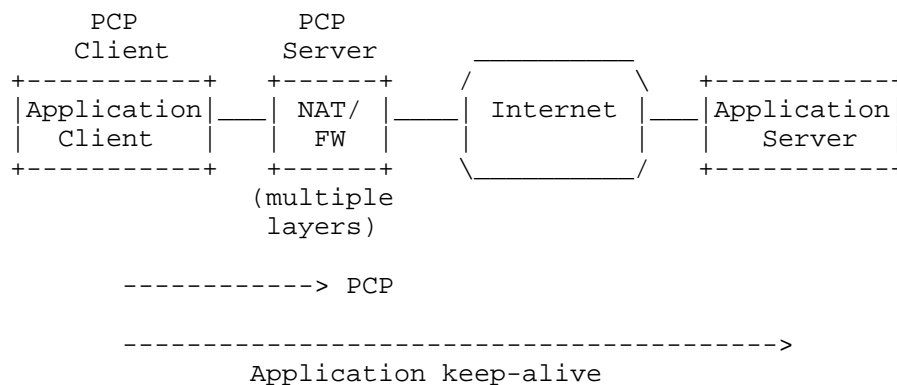


Figure 1: PCP with Client-Server applications

There are also scenarios where the long-term communication association is between two peers, both of whom may reside behind one or more layers of NAT/FW. This is depicted in Figure 2. The initiation of the association may have happened using mechanisms such as Interactive Communications Establishment (ICE), perhaps first triggered by a "signaling" protocol such as SIP or XMPP or RTCWeb. Examples of the peer-to-peer protocols include RTP and RTCWeb data channel. A number of proprietary VoIP or video call or streaming or file transfer protocols also exist in this category. Typically the communication is based on UDP, but TCP or SCTP may be used. Unless

there is no traffic flowing otherwise, the peers have to inject periodic keep-alive packets to keep the NAT/FW mappings on both sides of the communication active. Instead of application keep-alives, both peers can use PCP to control the mappings on the NAT/FWs to reduce the keep-alive frequency as explained in Section 3.4.

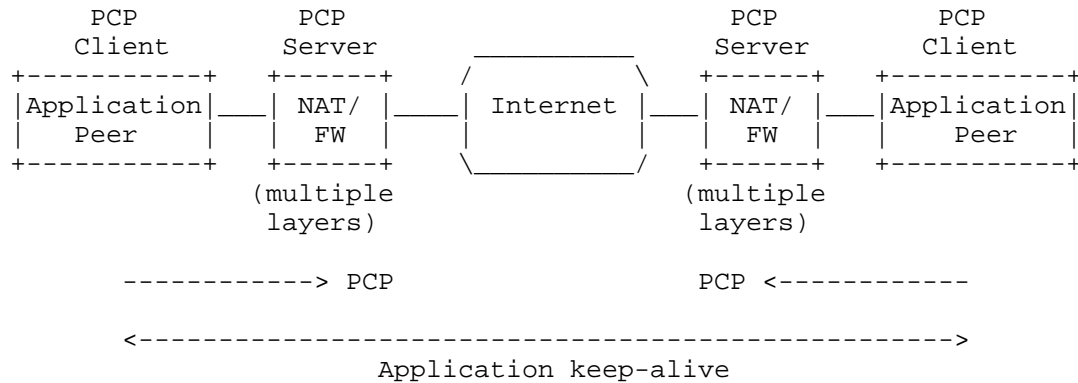


Figure 2: PCP with Peer-to-Peer applications

3.2. NAT and Firewall Topologies and Detection

Before an application can reduce its keep-alive rate, it has to make sure it has all of the NATs and Firewalls on its path under control. This means it has to detect the presence of any PCP-unaware NATs and Firewalls on its path. PCP itself is able to detect unexpected NATs between the PCP client and server as depicted in Figure 3. The PCP client includes its own IP address and UDP port within the PCP request. The PCP server compares them to the source IP address and UDP port it sees on the packet. If they differ, there are one or more additional NATs between the PCP client and server, and the server will return an error. Unless the application has some other means to control these PCP unaware NATs, it has to fall back to its default keep-alive mechanism.

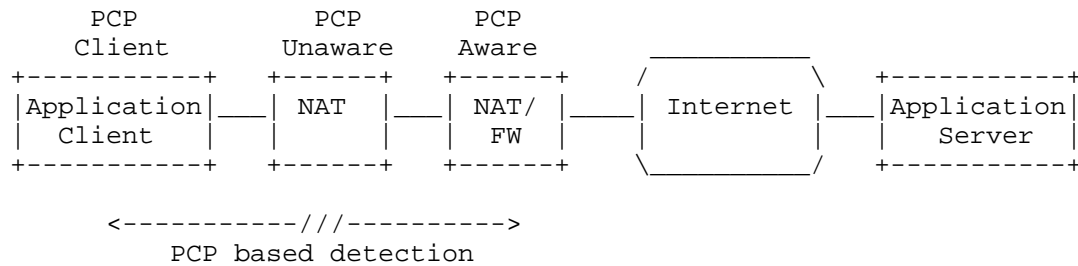


Figure 3: PCP unaware NAT between PCP client and server

Figure 4 shows a topology where one or more PCP unaware NATs are deployed on the exterior of the PCP capable NAT/FWs. To detect this, the application must have the capability to request from its server or peer what IP and transport address it sees. If those differ from the IP and transport address given to the application by the out most PCP aware NAT/FW, the application can detect that there is at least one more PCP unaware NAT on the path. In this case, the application has to fall back to its default keep-alive mechanism.

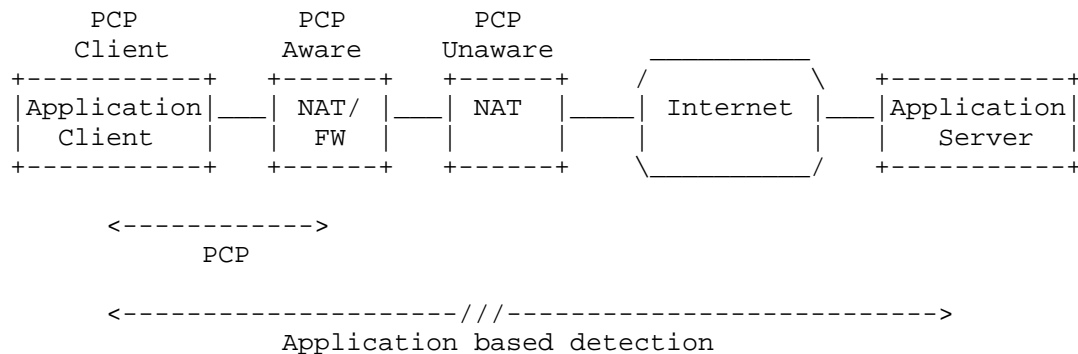


Figure 4: PCP unaware NAT external to the last PCP aware NAT

Section 5 describes how the detection works in a number of real application protocols.

The caveat is that Firewalls can not be detected this way. The client will have to use the alternative procedure explained in Section 3.3 to detect PCP unaware Firewalls.

3.3. Detect PCP Unaware Firewalls

The client sends a STUN Binding Request to the STUN server. STUN server will return its alternate IP address and alternate port in OTHER-ADDRESS in the binding response [RFC5780]. The client then sends MAP request with FILTER option to PCP server to permit STUN server to reach the client using the STUN servers alternate IP address and alternate port. The client then sends a binding request to the primary address of the STUN server with the CHANGE-REQUEST attribute set to change-port and change-IP. This will cause the server to send its response from its alternate IP address and alternate port. If the client receives a response then the client is aware that on path Firewall devices are PCP aware. If the client does not receive a response then the client is aware that could be one or more on path PCP unaware Firewall devices. PCP client will perform the tests separately for each transport protocol. If no response is received, the client will then repeat the test atmost three times for connectionless transport protocols.

If the STUN server does not support OTHER-ADDRESS then this test cannot be run. This procedure can be adopted by other protocols to detect PCP unaware Firewalls.

3.4. Keepalive Optimization

If the application determines that all NATs and Firewalls on its path to the Internet support PCP, it can start using PCP instead of its default keep-alives to maintain the NAT/FW state. It can use PCP PEER Request with the Requested Lifetime set to an appropriate value. The application may still send some application-specific heartbeat messages end-to-end.

Processing the lifetime value of the PEER Opcode is described in Section 15 of [I-D.ietf-pcp-base]. Sending a PEER request with a very short Requested Lifetime can be used to query the lifetime of an existing mapping. PCP recommends that lifetimes of mapping created or lengthened with PEER be longer than the lifetimes of implicitly-created NAT and Firewall mappings. Thus PCP can be used to save battery consumption by making PCP PEER message interval longer than what the application would normally use the keep middle box state alive, and strictly shorter than the server state refresh interval.

4. Keepalive Interval Determination Procedure when PCP unaware Firewall or NAT is detected

If PCP unaware NAT/Firewall is detected then a client can use the following heuristics method to determine the keepalive interval :

1. The client sends a STUN Binding Request to the STUN server. This connection is called the Primary Channel. STUN server will return its alternate IP address and alternate port in OTHER-ADDRESS in the binding response [RFC5780].
2. The client then sends STUN Binding Request to the STUN server using alternate IP address and alternate port. This connection is called the Secondary Channel.
3. The Client will initially set the default keepalive interval for NAT/FW mappings to 60 seconds (FWa).
4. After FWa seconds the Client will send a binding request to the STUN server using the Primary Channel with the CHANGE-REQUEST attribute set to change-port and change-IP. This will cause the STUN server to send its response from the Secondary channel.
5. If the client receives response from the server then it will increase the keepalive interval value $FWa = (old\ FWa) + (old\ FWa)/2$. This indicates that NAT/FW mappings are alive.
6. Steps 4 and 5 will be repeated until there is no response from the STUN server. If there is no response from the STUN server then the client will use the FWa value as Keepalive interval to refresh FW/NAT mappings.

The above procedure will be done separately for each transport protocol. For connectionless transport protocols like UDP if timer of 2 seconds elapses without response from the STUN server then the client will repeat step 4 atmost three times to handle packet loss.

This procedure can be adopted by other protocols to use Primary and Secondary channels, so that the client can determine the keepalive interval to refresh FW/NAT mapping. This procedure only serves as a guideline and if applications already use some other heuristics method to determine keepalive, they can continue with the existing logic. For example Teredo determines Refresh interval using the procedure in "Optional Refresh Interval Determination Procedure" (Section 5.2.7 of [RFC4380]).

To improve reliability, applications SHOULD continue to use PCP to lengthen the FW/NAT mappings even if the above described mechanism is used to detect PCP unaware NAT/Firewall. This ensures that PCP aware FW/NAT do not close old mappings with no packet exchange when there is a resource-crunch situation.

5. Application-Specific Operation

This section describes how PCP is used with specific application protocols.

5.1. SIP

For connection-less transports the User Agent (UA) sends a STUN Binding Request over the SIP flow as described in section 4.4.2 of [RFC5626]. The UA then learns the External IP Address and Port using a PEER request/response. If the XOR-MAPPED-ADDRESS in the STUN Binding Response matches the external address and port provided by PCP PEER response then the UA optimizes the keepalive traffic as described in Section 3.4. There is no further need to send STUN Binding Requests over the SIP flow to keep the NAT binding alive.

If the XOR-MAPPED-ADDRESS in the STUN Binding Response does not match the external address and port provided by the PCP PEER response then PCP will not be used to keep the NAT bindings alive for the flow that is being used for the SIP traffic. This means that multiple layers of NAT are involved and intermediate NATs are not PCP aware. In this case the UA will continue to use the technique in section 4.4.2 of [RFC5626].

For connection-oriented transports, the UA sends a STUN Binding Request multiplexed with SIP over the TCP connection. STUN multiplexed with other data over a TCP or TLS-over-TCP connection is explained in section 7.2.2 of [RFC5389]. The UA then learns the External IP address and port using a PEER request/response. If the XOR-MAPPED-ADDRESS in the STUN Binding Response matches the external address and port provided by PCP PEER response then the UA optimizes the keepalive traffic as described in Section 3.4.

If the XOR-MAPPED-ADDRESS in the STUN Binding Response does not match the external address and port provided by PCP PEER response then PCP will not be used to keep the NAT bindings alive. In this case the UA performs a keep-alive check by sending a double-CRLF (the "ping") then waits to receive a single CRLF (the "pong") using the technique in section 4.4.1 of [RFC5626].

5.2. HTTP

Web Applications that require persistent connections use techniques such as HTTP long polling and Websockets for session keep alive as explained in section 3.1 of [I-D.isomaki-rtcweb-mobile]. In such scenarios, after the client establishes a connection with the HTTP server, it can execute server side scripts such as PHP residing on the server to provide the transport address and port of the HTTP

client seen at the HTTP server. In addition, the HTTP client also learns the external IP Address and port using the PCP PEER request/response.

If the IP address and port learned from the server matches the external address and port provided by PCP PEER response then the HTTP client optimizes keepalive traffic as described in Section 3.4.

If the IP address and port do not match then PCP will not be used to keep the NAT bindings alive for the flow that is being used for the HTTP traffic. This means that there are NATs or HTTP proxies between the PCP server and the HTTP server. The HTTP client will have to resort to use existing techniques for keep alive. Please see Appendix A for an example server side PHP script to obtain the client source IP address.

HTTP protocol allows intermediaries like transparent proxies to be involved and there is no way for the client to know that request/response is relayed through a proxy.

5.3. Media and data channels with ICE

The ICE agent learns the External IP Address and Port using a MAP request/response. This candidate learnt through PCP is encoded in the ICE offer and answer just like the server reflexive candidate, If the server reflexive candidate and External IP address learnt using PCP are different. When using the Recommended Formula in section 4.1.2.1 of [RFC5245] to compute priority for the candidates learnt through PCP, the ICE agent can use a preference value greater than or equal to the server reflexive candidates.

The ICE agent in addition to ICE connectivity checks and performs the following :

The ICE agent checks if the XOR-MAPPED-ADDRESS from the STUN [RFC5389] Binding response received as part of ICE connectivity check matches the external address and port provided by PCP MAP response.

1. If the match is successful then PCP will be used to keep the NAT bindings alive. The ICE agent optimizes keepalive traffic by refreshing the mapping via a new PCP MAP request containing information from the earlier PCP response.
2. If the match is not successful then PCP will not be used for keep NAT binding alive. The ICE agent will use the technique in section 4.4 of [RFC6263] to keep NAT bindings alive. This means that multiple layers of NAT are involved and intermediate NATs are not PCP aware.

Some network operators deploying a PCP Server may allow PEER but not MAP. In such cases the ICE agent learns the external IP address and port using a STUN binding request/response during ICE connectivity checks. The ICE agent also learns the external IP Address and port using a PCP PEER request/response. If the IP address and port learned from the STUN binding response matches the external address and port provided by the PCP PEER response then the ICE agent optimizes keepalive traffic as described in Section 3.4.

5.4. Detecting Flow Failure

Using the Rapid Recovery technique in section 14 of [I-D.ietf-pcp-base] PCP client upon receiving a PCP ANNOUNCE from a PCP server becomes aware that PCP server has rebooted or lost its mapping state. The PCP client issues new PCP requests to recreate any lost mapping state and thus reconstructs lost mappings fast enough that existing media, HTTP and SIP flows do not break. If the NAT state cannot be recovered the endpoint will find the new external address and port as part of the Rapid Recovery technique in PCP itself and reestablish a connection with the peer.

In lieu of this mechanism if a PCP server reboots and loses its mapping state or when a NAT gateway has its external IP address changed so that its current mapping state becomes invalid, it may take some time before the endpoints realize that the connectivity is lost.

5.5. Firewalls

PCP allows applications to communicate with Firewall devices with PCP functionality to create mappings for incoming connections. In such cases PCP can be used by the endpoint to create an explicit mapping on Firewall to permit inbound traffic and further use PCP to send keep-alives to keep the Firewall mappings alive.

5.5.1. IPv6 Network with Firewalls

As part of the call setup, the endpoint would gather its host candidates and relayed candidate from a TURN server, send the candidates in the offer to the peer endpoint. On receiving the answer from the peer endpoint, the PCP client sends a PCP MAP request with FILTER opcode to create a dynamic mapping in Firewall to permit ICE connectivity checks and subsequent media traffic from the remote peer.

5.5.2. Mobile Network with Firewalls

Mobile Networks are also making use of a Firewall to protect their customers from various attacks like downloading malicious content. The Firewall is usually configured to block all unknown inbound connections as explained in section 2.1 of [I-D.chen-pcp-mobile-deployment]. In such cases PCP can be used by Mobile devices to create an explicit mapping on the Firewall to permit inbound traffic and optimize the keepalive traffic as described in Section 3.4. This would result in saving of radio and power consumption of the Mobile device while protecting it from attacks.

6. IANA Considerations

None

7. Security Considerations

The security considerations in [RFC5245] and [I-D.ietf-pcp-base] apply to this use.

8. Acknowledgements

Authors would like to thank Dave Thaler, Basavaraj Patil for valuable inputs to the document.

9. Change History

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

9.1. Changes from draft-reddy-pcp-optimize-keepalives-00

- o Added sections 3.3, 4
- o Updated section 3 and 3.4 and Introduction

10. References

10.1. Normative References

- [I-D.ietf-pcp-base]
Wing, D., Cheshire, S., Boucadair, M., Penno, R., and P. Selkirk, "Port Control Protocol (PCP)", draft-ietf-pcp-base-29 (work in progress), November 2012.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC5245] Rosenberg, J., "Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols", RFC 5245, April 2010.
- [RFC5389] Rosenberg, J., Mahy, R., Matthews, P., and D. Wing, "Session Traversal Utilities for NAT (STUN)", RFC 5389, October 2008.
- [RFC5626] Jennings, C., Mahy, R., and F. Audet, "Managing Client-Initiated Connections in the Session Initiation Protocol (SIP)", RFC 5626, October 2009.
- [RFC5780] MacDonald, D. and B. Lowekamp, "NAT Behavior Discovery Using Session Traversal Utilities for NAT (STUN)", RFC 5780, May 2010.
- [RFC6263] Marjou, X. and A. Sollaud, "Application Mechanism for Keeping Alive the NAT Mappings Associated with RTP / RTP Control Protocol (RTCP) Flows", RFC 6263, June 2011.

10.2. Informative References

- [I-D.binet-v6ops-cellular-host-reqs-rfc3316update]
Binet, D., Boucadair, M., Ales, V., Byrne, C., and G. Chen, "Internet Protocol Version 6 (IPv6) for Cellular Hosts", draft-binet-v6ops-cellular-host-reqs-rfc3316update-03 (work in progress), October 2012.
- [I-D.chen-pcp-mobile-deployment]
Chen, G., Cao, Z., Boucadair, M., Ales, V., and L. Thiebaut, "Analysis of Port Control Protocol in Mobile Network", draft-chen-pcp-mobile-deployment-02 (work in progress), October 2012.
- [I-D.isomaki-rtcweb-mobile]
Isomaki, M., "RTCweb Considerations for Mobile Devices",

draft-isomaki-rtcweb-mobile-00 (work in progress),
July 2012.

- [RFC2177] Leiba, B., "IMAP4 IDLE command", RFC 2177, June 1997.
- [RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, June 2002.
- [RFC3921] Saint-Andre, P., Ed., "Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence", RFC 3921, October 2004.
- [RFC4380] Huitema, C., "Teredo: Tunneling IPv6 over UDP through Network Address Translations (NATs)", RFC 4380, February 2006.

Appendix A. Example PHP script

```
<html>
Connected to <?PHP echo gethostname(); ?> on port <?PHP echo
getenv(SERVER_PORT) ?> on <?PHP echo date("d-M-Y H:i:s"); ?> Pacific Time
<p>
Your IP address is: <?PHP echo getenv(REMOTE_ADDR); ?>,
port <?PHP echo getenv(REMOTE_PORT); ?>
</p>;
</html>
```

Authors' Addresses

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

Email: tireddy@cisco.com

Markus Isomaki
Nokia
Keilalahdentie 2-4
FI-02150 Espoo
Finland

Email: markus.isomaki@nokia.com

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

Email: dwing@cisco.com

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

Email: praspatti@cisco.com

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

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

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

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

Table of Contents

1. Introduction	2
1.1. Lightweight 4over6	2
1.2. Applications Using Port Sets	3
1.3. Firewall Control	3
2. Terminology	3
3. The need for PORT_SET	3
4. The PORT_SET Option	4
4.1. Client Behavior	5
4.2. Server Behavior	6
4.3. Port Set Renewal and Deletion	6
5. Operational Considerations	7
6. Security Considerations	7
7. IANA Considerations	7
8. Authors List	7
9. Acknowledgements	8
10. References	9
10.1. Normative References	9
10.2. informative References	9
Authors' Addresses	9

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.

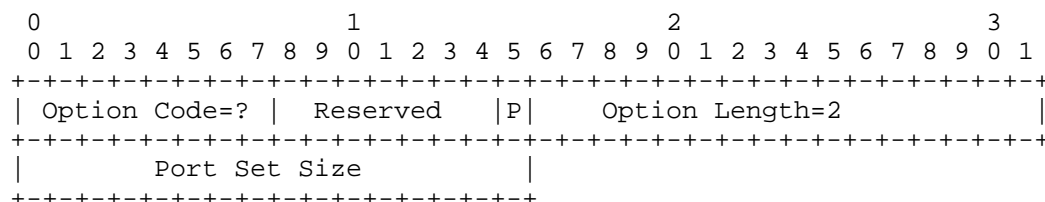


Figure 1: PORT_SET Option

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:

- o If multiple PORT_SET options are present in a single MAP request, a MALFORMED_OPTION error is returned.
- o 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-software-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