

CoRE Working Group
Internet-Draft
Intended status: Informational
Expires: January 4, 2015

C. Bormann
Universitaet Bremen TZI
A. Betzler
C. Gomez
I. Demirkol
Universitat Politecnica de Catalunya/Fundacio i2CAT
July 03, 2014

CoAP Simple Congestion Control/Advanced
draft-bormann-core-cocoa-02

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 -02, it is making use of input from simulations and experiments in real networks. The specification might still benefit from simplifying it further.

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 <http://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 January 4, 2015.

Internet-Draft

CoAP Simple CoCoA

July 2014

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1.	Introduction	2
1.1.	Terminology	3
2.	Context	3
3.	Advanced CoAP Congestion Control: RTO Estimation	4
3.1.	Blind RTO Estimate	4
3.2.	Measured RTO Estimate	5
3.2.1.	Modifications to the algorithm of RFC 6298	5
3.2.2.	Discussion	6
3.3.	Lifetime, Aging	6
4.	Advanced CoAP Congestion Control: Non-Confirmables	6
4.1.	Discussion	7
5.	IANA Considerations	7
6.	Security Considerations	7
7.	Acknowledgements	7
8.	References	8
8.1.	Normative References	8
8.2.	Informative References	8
	Authors' Addresses	9

[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

[1.1.](#) Terminology

This specification uses terms from [\[RFC7252\]](#). In addition, it defines the following terminology:

Initiator: The endpoint that sends the message that initiates an exchange. E.g., the party that sends a confirmable message, or a non-confirmable message conveying a request.

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.

The present specification is based on the approved text in the [\[RFC7252\]](#) base specification. It is making use of the text that permits advanced congestion control mechanisms and allows them to change protocol parameters, including NSTART and the binary exponential backoff mechanism. Note that [Section 4.8 of \[RFC7252\]](#) limits the leeway that implementations have in changing the CoRE protocol parameters.

The present specification also assumes that, outside of exchanges, non-confirmable messages can only be used at a limited rate without an advanced congestion control mechanism (this is mainly relevant for -observe). It is also intended to address the [\[RFC5405\]](#) guideline about combining congestion control state for a destination; and to clarify its meaning for CoAP using the definition of an endpoint.

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

[3.](#) Advanced CoAP Congestion Control: RTT Estimation

For an initiator that plans to make multiple requests to one destination endpoint, 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, as proposed in [Section 3.2.1](#). Note that such a mechanism must, during idle periods, decay RTT estimates that are shorter or longer than the basic RTT estimate back to the basic RTT estimate, until fresh measurements become available again, as proposed in [Section 3.3](#).

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. Also, for communications with networks of constrained devices that apply radio duty cycling, large and variable round-trip times are likely to be observed. Servers will only trigger their 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 shorter than 1 s SHOULD therefore use a larger backoff factor for retransmissions to avoid expending all of its retransmissions in the default interval of 2 to 3 s. A proposal for a mechanism with variable backoff factors is presented in [Section 3.2.1](#).

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 RT0 Estimate

The initial RT0 estimate for an endpoint is set to 2 seconds.

If only the initial RT0 estimate is available, the RT0 estimate for each of up to NSTART exchanges started in parallel is set to 2 s times the number of parallel exchanges, e.g. if two exchanges are already running, the initial RT0 estimate for an additional exchange is 6 seconds.

[3.2.](#) Measured RT0 Estimate

The RT0 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, where only the first two retransmissions are considered (the "weak estimator"). For the latter, there is some ambiguity whether a response is based on the initial transmission or the retransmissions. For the purposes of the weak estimator, the time from the initial transmission counts. Responses obtained after the third retransmission are not used to update an estimator.

The overall RT0 estimate is an 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 (1) or to the strong estimator (2), from the estimator that made the most recent contribution:

$$\text{RT0_overall_} := 0.25 * \text{RT0_weak_} + 0.75 * \text{RT0_overall_} \quad (1)$$

$$\text{RT0_overall_} := 0.5 * \text{RT0_strong_} + 0.5 * \text{RT0_overall_} \quad (2)$$

(Splitting this update into the two cases avoids making the contribution of the weak estimator too big in naturally lossy networks.)

3.2.1. Modifications to the algorithm of [RFC 6298](#)

This subsection presents three modifications that must be applied to the algorithm of [[RFC6298](#)] as per this document. The first two recommend new parameter settings. The third one is the variable backoff factor mechanism.

The initial value for each of the two RT0 estimators is 2 s.

For the weak estimator, the factor K (the RTT variance multiplier) is set to 1 instead of 4. This is necessary to avoid a strong increase of the RT0 in the case that the RTTVAR value is very large, which may be the case if a weak RTT measurement is obtained after one or more retransmissions.

If an RT0 estimation is lower than 1 s or higher than 3 s, instead of applying a binary backoff factor in both cases, a variable backoff factor is used. For RT0 estimations below 1 s, the RT0 for a retransmission is multiplied by 3, while for estimations above 3 s, the RT0 is multiplied only by 1.5 (this updated choice of numbers to be verified by more simulations). This helps to avoid that exchanges

with small initial RT0s use up all retransmissions in a short interval of time and exchanges with large initial RT0s may not be able to carry out all retransmissions within MAX_TRANSMIT_WAIT (93 s).

The binary exponential backoff is truncated at 32 seconds. Similar to the way retransmissions are handled in the base specification, they are dithered between $1 \times \text{RT0}$ and $\text{ACK_RANDOM_FACTOR} \times \text{RT0}$.

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 RT0 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 16 times its current value, the RTO estimate is doubled. If an estimator has a value that is higher than 3 s, and it is left without further update for 4 times its current value, the RTO estimate is set to be

$$\text{RTO_overall_} := 1 \text{ s} + (0.5 * \text{RTO_overall_})$$

(Note that, instead of running a timer, it is possible to implement these RTO aging calculations cumulatively at the time the estimator is used next.)

[4.](#) Advanced CoAP Congestion Control: Non-Confirmables

(TO DO: Align this with final consensus on -observe!)

A CoAP endpoint MUST NOT send non-confirmables to another CoAP endpoint at a rate higher than defined by this document. Independent of any congestion control mechanisms, a CoAP endpoint can always send non-confirmables if their rate does not exceed 1 B/s.

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 RTT estimator, as specified in [Section 3](#).
3. The packet rate of non-confirmable messages cannot exceed $1/RTT$, where RTT is the overall RTT 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 $1/RTT$.

[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 [[RFC7252](#)].)

[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 Beroaset, Jari Arkko, Zach Shelby, Matthias Kovatsch and many others provided very useful additions.

in part by the Spanish Government's Ministerio de Economía y Competitividad through projects TEC2009-11453 and TEC2012-32531, and FEDER.

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.
- [RFC7252] Shelby, Z., Hartke, K., and C. Bormann, "The Constrained Application Protocol (CoAP)", [RFC 7252](#), June 2014.

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-observe]
Hartke, K., "Observing Resources in CoAP", [draft-ietf-core-observe-14](#) (work in progress), June 2014.
- [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.

Authors' Addresses

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

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

August Betzler
Universitat Politecnica de Catalunya/Fundacio i2CAT
Departament d'Enginyeria Telematica
C/Jordi Girona, 1-3
Barcelona 08034
Spain

Email: august.betzler@entel.upc.edu

Carles Gomez
Universitat Politecnica de Catalunya/Fundacio i2CAT
Escola d'Enginyeria de Telecomunicacio i Aeroespacial
de Castelldefels
C/Esteve Terradas, 7
Castelldefels 08860
Spain

Phone: +34-93-413-7206
Email: carlesgo@entel.upc.edu

Ilker Demirkol
Universitat Politecnica de Catalunya/Fundacio i2CAT
Departament d'Enginyeria Telematica
C/Jordi Girona, 1-3
Barcelona 08034
Spain

Email: ilker.demirkol@entel.upc.edu

