CORE M. Boucadair
Internet-Draft Orange
Intended status: Standards Track T. Reddy
Expires: ~~August 30,~~ **December 27,** 2019 McAfee
J. Shallow
~~NCC Group~~
~~February 26,~~
June 25, 2019

Constrained Application Protocol (CoAP) Hop Limit Option
~~draft-ietf-core-hop-limit-03~~
draft-ietf-core-hop-limit-04

Abstract

The presence of Constrained Application Protocol (CoAP) proxies may lead to infinite forwarding loops, which is undesirable. To prevent and detect such loops, this document specifies the Hop-Limit CoAP option.

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 https://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 30,~~ **December 27,** 2019.

Table of Contents

1. Introduction

More and more applications are using **the** Constrained Application Protocol (CoAP) [RFC7252] as a communication protocol between involved application agents. For example, [I-D.ietf-dots-signal-channel] specifies how CoAP is used as a distributed denial-of-service (DDoS) attack signaling protocol **for** seeking for help from DDoS mitigation providers. In such contexts, a CoAP client can communicate directly with a server or indirectly via proxies.

When multiple proxies are involved, infinite forwarding loops may be ~~experienced.~~ **experienced (e.g., routing misconfiguration, policy conflicts).** To prevent such loops, this document defines a new CoAP option, called Hop-Limit (Section ~~3), which is inserted in particular by on-path proxies.~~ **3).** Also, the document defines a new CoAP Response Code (Section 4.1) to report loops together with relevant diagnostic information to ease troubleshooting.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119][RFC8174] when, and only when, they appear in all capitals, as shown here.

Readers should be familiar with the terms and concepts defined in [RFC7252].

3. Hop-Limit Option

The Hop-Limit option (see Section 4.2) is an elective option used to detect and prevent infinite loops when proxies are involved. The option is not repeatable. Therefore, any message carrying multiple Hop-Limit options MUST be ~~rejected using 4.00 (Bad Request) error message.~~ **handled following the procedure specified in Section 5.4.5 of [RFC7252].**

The value of the Hop-Limit option is encoded as an ~~8-bit~~ unsigned integer (see Section 3.2 of [RFC7252]). This value MUST be between 1 and 255 inclusive. CoAP messages received with a Hop-Limit option set to '0' or greater than '255' MUST be rejected by a CoAP ~~server/proxy~~ **server/proxy** using 4.00 (Bad Request).

The Hop-Limit option is safe to forward. That is, a CoAP proxy which does not understand the Hop-Limit option should forward it on. The option is also part of the cache key. As such, a CoAP proxy which does not understand the Hop-Limit option must follow the recommendations in Section 5.7.1 of [RFC7252] for caching. Note that loops which involve only such proxies won't be detected. Nevertheless, the presence of such proxies won't prevent infinite loop detection if at least one CoAP proxy which support the Hop-Limit option is involved in the loop.

A CoAP proxy which understands the Hop-Limit option MAY be

instructed, using a configuration parameter, to insert a Hop-Limit option when relaying a request which do not include the Hop-Limit option.

The initial Hop-Limit value SHOULD be configurable. If no initial value is explicitly provided, the default initial Hop-Limit value of 16 MUST be used. This value is chosen to be sufficiently large to guarantee that a CoAP request would not be dropped in networks when there were no loops, but not so large as to consume CoAP proxy resources when a loop does occur. Lower values should be used with caution and only in networks where topologies are known by the CoAP client (or proxy) inserting the Hop-Limit option.

Because forwarding errors may occur if inadequate Hop-Limit values are used, proxies at the boundaries of an administrative domain MAY be instructed to remove or rewrite the value of Hop-Limit carried in received messages (i.e., ignore the value of Hop-Limit received in a message). This modification should be done with caution in case proxy-forwarded traffic repeatedly crosses the administrative domain boundary in a loop and so Hop-Limit detection gets broken.

Otherwise, a CoAP proxy which understands the Hop-Limit option MUST decrement the value of the option by 1 prior to forwarding it. A CoAP proxy which understands the Hop-Limit option MUST NOT use a stored TBA1 (Hop Limit Reached) error response unless the value of the Hop-Limit option in the presented request is less than or equal to the value of the Hop-Limit option in the request used to obtain the stored response. Otherwise, the CoAP proxy follows the behavior in Section 5.6 of [RFC7252].

> Note: If a request with a given value of Hop-Limit failed to reach a server because the hop limit is exhausted, then the same failure will be observed if a less value of the Hop-Limit option is used instead.

CoAP messages MUST NOT be forwarded if the Hop-Limit option is set to '0' after decrement. Messages that cannot be forwarded because of exhausted Hop-Limit SHOULD be logged with a TBA1 (Hop Limit Reached) error response sent back to the CoAP peer. It is RECOMMENDED that CoAP implementations support means to alert administrators about loop errors so that appropriate actions are undertaken.

To ease debugging and troubleshooting, the CoAP proxy which detects a loop SHOULD include its information (e.g., proxy name, proxy alias, IP address) in the diagnostic payload under the conditions detailed in Section 5.5.2 of [RFC7252]. That information MUST NOT include any space character.

Each intermediate proxy involved in relaying a TBA1 (Hop Limit Reached) error message SHOULD prepend its own information in the diagnostic payload with a space character used as separator. Only one information per proxy SHOULD appear in the diagnostic payload. Doing so allows to limit the size of the TBA1 (Hop Limit Reached) error message, and to ease correlation with hops count. **Note that an intermediate proxy prepends its information only if there is enough space. If not, an intermediate proxy forwards the TBA1 (Hop Limit Reached) error message to the next hop without updating the diagnostic payload.**

4. IANA Considerations

4.1. CoAP Response Code

IANA is requested to add the following entry to the "CoAP Response Codes" sub-registry available at https://www.iana.org/assignments/core-parameters/core-parameters.xhtml#response-codes:

Code	Description	Reference
TBA1	Hop Limit Reached	[RFCXXXX]

Table 1: CoAP Response Codes

This document suggests 5.06 as a code to be assigned for the new response code.

Editorial Note: Please update TBA1 statements within the document with the assigned code.

4.2. CoAP Option Number

IANA is requested to add the following entry to the "CoAP Option Numbers" sub-registry available at https://www.iana.org/assignments/core-parameters/core-parameters.xhtml#option-numbers:

Number	C	U	N	R	Name	Reference
TBA2					Hop-Limit	[RFCXXXX]

C=Critical, U=Unsafe, N=NoCacheKey, R=Repeatable

Table 2: CoAP Option Number

5. Security Considerations

Security considerations related to CoAP proxying are discussed in Section 11.2 of [RFC7252].

The diagnostic payload of a TBA1 (Hop Limit Reached) error message may leak sensitive information revealing the topology of an administrative domain. To prevent that, a CoAP proxy which is located at the boundary of an administrative domain MAY be instructed to strip the diagnostic payload or part of it before forwarding on the TBA1 response.

6. Acknowledgements

This specification was part of [I-D.ietf-dots-signal-channel]. Many thanks to those who reviewed DOTS specifications.

Thanks to Klaus Hartke, Carsten Bormann, Peter van der Stok, and Jim Schaad for the reviews.

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, <https://www.rfc-editor.org/info/rfc2119>.

[RFC7252] Shelby, Z., Hartke, K., and C. Bormann, "The Constrained

Application Protocol (CoAP)", RFC 7252,
DOI 10.17487/RFC7252, June 2014,
<https://www.rfc-editor.org/info/rfc7252>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC
2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174,
May 2017, <https://www.rfc-editor.org/info/rfc8174>.

7.2. Informative References

[I-D.ietf-dots-signal-channel]
K, R., Boucadair, M., Patil, P., Mortensen, A., and N.
Teague, "Distributed Denial-of-Service Open Threat
Signaling (DOTS) Signal Channel Specification", draft-
~~ietf-dots-signal-channel-28~~
ietf-dots-signal-channel-34 (work in progress), ~~January~~ **May** 2019.

Authors' Addresses

Mohamed Boucadair
Orange
Rennes 35000
France

Email: mohamed.boucadair@orange.com

Tirumaleswar Reddy
McAfee, Inc.
Embassy Golf Link Business Park
Bangalore, Karnataka 560071
India

Email: kondtir@gmail.com
Jon Shallow
~~NCC Group~~
United Kingdom

Email: ~~jon.shallow@nccgroup.com~~ **supjps-ietf@jpshallow.com**