Internet Engineering Task Force (IETF)
Request for Comments: 8516
Category: Standards Track
ISSN: 2070-1721

A. Keranen
Ericsson

January 2019

"Too Many Requests" Response Code for the Constrained Application Protocol

Abstract

A Constrained Application Protocol (CoAP) server can experience temporary overload because one or more clients are sending requests to the server at a higher rate than the server is capable or willing to handle. This document defines a new CoAP response code for a server to indicate that a client should reduce the rate of requests.

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at https://www.rfc-editor.org/info/rfc8516.

Table of Contents

1. Introduction

The Constrained Application Protocol (CoAP) [RFC7252] response codes are used by a CoAP server to indicate the result of an attempt to understand and satisfy a request sent by a client.

CoAP response codes are similar to the HTTP [RFC7230] status codes, and many codes are shared with similar semantics by both CoAP and HTTP. HTTP has the code "429" registered for "Too Many Requests" [RFC6585]. This document registers a CoAP response code "4.29" for similar purposes and uses the Max-Age option (see Section 5.10.5 of [RFC7252]) to indicate a back-off period after which a client can try the request again.

While a server may not be able to respond to one kind of request, it may be able to respond to a request of a different kind, even from the same client. Therefore, the back-off period applies only to similar requests. For the purpose of this response code, a request is similar if it has the same method and Request-URI. Also, if a client is sending a sequence of requests that are part of the same series (e.g., a set of measurements to be processed by the server), they can be considered similar even if request URIs are different. Because request similarity is context-dependent, it is up to the application logic to decide how the similarity of the requests should be evaluated.

The 4.29 code is similar to the 5.03 "Service Unavailable" [RFC7252] code in that the 5.03 code can also be used by a server to signal an overload situation. The 5.03 code also uses the Max-Age option to indicate the time after which a client can retry. However, the 4.29 code indicates that the too-frequent requests from the requesting client are the reason for the overload.

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 also be familiar with the terms and concepts discussed in [RFC7252].

3. CoAP Server Behavior

If a CoAP server is unable to serve a client that is sending CoAP request messages more often than the server is capable or willing to handle, the server SHOULD respond to the request(s) with the response code 4.29, "Too Many Requests". The Max-Age option is used to indicate the number of seconds after which the server assumes it is OK for the client to retry the request.

An action result payload (see Section 5.5.1 of [RFC7252]) can be sent by the server to give more guidance to the client, e.g., details of the overload situation.

The 4.29 response code is only returned to the client(s) sending requests too frequently; if other clients are sending requests that cannot be served due to server overload, the 5.03 response code is more appropriate.

If a client repeats a request that was answered with 4.29 before Max-Age time has passed, it is possible that the client sent multiple requests before receiving the first answer or that the client did not recognize the response code. To slow down clients that do not recognize the 4.29 code, the server MAY respond with a more generic error code (e.g., 5.03). The server SHOULD rate-limit 4.29 replies taking into account its usual load-shedding policies. However, any such method that adds per-client state to the server may be counterproductive to reducing the load.

4. CoAP Client Behavior

If a client receives the 4.29 response code from a CoAP server to a request, it SHOULD NOT send a similar request to the server before the time indicated in the Max-Age option has passed. If the 4.29 response does not contain a Max-Age option, the default value (60 seconds, as defined in Section 5.10.5 of [RFC7252]) is assumed.

Note that a client may receive a 4.29 response code on a first request to a server. This can happen, for example, if there is a proxy on the path and the server replies based on the load from multiple clients aggregated by the proxy, or if a client has restarted recently and does not remember its recent requests.

A client should not rely on a server being able to send the 4.29 response code in an overload situation because an overloaded server may not be able to reply at all to some requests.

5. Security Considerations

Security considerations of [RFC7252] apply to this response code also.

Replying to CoAP requests with a response code consumes resources from a server. For a server under attack, it may be more appropriate to simply drop requests without responding at all. However, dropping requests is also likely to cause well-behaving clients to simply retry the requests.

As with any other CoAP reply, a client should trust this response code only to the extent that it trusts the underlying security mechanisms (e.g., DTLS [RFC6347]) for authentication and freshness. If a CoAP reply with the "Too Many Requests" response code is not authenticated and integrity protected, an attacker can attempt to spoof a reply and make the client wait for an extended period of time before trying again.

If the response code is sent without encryption, it may leak information about the server overload situation and client traffic patterns.

6. IANA Considerations

IANA has registered the following response code in the "CoAP Response Codes" subregistry within the "Constrained RESTful Environments (CoRE) Parameters" registry:

- Response Code: 4.29
- Description: Too Many Requests
- Reference: RFC 8516

IANA has added this document as an additional reference for the Max-Age option in the "CoAP Option Numbers" subregistry.

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

[CoAP-BROKER] Koster, M., Keranen, A., and J. Jimenez, "Publish-Subscribe Broker for the Constrained Application Protocol (CoAP)", Work in Progress, draft-ietf-core-coap-pubsub-06, January 2019.

[RFC6347] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security Version 1.2", RFC 6347, DOI 10.17487/RFC6347, January 2012, <https://www.rfc-editor.org/info/rfc6347>.

[RFC6585] Nottingham, M. and R. Fielding, "Additional HTTP Status Codes", RFC 6585, DOI 10.17487/RFC6585, April 2012, <https://www.rfc-editor.org/info/rfc6585>.

[RFC7230] Fielding, R., Ed. and J. Reschke, Ed., "Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing", RFC 7230, DOI 10.17487/RFC7230, June 2014, <https://www.rfc-editor.org/info/rfc7230>.

Acknowledgements

This response code definition was originally part of the "Publish-Subscribe Broker for CoAP" document [CoAP-BROKER]. The author would like to thank Abhijan Bhattacharyya, Carsten Bormann, Daniel Migault, Gyorgy Rethy, Jana Iyengar, Jim Schaad, Klaus Hartke, Mohit Sethi, and Sandor Katona for their contributions and reviews.

Author's Address

Ari Keranen
Ericsson
Hirsalantie 11
02420 Jorvas
Finland

Email: ari.keranen@ericsson.com