CoRE Working Group
Internet-Draft
Intended status: Informational
Expires: February 14, 2013

C. Bormann
Universitaet Bremen TZI
August 13, 2012

CoAP Simple Congestion Control/Advanced
draft-bormann-core-cocoa-00

Abstract

The CoAP protocol needs to be implemented in such a way that it does not cause persistent congestion on the network it uses. The CoRE CoAP specification defines basic behavior that exhibits low risk of congestion with minimal implementation requirements. It also leaves room for combining the base specification with advanced congestion control mechanisms with higher performance.

This specification defines some simple advanced CoRE Congestion Control mechanisms, Simple CoCoA. In the present version -00, it is mainly a straw-man document to gauge the implementation effort required, with a view towards simplifying it further.

Table of Contents

1. Introduction

(See Abstract.)

Extended rationale for this specification can be found in [I-D.bormann-core-congestion-control] and [I-D.eggert-core-congestion-control], as well as in the minutes of the IETF 84 CoRE WG meetings.

1.1. 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] when they appear in ALL CAPS. These words may also appear in this document in lower case as plain English words, absent their normative meanings.

(Note that this document is itself informational, but it is discussing normative statements.)

The term "byte", abbreviated by "B", is used in its now customary sense as a synonym for "octet".

2. Context

In the Vancouver IETF 84 CoRE meeting, a path forward was defined that includes a very simple basic scheme (lock-step with a number of parallel exchanges of 1) in the base specification together with performance-enhancing advanced mechanisms.

This specification assumes approximately the following text in the [I-D.ietf-core-coap] base specification:

1. Change SHOULD in second paragraph of [I-D.ietf-core-coap] 4.7 to MUST; define protocol parameter NSTART as 1.

2. Add text that permits advanced congestion control mechanisms and allows them to change protocol parameters, including NSTART and the binary exponential backoff mechanism.

3. Specify that, outside of exchanges, non-confirmable messages cannot be used without an advanced congestion control mechanism (this is mainly relevant for -observe).

4. Add reference to (and/or cite) [RFC5405] guideline about combining congestion control state for a destination; clarify its meaning for CoAP using the definition of an endpoint.

Additional changes have been made to limit the leeway that implementations have in changing the CoRE protocol parameters; these changes are already gathered in Section 4.8 of [I-D.ietf-core-coap] and will not be repeated here.

The present specification does not address multicast or dithering beyond retransmission dithering.

3. Advanced CoAP Congestion Control: RTO Estimation

For an initiator that plans to make multiple requests to one destination end-point, it may be worthwhile to make RTT measurements in order to obtain a better RTT estimation than that implied by the default initial timeout of 2 to 3 s. This is based on the usual algorithms for RTT estimation [RFC6298], with appropriately extended default/base values. Note that such a mechanism must, during idle periods, decay RTT estimates that are shorter than the basic RTT estimate back to the basic RTT estimate, until fresh measurements become available again.

One important consideration not relevant for TCP is the fact that a CoAP round-trip may include application processing time, which may be hard to predict, and may differ between different resources available at the same endpoint. Servers will only trigger early ACKs (with a non-piggybacked response to be sent later) based on the default timers, e.g. after 1 s. A client that has arrived at a RTT estimate much shorter than the 2 to 3 s used as a default SHOULD therefore not expend all of its retransmissions in the shorter estimated timescale.

It may also be worthwhile to do RTT estimates not just based on information measured from a single destination endpoint, but also based on entire hosts (IP addresses) and/or complete prefixes (e.g., maintain an RTT estimate for a whole /64). The exact way this can be used to reduce the amount of state in an initiator is for further study.

3.1. Blind RTO Estimate

The initial RTO estimate for an endpoint is set to 2 seconds.

Up to four (4) exchanges to an endpoint can be started in parallel. If only the initial RTO estimate is available, the RTO estimate for each exchange started in parallel to other exchanges is set to the highest binary multiple of the parallel exchanges, e.g., if another exchange is already running and is into its second retransmission, the RTO estimate for the additional exchange is 8 seconds.

The binary exponential backoff is truncated at 32 seconds. Similar to the way retransmissions are handled in the base specification, they are dithered between 1 x RTO and ACK_RANDOM_FACTOR x RTO.

3.2. Measured RTO Estimate

The RTO estimator runs two copies of the algorithm defined in [RFC6298], as modified in Section 3.2.1: One copy for exchanges that complete on initial transmissions (the "strong estimator"), and one copy for exchanges that have run into retransmissions (the "weak estimator"). For the latter, there is some ambiguity whether a response is based on the initial transmission or any retransmission. For the purposes of the weak estimator, the time from the initial transmission counts.

The overall RTO estimate is a exponentially weighted moving average (alpha = 0.5) computed of the strong and the weak estimator, which is evolved after each contribution to the weak estimator or to the strong estimator, from the estimator that made the most recent contribution:

```
RTO_overall_ := 0.5 * RTO_recent_ + 0.5 * RTO_overall_
```

3.2.1. Modifications to the algorithm of RFC 6298

The initial value for each of the two RTO estimators is 2 s.

3.2.2. Discussion

In contrast to [RFC6298], this algorithm attempts to make use of ambiguous information from retransmissions. This is motivated by the high non-congestion loss rates expected in constrained node networks, and the need to update the RTO estimators even in the presence of loss. Additional investigation is required to determine whether this is indeed justified.

3.3. Lifetime, Aging

The state of the RTO estimators for an endpoint SHOULD be kept as long as possible. If other state is kept for the endpoint (such as a DTLS connection), it is very strongly RECOMMENDED to keep the RTO state alive at least as long as this other state. It MUST be kept for at least 255 s.

If an estimator has a value that is lower than 1 s, and it is left without further update for a time that is more than 16 times its current value, its value is doubled.

(It is allowed to implement this cumulatively at the time it is used next, possibly approximating multiple doublings by replacing the value with 1/8th of the time that has elapsed since the last update. Alternatively, simple estimators can be simply updated to 1 s after being without update for a time that is more than 16 times its value, or, even simpler, be clamped at 1 s or above.)

4. Advanced CoAP Congestion Control: Non-Confirmables

A CoAP endpoint can send non-confirmables to another CoAP endpoint only at a rate as defined by this document.

Independent of any congestion control mechanisms, a CoAP endpoint can always send non-confirmables if a rate of 1 B/s is not exceeded.

Non-confirmables that form part of exchanges are governed by the rules for exchanges.

Non-confirmables outside exchanges (e.g., [I-D.ietf-core-observe] notifications sent as non-confirmables) are governed by the following rules:

1. Of any 16 consecutive messages towards this endpoint that aren’t responses or acknowledgments, at least 2 of the messages must be confirmable.

2. The confirmable messages must be sent under an RTO estimator, as specified above.

3. The packet rate of non-confirmable messages cannot exceed 1/RTO, where RTO is the overall RTO estimator value at the time the non-confirmable packet is sent.

4.1. Discussion

This is relatively conservative. More advanced versions of this algorithm could run a TFRC-style Loss Event Rate calculator [RFC5348] and apply the TCP equation to achieve a higher rate than 1/RTO.

5. IANA Considerations

This document makes no requirements on IANA. (This section to be removed by RFC editor.)

6. Security Considerations

(TBD. The security considerations of, e.g., [RFC5681], [RFC2914], and [RFC5405] apply. Some issues are already discussed in the security considerations of [I-D.ietf-core-coap].)

7. Acknowledgements

The first document to examine CoAP congestion control issues in detail was [I-D.eggert-core-congestion-control], to which this draft owes a lot.

Michael Scharf did a review of CoAP congestion control issues that asked a lot of good questions. Several Transport Area representatives made further significant inputs this discussion during IETF84, including Lars Eggert, Michael Scharf, and David Black. Andrew McGregor, Eric Rescorla, Richard Kelsey, Ed Beroset, Jari Arkko, Zach Shelby, Matthias Kovatsch and many others provided very useful additions.

8. References

8.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2914] Floyd, S., "Congestion Control Principles", BCP 41, RFC 2914, September 2000.

[RFC5405] Eggert, L. and G. Fairhurst, "Unicast UDP Usage Guidelines for Application Designers", BCP 145, RFC 5405, November 2008.

[RFC6298] Paxson, V., Allman, M., Chu, J., and M. Sargent, "Computing TCP's Retransmission Timer", RFC 6298, June 2011.

8.2. Informative References

[I-D.bormann-core-congestion-control]
Bormann, C. and K. Hartke, "Congestion Control Principles for CoAP", draft-bormann-core-congestion-control-02 (work in progress), July 2012.

[I-D.eggert-core-congestion-control]
Eggert, L., "Congestion Control for the Constrained Application Protocol (CoAP)", draft-eggert-core-congestion-control-01 (work in progress), January 2011.

[I-D.ietf-core-coap]
Shelby, Z., Hartke, K., Bormann, C., and B. Frank, "Constrained Application Protocol (CoAP)", draft-ietf-core-coap-11 (work in progress), July 2012.

[I-D.ietf-core-observe]
Hartke, K., "Observing Resources in CoAP", draft-ietf-core-observe-05 (work in progress), March 2012.

[RFC5348] Floyd, S., Handley, M., Padhye, J., and J. Widmer, "TCP Friendly Rate Control (TFRC): Protocol Specification", RFC 5348, September 2008.

[RFC5681] Allman, M., Paxson, V., and E. Blanton, "TCP Congestion Control", RFC 5681, September 2009.

Author's Address

Carsten Bormann
Universitaet Bremen TZI
Postfach 330440
Bremen D-28359
Germany

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