Independent Submission
Request for Comments: 7767
Category: Informational
ISSN: 2070-1721

S. Vinapamula
Juniper Networks
S. Sivakumar
Cisco Systems
M. Boucadair
Orange
T. Reddy
Cisco

February 2016

Application-Initiated Check-Pointing via the Port Control Protocol (PCP)

Abstract

This document specifies a mechanism for a host to indicate via the Port Control Protocol (PCP) which connections should be protected against network failures. These connections will then be subject to high-availability mechanisms enabled on the network side.

This approach assumes that applications and/or users have more visibility about sensitive connections than any heuristic that can be enabled on the network side to guess which connections should be check-pointed.

Status of This Memo

This document is not an Internet Standards Track specification; it is published for informational purposes.

This is a contribution to the RFC Series, independently of any other RFC stream. The RFC Editor has chosen to publish this document at its discretion and makes no statement about its value for implementation or deployment. Documents approved for publication by the RFC Editor are not a candidate for any level of Internet Standard; see Section 2 of RFC 5741.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at http://www.rfc-editor.org/info/rfc7767.

Copyright Notice

Table of Contents

1. Introduction

The risk of Internet service disruption is critical in service providers and enterprise networking environments. Such a risk is often mitigated with the introduction of active/backup systems. Such designs not only contribute to minimize the risk of service disruption, but also facilitate maintenance operations (e.g., hitless hardware or software upgrades).

In addition, the nature of some connections leads to the establishment and the maintenance of connection-specific states by some of the network functions invoked when the connection is established. During active/backup failover in case of a network failure, the said states need to be check-pointed by the backup system. Additional issues are discussed in Section 2.

Heuristics based on the protocol, mapping lifetime, etc., are used in the network to elect which connections need to be check-pointed (e.g., by means of high-availability (HA) techniques). This document advocates for an application-initiated approach that would allow applications and/or users to signal to the network which of their connections are critical.

Within this document, "check-pointing" refers to a process of state replication and synchronization between active and backup PCP-controlled devices. When the active PCP-controlled device fails, the backup PCP-controlled device will take over all the existing established sessions that were check-pointed. This process is transparent to internal hosts.

This document specifies how PCP [RFC6887] can be extended to indicate which connection should be check-pointed for high availability (Section 3). A set of use cases are provided for illustrative purposes in Section 4. This document does not make any assumptions about the PCP-controlled device that will process the PCP-formatted signaling information from PCP clients. These devices are likely to be flow aware.

The approach in this document is aligned with the networking trends advocating for open network APIs to interact with applications/services (e.g., [RFC7149]). For instance, the decision-making process about policy on the network side will be enriched with information provided by applications using PCP.

1.1. Note

The CHECKPOINT_REQUIRED PCP option (Section 3) is defined in the "Specification Required" range (see Section 6). In order to be assigned a code point in that range, a permanent publication is required as per Section 4.1 of [RFC5226]. Publication of an RFC is an ideal means of achieving this requirement and also to ease interoperability.

Note, this work was presented to the Port Control Protocol (PCP) WG, but there was no consensus to define this option in the "Standards Action" range despite positive feedback that was received from the working group. Technical comments that were received during PCP meetings and those received on the mailing list were addressed.

1.2. 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].

2. Issues with the Existing Implementations

Regardless of the selected technology or design like HA-based designs, reliably securing connections is expensive in terms of memory, CPU usage, and other resources. Also, check-pointing may not be required for all connections, as all connections may not be critical. But, this leaves a challenge to identify what connections to check-point.

Typically, this is addressed by identifying long-lived connections and check-pointing the state of only those connections that lived long enough, to the backup for service continuity.

However, check-pointing long-lived connections raises the following issues:

1. It is hard for a network to identify (or guess) which connection is (business) critical. This characterization is often customer-specific: a flow can be sensitive for a User #1, while it is not for another User #2. Furthermore, this characterization can vary over time: a flow can be sensitive during hour X, while it is not during other times.

2. Heuristics are not deterministic.

3. A potentially long-lived connection may experience disruption upon failure of the active system, but before it is check-pointed.

4. A connection may not be long-lived but it may be critical, e.g., for Voice over IP (VoIP) conversations.

5. Likewise, not all long-lived connections are deemed critical: for example, connections that pertain to free Internet services are usually considered not critical compared to the equivalent connections for paid services. Only the latter need to be check-pointed.

3. CHECKPOINT_REQUIRED PCP Option

3.1. Format

The solution is based on the assumption that an application or user is the best judge of which of its connections are critical.

An application or user may explicitly identify the connections that need to be check-pointed by means of a PCP client, using the CHECKPOINT_REQUIRED option as described in Figure 1.

The entry to be backed up is indicated by the content of a MAP or PEER message.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Option Code=192|  Reserved     |        Option Length          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

         Option Name: CHECKPOINT_REQUIRED
         Number: 192
         Purpose:  Indicate if an entry needs to be check-pointed.
         Valid for Opcodes: MAP, PEER
         Length: 0.
         May appear in: Request and response.
         Maximum occurrences: 1.
```

Figure 1: CHECKPOINT_REQUIRED PCP Option

The description of the fields is as follows:

- Option Code: 192 (see Section 6).
- Reserved: This field is initialized as specified in Section 7.3 of [RFC6887].
- Option Length: 0. This means no data is included in the option.

An application or user can take advantage of this PCP option to explicitly indicate which of the connections need to be check-pointed and should not be disrupted. The processing of this option by the PCP server will then yield the check-pointing of the corresponding states by the relevant devices or functions dynamically controlled by the PCP server.

Communication between application/user and PCP client is implementation specific.

3.2. Operation

Support of the CHECKPOINT_REQUIRED option by PCP servers and PCP clients is optional. This option (Code 192; see Figure 1) may be included in a PCP MAP or PEER request to indicate a connection is to be protected against network failures.

There is a risk that every PCP client may wish to check-point every connection; this can potentially load the system. Administration SHOULD restrict the number of connections that can be elected to be backed up and the rate of check-pointing per network attachment point (e.g., Customer Premises Equipment (CPE), host). To that aim, the PCP server should unambiguously identify the network attachment point a PCP client belongs to. For example, the PCP server may rely on the PCP identity [RFC7652], the assigned prefix to a CPE or host, the subscriber-mask [PREFIX-BINDING], or other identification means.

The PCP client includes a CHECKPOINT_REQUIRED option in a MAP or PEER request to signal that the corresponding mapping is to be protected.

If the PCP client does not receive a CHECKPOINT_REQUIRED option in response to a PCP request that enclosed the CHECKPOINT_REQUIRED option, this means that either the PCP server does not support the option, or the PCP server is configured to ignore the option, or the PCP server cannot satisfy the request expressed in this option (e.g., because of a lack of resources).

If the CHECKPOINT_REQUIRED option is not included in the PCP client request, the PCP server MUST NOT include the CHECKPOINT_REQUIRED option in the associated response.

When the PCP server receives a CHECKPOINT_REQUIRED option, the PCP server checks if it can honor this request depending on whether resources are available for check-pointing. If there are no resources available for check-pointing, but there are resources available to honor the MAP or PEER request, a response is sent back to the PCP client without including the CHECKPOINT_REQUIRED option (i.e., the request is processed as any MAP or PEER request that does not convey a CHECKPOINT_REQUIRED option). If check-pointing resources are still available and the quota for this PCP client has not been reached, the PCP server tags the corresponding entry as eligible to the HA mechanism and sends back the CHECKPOINT_REQUIRED option in the positive answer to the PCP client.

To update the check-pointing behavior of a mapping maintained by the PCP server, the PCP client generates a PCP MAP or PEER renewal request that includes a CHECKPOINT_REQUIRED option to indicate this mapping has to be check-pointed or that doesn't include a CHECKPOINT_REQUIRED option to indicate this mapping does not need be check-pointed anymore. Upon receipt of the PCP request, the PCP server proceeds with the same operations to validate a MAP or PEER request to update an existing mapping. If validation checks are passed, the PCP server updates the check-point flag associated with that mapping accordingly (i.e., it is set if a CHECKPOINT_REQUIRED option was included in the update request or it is cleared if no CHECKPOINT_REQUIRED option was included), and the PCP server returns the response to the PCP client accordingly.

What information to check-point and how to check-point are outside the scope of this document and are left for implementations. Also, the mechanism for users or applications to indicate check-pointing in a PCP request may be automatic, semiautomatic, or require human intervention. This behavior is also left for application implementations. For managed CPEs, a service provider may influence what connections are to be check-pointed.

For honored requests, it is RECOMMENDED to check-point state on backup before a response is sent to the PCP client.

4. Sample Use Cases

Below are provided some examples for illustrative purposes:

Example 1: Consider a streaming service such as live TV broadcasting, or any other media streaming, that supports check-pointing signaling functionality. Suppose this application is installed in three hosts A, B and C. For A, the application is critical and should not be interrupted, while for B it is not. While for C, only some programs are of interest. At the time of installing this application's software, corresponding preferences can be provisioned. When the application starts streaming:

* All the flows associated with the streaming application are critical for A. Limiting the number of flows to be backed up will ensure that host doesn't exceed the user's limit.

* For B, none of these flows are critical for check-pointing. The CHECKPOINT_REQUIRED option is not included in the PCP requests.

* For C, the user is invited to interact with the application by means of a configuration option that is provided to dynamically select which streaming to check-point, based on the user's interest.

Example 2: Consider a streaming service offered by a provider. Suppose three levels of subscriptions are offered by that provider, e.g., gold, silver, and bronze. To guarantee a certain level of quality of service for each subscription, policies are configured such that:

* All flows associated with a gold subscription should be check-pointed.

* Only some flows associated with a silver subscription are check-pointed.

- None of the flows associated with a bronze subscription are check-pointed.

When a user invokes the streaming service, he/she may fall into one of those buckets, and according to the configured policy, his/her associated streaming flows are automatically check-pointed. Login credentials can be used as a trigger to determine the subscription level (and therefore the associated check-pointing behavior).

Example 3: Consider a VoIP application that is able to request that its flows be check-pointed. No matter what is configured by the user, some calls such as emergency calls should be check-pointed. The application has to identify such calls.

Example 4: In the context of an enterprise network, applications are customized by the administrator. Instructions about whether a CHECKPOINT_REQUIRED option is to be included are determined by the administrator. Only the subset of applications identified by the administrator will make use of this option in conformance with the enterprise network's management policies. Any misbehavior can be considered as abuse.

In order to prevent every application from including a CHECKPOINT_REQUIRED option in its PCP requests, the following items are assumed:

- Applications may be delivered with some default settings for check-pointing, and these settings should be programmable by end user.

- Exposing and enforcing these settings is application specific.

- The end user may customize these settings based on the requirements.

5. Security Considerations

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

The CHECKPOINT_REQUIRED option can be used by an attacker to identify critical flows; this is sensitive from a privacy standpoint. Also, an attacker can cause critical flows to not be check-pointed by stripping the CHECKPOINT_REQUIRED option or by consuming the quota by adding the option to other flows.

These two issues can be mitigated if the network on which the PCP messages are to be sent is fully trusted. Means to defend against attackers who can intercept packets between the PCP server and the PCP client should be enabled. In some deployments, access control lists (ACLs) can be installed on the PCP client, PCP server, and the network between them, so those ACLs allow only communications between trusted PCP elements. If the networking environment between the PCP client and the PCP server is not secure, PCP authentication [RFC7652] MUST be enabled.

A network device can always override the end-user signaling, i.e., what is signaled by the PCP client, if the instructions conflict with the network policies.

6. IANA Considerations

The following PCP Option Code has been allocated in the "Specification Required" range of the "PCP Options" registry (http://www.iana.org/assignments/pcp-parameters):

192 CHECKPOINT_REQUIRED (see Section 3.1)

7. References

7.1. Normative References

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

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

[RFC7652] Cullen, M., Hartman, S., Zhang, D., and T. Reddy, "Port Control Protocol (PCP) Authentication Mechanism", RFC 7652, DOI 10.17487/RFC7652, September 2015, <http://www.rfc-editor.org/info/rfc7652>.

7.2. Informative References

[PREFIX-BINDING] Vinapamula, S. and M. Boucadair, "Recommendations for Prefix Binding in the Softwire DS-Lite Context", Work in Progress, draft-vinapamula-softwire-dslite-prefix-binding-12, October 2015.

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, DOI 10.17487/RFC5226, May 2008, <http://www.rfc-editor.org/info/rfc5226>.

[RFC7149] Boucadair, M. and C. Jacquenet, "Software-Defined Networking: A Perspective from within a Service Provider Environment", RFC 7149, DOI 10.17487/RFC7149, March 2014, <http://www.rfc-editor.org/info/rfc7149>.

Appendix A. Additional Considerations

It was tempting to include additional fields in the option but this would lead to a more complex design that is not justified. For example, we considered the following.

- Define a dedicated field to indicate a priority level. This priority is intended to be used by the PCP server as a hint when processing a request with a CHECKPOINT_REQUIRED option. Nevertheless, an application may systematically choose to set the priority level to the highest value so that it increases its chance to be serviced!

- Return a more granular failure error code to the requesting PCP client. However, this would require extra processing at both the PCP client and server sides for handling the various error codes without any guarantee that the PCP client would have its mappings check-pointed.

Acknowledgments

Thanks to Reinaldo Penno, Stuart Cheshire, Dave Thaler, Prashanth Patil, and Christian Jacquenet for their comments.

Authors' Addresses

Suresh Vinapamula
Juniper Networks
1194 North Mathilda Avenue
Sunnyvale, CA 94089
United States

Phone: +1 408 936 5441
Email: sureshk@juniper.net

Senthil Sivakumar
Cisco Systems
7100-8 Kit Creek Road
Research Triangle Park, NC 27760
United States

Phone: +1 919 392 5158
Email: ssenthil@cisco.com

Mohamed Boucadair
Orange
Rennes 35000
France

Email: mohamed.boucadair@orange.com

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

Email: tireddy@cisco.com