CORE
Internet-Draft
Intended status: Standards Track
Expires: December 13, 2016

C. Bormann, Ed.
K. Hartke
Universitaet Bremen TZI

June 11, 2016

CoAP Signaling Messages
draft-bormann-core-coap-sig-01

Abstract

draft-ietf-core-coap-tcp-tls defines how to transport CoAP messages on reliable transports such as TCP, TLS, or WebSockets.

All these underlying protocols have ways to set up connection properties and manage the connection. In many cases, these ways cannot be used very well for managing CoAP's use of the connection.

Signaling messages are a way to signal information that is about the connection. They form a third basic kind of messages in CoAP, beyond requests and responses. Message class 7 is used for signaling messages.

Signaling messages are only relevant for the connection they appear in. The present draft assumes reliable, sequence-preserving connections. It is for further study whether signaling messages are needed or useful for DTLS connections.

The present draft, when adopted, would resolve CoRE tickets #400 (message sizes), #388 (by providing a foundation for a mechanism for version negotiation, once that is needed), #390 (connection close reason), #391 (server name indication), #394 (ping/pong).

This Internet-Draft will expire on December 13, 2016.

Copyright Notice

Table of Contents

1. Introduction

(Please see abstract for now.)

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 [RFC2119].

The definitions of [RFC7252] apply.

In this document, the term "byte" is used in its now customary sense as a synonym for "octet".

Where bit arithmetic is explained, this document uses the notation familiar from the programming language C, except that the operator "**" stands for exponentiation.

3. Signaling messages

Signaling messages are structured like any other CoAP message; they have a code, a token, options, and optionally a payload. The code for a signaling message comes from the 7.xx space.

Option numbers for signaling messages are specific to the message code, i.e., they do not share the number space with CoAP options for request/response messages or with signaling messages using other codes.

Signaling options can be elective or critical (see Section 5.4.1 of [RFC7252]); if a signaling message option is critical and not understood by the receiver, it MUST abort the connection (see Section 7. (If the option is understood but somehow cannot be carried out, the option defines how to handle the situation.)

Payloads in signaling messages are diagnostic payloads (see Section 5.5.2 of [RFC7252]), unless otherwise determined by a signaling message option.

This specification lays out five kinds of signaling messages, without necessarily defining an instance of each of the kinds.

For each message, there is an emitter (that sends the message) and a peer receiving the message.

4. Capability and Settings Messages

Capability and Settings messages are used for two purposes:

- Capability indication options indicate a capability of the emitter to the peer. Capability options are generally elective options.

- Setting options indicate a setting that will be applied by the emitter. Setting options are generally critical options.

Both capability indication options and setting options are cumulative, i.e., a capability message without any option is a no-operation (and can be used as such). (An option that is given might override a previous value for the same option; the option defines how to handle this, if needed.) Most CSM options are useful mainly as initial messages in the connection.

Capability and Settings messages carry the code 7.01 (CSM to be defined).

Code	Name	Reference
7.01	CSM	[RFCthis]

A number of options for Capability and Settings messages are defined in the following subsections.

4.1. Server-Name Setting Option

A client can indicate a default value that it wants to set for the Uri-Host options in the messages it sends to the server:

The Server-Name option is defined as follows:

Option Number	Applies to	Option Name	Reference
1	CSM	Server-Name	[RFCthis]

The Server-Name option is a critical option (1 is odd) and carries a "string" value, with the same restrictions as for Uri-Host (Section 5.10 of RFC 7252: length is between 1 and 255).

For TLS, the initial value for the Server-Name option is given by the SNI value. SECURITY CONSIDERATIONS. For Websockets, the initial value for the Server-Name is given by the HTTP Host header field.

4.2. Max-Message-Size Capability Indication Option

An emitter can indicate a maximum message size that it can comfortably operate on as a recipient.

The Max-Message-Size option is defined as follows:

Option Number	Applies to	Option Name	Reference
2	CSM	Max-Message-Size	[RFCthis]

The Max-Message-Size option is an elective option (2 is even) and carries a "uint" value, indicating the message size in bytes. As per Section 4.6 of [RFC7252], the default value (and the value used when this Option is not implemented) is 1152.

4.3. Using the Capability and Settings message for version negotiation

CoAP is defined in RFC 7252 with a version number of 1. In contrast to the message layer for UDP and DTLS, the CoAP over TCP message layer does not send the version number in each single message. Instead, options for the Capability and Settings message can be used to perform a version negotiation.

At the time of writing, there is no known reason for supporting version numbers different from 1. The details of a version negotiation, once it is actually needed, will depend on the specifics of the new version(s), so the present specification makes no attempt to specify these details. However, Capability and Settings messages have been specifically designed with a view to supporting such a potential future need.

5. Ping and Pong Messages

NOTE: The present specification assumes that the CoAP over TCP specification specifies that empty messages ([RFC7252]) can always be sent and will be ignored. This provides for a basic keep-alive function that can, e.g., refresh NAT bindings. In contrast, Ping and Pong messages are a bidirectional exchange.

A Ping message is responded to by a single Pong message with the same token. As with all signaling messages, the recipient of a Ping or Pong message MUST ignore elective options it does not understand.

Code	Name	Reference
7.02	Ping	[RFCthis]
7.03	Pong	[RFCthis]

5.1. Custody Option

A peer replying to a Ping message can add a Custody Option to the Pong message it returns. The Option indicates that the application has processed all request/response messages that it has received in the present connection ahead of the Ping message that prompted the Pong message. (Note that there is no definition of specific application semantics of "processed", but there is an expectation that the emitter of the Ping leading to the Pong with a Custody Option should be able to free buffers based on this indication.)

A Custody Option can also be sent in a Ping message to explicitly request the return of a Custody Option in the Pong message. A peer is, however, always free to indicate that it has finished processing all previous request/response messages by sending a Custody Option (which is therefore elective) in a Pong message. A peer is also free NOT to send a Custody Option in case it is still processing previous request/response messages, however, it SHOULD delay its response to a Ping with a Custody Option until it also can return one.

Option Number	Applies to	Option Name	Reference
2	Ping, Pong	Custody	[RFCthis]

The Custody option is an elective option (2 is even) and carries an "empty" value.

6. Release Messages

A release message indicates that the emitter does not want to continue maintaining the connection and opts for an orderly shutdown; the details are in the options. A diagnostic payload MAY be included. A release message will normally be replied to by the peer by closing the TCP/TLS connection. Messages may be in flight when the emitter decides to send a Release message; the general expectation is that these will still be processed.

Code	Name	Reference
7.04	Release	[RFCthis]

Release messages can indicate one or more reasons using elective options; the following options are defined:

Option Number	Applies to	Option Name	Reference
2	Release	Bad-Server-Name	[RFCthis]
4	Release	Alternative-Address	[RFCthis]
6	Release	Hold-Off	[RFCthis]

The Bad-Server-Name option indicates that the default as set by the CSM option Server-Name is unlikely to be useful for this server. The value is empty. (2 is even, i.e., the option is elective.)

The Alternative-Address option requests the peer to instead open a connection of the same kind as the present connection to the alternative transport address given. The value is a string, of the form "authority" defined in Section 3.2 of RFC 3986. (4 is even, i.e., the option is elective.) SECURITY CONSIDERATIONS.

The Hold-Off option indicates that the server is requesting that the peer not reconnect to it for the number of seconds given as the value. The value is a uint. (6 is even, i.e., the option is elective.) (Question: Do we need a "go away forever"?)

7. Abort Messages

An abort message indicates that the emitter is unable to continue maintaining the connection and cannot even wait for an orderly release; the emitter shuts down the connection immediately after the abort (and may or may not wait for a release or abort message or connection shutdown in the inverse direction). A diagnostic payload SHOULD be included in the Abort message. Messages may be in flight when the emitter decides to send an abort message; the general expectation is that these will NOT be processed.

Code	Name	Reference
7.05	Abort	[RFCthis]

Abort messages can indicate one or more reasons using elective options; the following options are defined:

Option Number	Applies to	Option Name	Reference
2	Abort	Bad-CSM-Option	[RFCthis]

The Bad-CSM-Option indicates that the emitter is unable to process the CSM option identified by its option number, e.g. when it is critical and the option number is unknown by the emitter, or when there is parameter problem with the value of an elective option. The value is a uint. (2 is even, i.e., the option is elective.) (More detailed information SHOULD be given as a diagnostic payload.)

One reason for an emitter to generate an abort message is a general syntax error in the byte stream received; no specific option has been defined for this, as the details of that syntax error are best left to a diagnostic payload.

8. Examples

An encoded example of a Ping message with a non-empty token is shown in Figure 1.

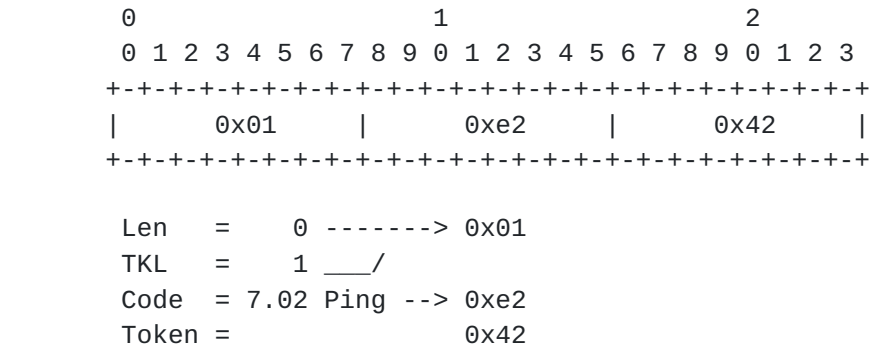

Figure 1: Ping Message Example

An encoded example of the corresponding Pong message is shown in Figure 2.

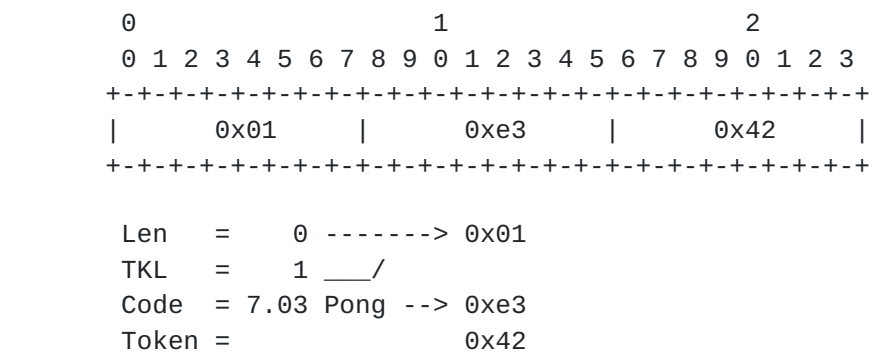

Figure 2: Pong Message Example

9. Security Considerations

The security considerations of [RFC7252] apply.

- The guidance given by an Alternative-Address option cannot be followed blindly.

- SNI vs. Server-Name: The security is for the SNI name.

10. IANA Considerations

10.1. Message Codes

IANA is requested to create a third sub-registry for values of the Code field in the CoAP header (cf. Section 12.1 of [RFC7252]).

(IANA policy TBD.)

(remember to copy down values from above)

Code	Name	Reference
7.01	CSM	[RFCthis]
7.02	Ping	[RFCthis]
7.03	Pong	[RFCthis]
7.04	Release	[RFCthis]
7.05	Abort	[RFCthis]

10.2. Signaling Options

IANA is requested to create a sub-registry for signaling options similar to the CoAP Option Numbers Registry (Section 12.2 of [RFC7252]), with the single change that a fourth column is added to the sub-registry that is one of the message codes in the message code subregistry (Section 10.1).

(IANA policy TBD.)

(remember to copy down values from above)

Option Number	Applies to	Option Name	Reference
1	CSM	Server-Name	[RFCthis]
2	CSM	Max-Message-Size	[RFCthis]
2	Ping, Pong	Custody	[RFCthis]
2	Release	Bad-Server-Name	[RFCthis]
4	Release	Alternative-Address	[RFCthis]
6	Release	Hold-Off	[RFCthis]
2	Abort	Bad-CSM-Option	[RFCthis]

11. References

11.1. 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>.

[RFC7252] Shelby, Z., Hartke, K., and C. Bormann, "The Constrained Application Protocol (CoAP)", RFC 7252, DOI 10.17487/RFC7252, June 2014, <http://www.rfc-editor.org/info/rfc7252>.

11.2. Informative References

[I-D.ietf-core-coap-tcp-tls]
Bormann, C., Lemay, S., Technologies, Z., and H. Tschofenig, "A TCP and TLS Transport for the Constrained Application Protocol (CoAP)", draft-ietf-core-coap-tcp-tls-02 (work in progress), April 2016.

Acknowledgements

Significant parts of the present text have been contributed by Hannes Tschofenig. Matthias Kovatsch contributed significantly to this draft.

Authors' Addresses

Carsten Bormann (editor)
Universitaet Bremen TZI
Postfach 330440
Bremen D-28359
Germany

Phone: +49-421-218-63921
Email: cabo@tzi.org

Klaus Hartke
Universitaet Bremen TZI
Postfach 330440
Bremen D-28359
Germany

Phone: +49-421-218-63905
Email: hartke@tzi.org