Workgroup: 6lo Working Group
Internet-Draft: draft-gomez-core-coap-bp-00
Published: March 2024
Intended Status: Standards Track
Expires: 2 September 2024
Authors: C. Gomez A. Calveras
UPC UPC

Constrained Application Protocol (CoAP) over Bundle Protocol (BP)

Abstract

The Bundle Protocol (BP) was designed to enable end-to-end communication in challenged networks. The Constrained Application Protocol (CoAP), which was designed for constrained-node networks, may be a suitable application-layer protocol for the scenarios where BP is used. This document specifies how CoAP is carried over BP.

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 2 September 2024.

Table of Contents

1. Introduction

The Delay-Tolerant Networking (DTN) architecture has been designed to enable communication in challenged networks, which are characterized by long delays, intermittent connectivity, and high error rates, among other constraints [RFC4838][RFC7228]. DTN was mainly intended for deep space communication (e.g., to enable an Interplanetary Internet). However, it is also applicable to enable communication on Earth in environments exhibiting relatively similar features, such as sensor networks or temporarily disconnected areas.

The Bundle Protocol (BP) is the fundamental component of DTN. BP is a message-oriented protocol that operates as a store-carry-forward overlay atop the transport-layer protocols of a number of constituent networks [RFC9171]. The protocol data unit of BP is called a bundle. Application-layer functionality runs atop BP.

The Constrained Application Protocol (CoAP) is an application-layer protocol that was specifically designed for constrained-node networks [RFC7252][RFC7228], which are typical in Internet of Things (IoT) scenarios. Such environments are often characterized by

significantly constrained node and network features, including low computational capacity, limited energy availability (which often leads to the use of duty-cycled links), low bandwidth, high latency, and high loss rates. Accordingly, CoAP offers several features, which are also suitable for DTN, including lightweight operation, asynchronous message exchanges, and a significant degree of flexibility, based on RESTful principles.

The present document specifies how CoAP is carried over BP.

2. Terminology

2.1. Requirements language

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 BCP14 [RFC2119], [RFC8174], when, and only when, they appear in all capitals, as shown here.

2.2. Background on previous specifications

The reader is expected to be familiar with the terms and concepts defined by the DTN main specifications (e.g., [RFC4838], [RFC9171], and [RFC9172]), and the CoAP main specifications (e.g., [RFC7252], [RFC7641], [RFC7959], [RFC8323], and [RFC9177]).

3. Architecture

Figure 1 illustrates the protocol stack model for CoAP over BP. (Note: this figure is the same as Figure 1 of RFC 9171, except for the indication of CoAP's location in the protocol stack model.) In this model, CoAP entities exchange application-layer messages carried by BP over an end-to-end path composed of a number of constituent networks.

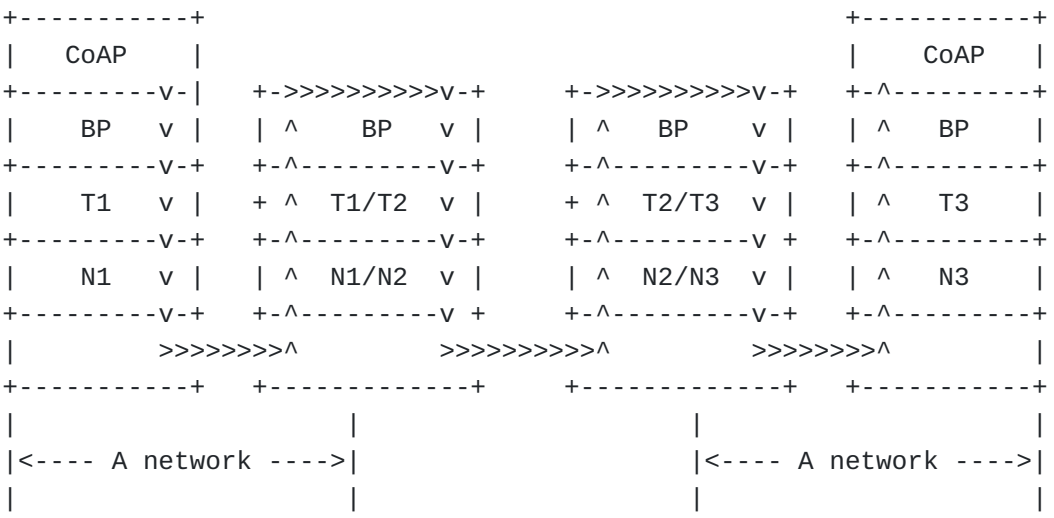

Figure 1: BP and CoAP in the protocol stack model

4. Messages

4.1. Messaging model

The CoAP base specification was produced assuming UDP as the underlying transport-layer protocol [RFC 7252]. Like UDP, BP is a message-oriented protocol. Furthermore, BP does not provide bundle retransmission. Therefore, when CoAP is used over BP, the same messaging model defined for CoAP in RFC 7252 is used, and the CoAP signaling messages defined in RFC 8323 (which are intended for use over reliable transports) MUST NOT be used.

Figure 2 shows the two-sublayer structure of CoAP, when used over BP.

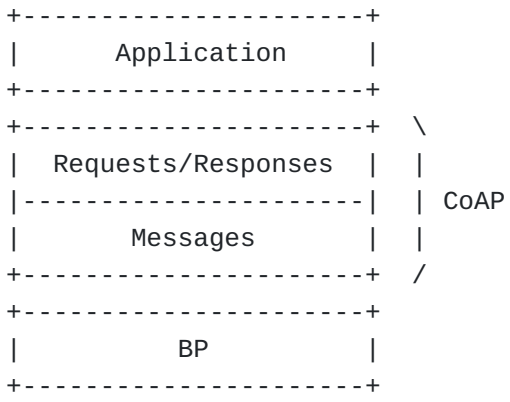

Figure 2: Abstract Layering of CoAP over BP

CoAP follows a client/server model, whereby a client may request an action on a resource on a server. Upon receipt of a request, the server sends a response, including a response code, which may also include a resource representation. Requests and responses are encapsulated in messages.

CoAP defines four message types: Confirmable (CON), Non-confirmable (NON), Acknowledgment (ACK), and Reset (RST). CON messages elicit ACKs, whereas NON messages do not. For CON messages, CoAP uses stop-and-wait retransmission with exponential back-off. A RST message is sent by a CoAP endpoint that has received a message but is unable to process it.

When CoAP is used over BP, a source bundle node MAY set the "request reporting of bundle delivery" flag in the bundle's status report request field of a bundle that encapsulates a CoAP CON message. Upon receipt of a bundle that carries a CoAP CON message with the "request reporting of bundle delivery" flag set, the receiver MAY opt to only send the corresponding bundle delivery status report and omit sending a bundle encapsulating a CoAP ACK message, if and only if it is not possible to transmit a piggybacked response (e.g., because the response is not ready at the moment, or because the CON message does not elicit a response). In that case, if the CoAP CON message sender receives the status report sent in response to its bundle-encapsulated CON message, it MUST assume that the status report serves as CoAP ACK for the CON message.

(Note: the assumption is that the status report size is shorter than the size of a bundle encapsulating a CoAP ACK message that does not carry a payload. To be further confirmed.)

4.2. Message format

In order to transmit a CoAP message over BP, the CoAP message MUST be carried as the block-type-specific data field of the Bundle Payload Block (block type 1) of an encapsulating bundle.

The CoAP message format for CoAP over BP (Figure 4) is the same as the CoAP message format defined in RFC 7252 (Figure 3), except for the Message ID size, which is increased to 24 bits for CoAP over BP. The reason for this change is avoiding a severe limitation on the number of messages a sender can send per time unit, considering the latency values in the environments where CoAP over BP may be used, and that, as stated in RFC 7252, "the same Message ID MUST NOT be reused (in communicating with the same endpoint) within the EXCHANGE_LIFETIME". See Appendix B for further details.

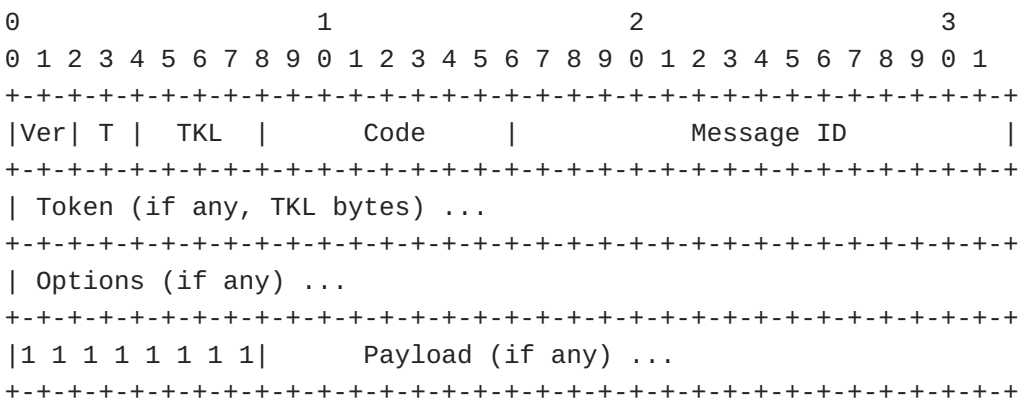

Figure 3: CoAP Message Format as defined in RFC 7252

Figure 4: CoAP Message Format over BP

5. CoAP parameter settings and related times

This section discusses the main CoAP parameters and times that are relevant in the environments where BP may be used. (Note that the complete set of parameters, assumptions, default values, and related times in CoAP can be found in Section 4.8 of RFC7252.)

As a congestion control measure, the maximum number of outstanding interactions between a client and a given server is limited to NSTART, which is set to a default value of 1. A greater value for NSTART can be used only when mechanisms that ensure congestion control safety are used.

The main parameters related with CON messages are indicated next.

ACK_TIMEOUT and ACK_RANDOM_FACTOR. These two parameters determine the duration of the initial retransmission timeout, which is set to a randomly chosen value between ACK_TIMEOUT and ACK_TIMEOUT * ACK_RANDOM_FACTOR. The default values for ACK_TIMEOUT and ACK_RANDOM_FACTOR are 2 s and 1.5, respectively. Therefore, the default initial retransmission timeout in CoAP is between 2 and 3 s.

For CoAP over BP, ACK_TIMEOUT should be set to a value of at least the expected RTT, which may be of an order of magnitude several times greater than the default one (see Appendix A).

ACK_RANDOM_FACTOR needs to be at least equal to or greater than 1.0. The default value of 1.5 is intended to avoid synchronization effects among different senders when RTTs are in the order of seconds. However, the greater latency in delay-tolerant environments may reduce the risk of synchronization effects therein. In such case, a lower ACK_RANDOM_FACTOR may help reduce total message delivery latency when retries are performed.

MAX_RETRANSMIT. This parameter defines the maximum number of retries for a given CON message. The default value for this parameter is 4. Since there is an exponential back-off between retransmissions, and considering the delay values in environments where BP is used, it may be suitable to set this parameter to a value lower than the default one (see Appendix A).

The following assumptions on the characteristics of the network and the nodes need to be considered:

MAX_LATENCY is the maximum time a datagram is expected to take from the start of its transmission to the completion of its reception. In RFC 7252, this value is arbitrarily set to 100 s, which is close to the historic Maximum Segment Lifetime (MSL) of 120 s defined in the TCP specification [RFC9293]. However, such value assumes communication in non-challenged environments. Therefore, in environments where BP is used, MAX_LATENCY may need to be increased by at least 2-3 orders of magnitude.

PROCESSING_DELAY is the time since a node receives a CON message until it transmits an ACK in response. In RFC 7252, this value is assumed to be of at most the default ACK_TIMEOUT value of 2 s. For the sake of limiting latency, it is assumed that the same value can be used also in environments where BP is used.

A relevant CON message derived time is EXCHANGE_LIFETIME. This time indicates the maximum possible time since a CON message is sent for the first time, until ACK reception (which may potentially occur after several retries). EXCHANGE_LIFETIME includes the following components: the total time since the first transmission attempt of a CON message until the last one (called MAX_TRANSMIT_SPAN in RFC 7252), a MAX_LATENCY for the CON, PROCESSING_DELAY, and a MAX_LATENCY for the ACK. The default value for EXCHANGE_LIFETIME is 247 s. However, in challenged environments (e.g., deep space), and considering the increased values for protocol parameters and network characteristics described above, EXCHANGE_LIFETIME will be at least

2 (and perhaps a greater number of) orders of magnitude greater than the default one (see Appendix A).

The main time related with NON messages is NON_LIFETIME. This is the time since a NON message is transmitted until its Message ID can be safely reused. This time is actually equal to MAX_LATENCY, therefore its default value is 100 s. However, as described earlier, in challenged environments (e.g, deep space) it may need to be increased by 2-3 orders of magnitude.

Note that CoAP implementations may also need to be adapted if they have been designed to use 8-bit timers to handle CON or NON message lifetimes (e.g., to retire Message IDs) in seconds.

6. Observe

The CoAP Observe Option allows a server to send notifications carrying a representation of the current state of a resource to interested clients called observers [RFC7641]. The latter need to initially register at a specific server that they are interested in being notified whenever the resource state changes.

Observe generally provides significant performance benefits, since, after the registration, the client does not have to send a request to receive a notification. This feature is particularly beneficial in environments where end-to-end latency is high, and energy and bandwidth resources may be constrained.

As per the Observe specification, when the time between the two last notifications received by a CoAP client is greater than 128 seconds, it can be concluded that the last one received is also the latest sent by the server. The duration of 128 seconds was chosen as a number greater than the default MAX_LATENCY value of the base CoAP specification. When CoAP is used over BP, the duration of 128 seconds may be insufficient in many scenarios. In such cases, the duration needs to be chosen as a value greater than the MAX_LATENCY of the scenario (see Appendix A).

7. Block-wise transfers

CoAP supports functionality that allows carrying large payloads by means of block-wise transfers [RFC7959], [RFC9177]. BP also supports fragmentation and reassembly functionality. RFC 7959 states, in the context of fragmentation and reassembly functionality being available at several protocol stack layers, that "the fragmentation/reassembly process burdens the lower layers with conversation state that is better managed in the application layer". However, an implicit assumption in RFC 7959 is that details on the data unit sizes that can be carried over the different links of an end-to-end path are known in advance by the sender.

When CoAP is used over BP, CoAP block-wise transfers MAY be used if the source knows in advance the duration and type of expected contacts (e.g., scheduled or predicted) between the BP nodes that will forward the bundles from the source bundle node to the destination bundle node. This does not preclude the use of BP fragmentation and reassembly when deemed necessary.

There exist two CoAP specifications that allow to perform block-wise transfers: [RFC7959] and [RFC9177].

As per RFC 7959, a CoAP endpoint can only ask for (or send) the next block after the previous block has been transferred. Furthermore, RFC 7959 recommends the use of CON messages. Therefore, communication follows a stop-and-wait pattern, which is not suitable for environments with long delays.

RFC 9177 is particularly suitable for DTN environments, as it enables block-wise transfers using NON messages. Thus, blocks can be transmitted serially without having to wait for a response or next request from the remote CoAP peer. Recovery of multiple missing blocks (which can be reported at once in a single CoAP message) is also supported.

7.1. Main CoAP block-wise transfer parameters

The following new parameters are defined by RFC 9177, for use with NON messages and the Q-Block1 and Q-Block2 options: MAX_PAYLOADS, NON_TIMEOUT, NON_TIMEOUT_RANDOM, NON_RECEIVE_TIMEOUT, NON_MAX_RETRANSMIT, NON_PROBING_WAIT, and NON_PARTIAL_TIMEOUT.

MAX_PAYLOADS indicates the number of consecutive blocks an endpoint can transmit without eliciting a message from the other endpoint. The default value defined for this parameter is 10, which is in line with the initial window size currently defined for TCP [RFC6928].

TO-DO: MAX_PAYLOADS for deep space?

NON_TIMEOUT is the minimum time between sending two consecutive sets of MAX_PAYLOADS blocks that correspond to the same body. The actual time between sending two consecutive sets of MAX_PAYLOADS blocks is called NON_TIMEOUT_RANDOM, which is calculated as NON_TIMEOUT * ACK_RANDOM_FACTOR. In RFC 9177, NON_TIMEOUT is defined as having the same value as ACK_TIMEOUT. ACK_RANDOM_FACTOR is set to 1.5, following RFC 7252. As a result, by default, NON_TIMEOUT_RANDOM is equal to a randomly chosen value between 2 and 3 s.

The NON_TIMEOUT_RANDOM inactivity interval described above is introduced to avoid causing congestion due to the transmission of MAX_PAYLOADS itself. As discussed previously, in challenged networks, ACK_TIMEOUT should be set to a value greater than default.

When CoAP is used in deep space, NON_TIMEOUT, and thus NON_TIMEOUT_RANDOM, need to be adjusted considering the characteristics of the end-to-end path, independent of ACK_TIMEOUT.

NON_RECEIVE_TIMEOUT is the initial time that a receiver will wait for a missing block within MAX_PAYLOADS before requesting retransmission for the first time. Every time the missing payload is re-requested, the time to wait value doubles. NON_RECEIVE_TIMEOUT has a default value of 2*NON_TIMEOUT. As described earlier, in challenged networks, NON_TIMEOUT needs to be adjusted considering the characteristics of the end-to-end path.

NON_MAX_RETRANSMIT is the maximum number of times a request for the retransmission of missing payloads can occur without a response from the remote peer. By default, NON_MAX_RETRANSMIT has the same value as MAX_RETRANSMIT (Section 4.8 of [RFC7252]). Accordingly, when CoAP is used in deep space, the same considerations regarding MAX_RETRANSMIT in Section 5 apply to NON_MAX_RETRANSMIT as well. That is, when CoAP is used in space, while the default value for this parameter is 4, it may be suitable to set this parameter to a value lower than the default one.

8. CoAP over BP URI

Previous specifications have defined various URI schemes for identifying CoAP resources and providing a means of locating the resources. Such URI schemes are the following: "coap" and "coaps", defined in [RFC 7252]; and "coap+tcp", "coaps+tcp", "coap+ws", and "coaps+ws", defined in [RFC 8323].

This document introduces an additional URI scheme:

o The "coap+bp" URI scheme for CoAP over BP.

In this section, the syntax for the URI schemes is specified using the Augmented Backus-Naur Form (ABNF) [RFC5234]. The definitions of "host", "port", "path-abempty", and "query" are adopted from [RFC3986].

As with the "coap" and "coaps" schemes defined in [RFC7252], and the "coap+tcp", "coaps+tcp", "coap+ws", and "coaps+ws" schemes defined in [RFC8323], the URI scheme defined in this section also supports the path prefix "/.well-known/" as defined by [RFC5785] for "well-known locations" in the namespace of a host. This enables discovery as per Section 7 of [RFC7252].

8.1. coap+bp URI Scheme

The "coap+bp" URI scheme identifies CoAP resources that are intended to be accessible using CoAP over BP.

coap-bp-URI = "coap+bp:" "//" endpoint_ID path-abempty ["?" query]

The syntax defined in Section 6.1 of [RFC7252] applies to this URI scheme, except that a BP endpoint ID (expressed as "endpoint_ID" above) is used instead of the "host" and "port" authority subcomponents.

Encoding considerations: The scheme encoding conforms to the encoding rules established for URIs in [RFC3986].

Interoperability considerations: None.

Security considerations: See Section 11.1 of [RFC7252].

9. IANA Considerations

IANA is requested to register the Uniform Resource Identifier (URI) scheme "coap+bp". This registration request complies with [RFC7595].

Scheme name: coap+bp

Status: Permanent

Applications/protocols that use this scheme name: The scheme is used by CoAP endpoints to access CoAP resources using BP.

Contact: IETF chair (chair@ietf.org)

Change controller: IESG (iesg@ietf.org)

Reference: Section 8.1 in [RFCthis]

10. Security Considerations

TO-DO

11. Acknowledgments

Carles Gomez and Anna Calveras have been funded in part by the Spanish Government through project PID2019-106808RA-I00, and by Secretaria d'Universitats i Recerca del Departament d'Empresa i Coneixement de la Generalitat de Catalunya 2021 throught grant SGR 00330.

12. References

12.1. 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>.

[RFC4838] Cerf, V., Burleigh, S., Hooke, A., Torgerson, L., Durst, R., Scott, K., Fall, K., and H. Weiss, "Delay-Tolerant Networking Architecture", RFC 4838, DOI 10.17487/RFC4838, April 2007, <https://www.rfc-editor.org/info/rfc4838>.

[RFC5234] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, DOI 10.17487/RFC5234, January 2008, <https://www.rfc-editor.org/info/rfc5234>.

[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>.

[RFC7595] Thaler, D., Ed., Hansen, T., and T. Hardie, "Guidelines and Registration Procedures for URI Schemes", BCP 35, RFC 7595, DOI 10.17487/RFC7595, June 2015, <https://www.rfc-editor.org/info/rfc7595>.

[RFC7641] Hartke, K., "Observing Resources in the Constrained Application Protocol (CoAP)", RFC 7641, DOI 10.17487/RFC7641, September 2015, <https://www.rfc-editor.org/info/rfc7641>.

[RFC7959] Bormann, C. and Z. Shelby, Ed., "Block-Wise Transfers in the Constrained Application Protocol (CoAP)", RFC 7959, DOI 10.17487/RFC7959, August 2016, <https://www.rfc-editor.org/info/rfc7959>.

[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>.

[RFC8323] Bormann, C., Lemay, S., Tschofenig, H., Hartke, K., Silverajan, B., and B. Raymor, Ed., "CoAP (Constrained Application Protocol) over TCP, TLS, and WebSockets", RFC 8323, DOI 10.17487/RFC8323, February 2018, <https://www.rfc-editor.org/info/rfc8323>.

[RFC9171] Burleigh, S., Fall, K., and E. Birrane, III, "Bundle Protocol Version 7", RFC 9171, DOI 10.17487/RFC9171, January 2022, <https://www.rfc-editor.org/info/rfc9171>.

[RFC9172] Birrane, III, E. and K. McKeever, "Bundle Protocol Security (BPSec)", RFC 9172, DOI 10.17487/RFC9172, January 2022, <https://www.rfc-editor.org/info/rfc9172>.

[RFC9177] Boucadair, M. and J. Shallow, "Constrained Application Protocol (CoAP) Block-Wise Transfer Options Supporting Robust Transmission", RFC 9177, DOI 10.17487/RFC9177, March 2022, <https://www.rfc-editor.org/info/rfc9177>.

12.2. Informative References

[Conf] S.M. Davidovich, J. Whittington, "Concept for continuous inter-planetary communications", May 1999.

[RFC7228] Bormann, C., Ersue, M., and A. Keranen, "Terminology for Constrained-Node Networks", RFC 7228, DOI 10.17487/RFC7228, May 2014, <https://www.rfc-editor.org/info/rfc7228>.

Appendix A. Reference CoAP parameter values for interplanetary communication

Figure 5 shows the Round-Trip Time (RTT) between two endpoints on (or close to) different celestial bodies of the Solar System, for the maximum distances between such endpoints [Conf], and in an idealized scenario where communication latency only comprises a propagation delay component. (Note that message storing until the next connectivity opportunity may significantly increase total communication latency.) The RTT also provides a lower bound on (and an approximation of) the ACK_TIMEOUT values required to avoid spurious retransmission timer expiration.

Figure 6 provides approximate EXCHANGE_LIFETIME values that would stem from the use of ACK_TIMEOUT values such as those shown in Figure 5, for MAX_RETRANSMIT=1. (Note that the values provided in Figure 5 are also approximately equal to EXCHANGE_LIFETIME, for MAX_RETRANSMIT=0, under the conditions considered.)

For the sake of comparison, Figure 7 also provides the hypothetical, approximate EXCHANGE_LIFETIME values that would correspond to MAX_RETRANSMIT= 1, but with a retransmission scheme using a constant RTO value for message retries.

Finally, Figure 8 provides the one-way delay for communication between endpoints on (or close to) different celestial bodies of the Solar System, for the maximum distances between such endpoints, and assuming an idealized scenario where communication latency only comprises a propagation delay component. The values in this figure correspond approximately to MAX_LATENCY in the described scenarios.

RTT, ACK_TIMEOUT (or EXCHANGE_LIFETIME, for MAX_RETRANSMIT=0)								
	Sun	Mercury	Venus	Earth	Mars	Jupiter	Saturn	Uranus	Neptune
Sun	-	466	727	1,014	1,661	5,444	10,007	20,214	30,288
Mercury	-	-	1,181	1,448	1,968	5,751	10,340	20,548	30,554
Venus	-	-	-	1,735	2,382	6,158	10,741	20,948	30,955
Earth	-	-	-	-	2,642	6,424	11,008	21,215	31,222
Mars	-	-	-	-	-	6,805	11,408	21,615	31,622
Jupiter	-	-	-	-	-	-	14,944	25,151	35,425
Saturn	-	-	-	-	-	-	-	29,220	39,961
Uranus	-	-	-	-	-	-	-	-	50,168
Neptune	-	-	-	-	-	-	-	-	-

Figure 5: ACK_TIMEOUT or EXCHANGE_LIFETIME (for MAX_RETRANSMIT=0), expressed in seconds.

EXCHANGE_LIFETIME (for MAX_RETRANSMIT=1)									
	Sun	Mercury	Venus	Earth	Mars	Jupiter	Saturn	Uranus	Neptune
Sun	-	1,397	2,182	3,042	4,983	16,331	30,021	60,642	90,863
Mercury	-	-	3,542	4,343	5,904	17,252	31,021	61,643	91,663
Venus	-	-	-	5,204	7,145	18,473	32,222	62,843	92,864
Earth	-	-	-	-	7,925	19,273	33,023	63,644	93,665
Mars	-	-	-	-	-	20,414	34,224	64,845	94,866
Jupiter	-	-	-	-	-	-	44,831	75,452	106,274
Saturn	-	-	-	-	-	-	-	87,661	119,883
Uranus	-	-	-	-	-	-	-	-	150,504
Neptune	-	-	-	-	-	-	-	-	-

Figure 6: EXCHANGE_LIFETIME (for MAX_RETRANSMIT=1), expressed in seconds.

EXCHANGE_LIFETIME (for MAX_RETRANSMIT=1 and no exponential backoff)									
	Sun	Mercury	Venus	Earth	Mars	Jupiter	Saturn	Uranus	Neptune
Sun	-	931	1,454	2,028	3,322	10,888	20,014	40,428	60,575
Mercury	-	-	2,362	2,895	3,936	11,501	20,681	41,095	61,109
Venus	-	-	-	3,469	4,763	12,315	21,482	41,896	61,909
Earth	-	-	-	-	5,284	12,849	22,015	42,429	62,443
Mars	-	-	-	-	-	13,609	22,816	43,230	63,244
Jupiter	-	-	-	-	-	-	29,887	50,301	70,849
Saturn	-	-	-	-	-	-	-	58,440	79,922
Uranus	-	-	-	-	-	-	-	-	100,336
Neptune	-	-	-	-	-	-	-	-	-

Figure 7: Hypothetical EXCHANGE_LIFETIME (for MAX_RETRANSMIT=1), assuming CoAP message retransmission without exponential backoff, expressed in seconds.

MAX_LATENCY									
	Sun	Mercury	Venus	Earth	Mars	Jupiter	Saturn	Uranus	Neptune
Sun	-	233	364	507	831	2,722	5,003	10,107	15,144
Mercury	-	-	590	724	984	2,875	5,170	10,274	15,277
Venus	-	-	-	867	1,191	3,079	5,370	10,474	15,477
Earth	-	-	-	-	1,321	3,212	5,504	10,607	15,611
Mars	-	-	-	-	-	3,402	5,704	10,807	15,811
Jupiter	-	-	-	-	-	-	7,472	12,575	17,712
Saturn	-	-	-	-	-	-	-	14,610	19,980
Uranus	-	-	-	-	-	-	-	-	25,084
Neptune	-	-	-	-	-	-	-	-	-

Figure 8: Approximate MAX_LATENCY, expressed in seconds.

Appendix B. Message ID size, EXCHANGE_LIFETIME, and maximum CoAP message rate

With default settings [RFC 7252], and a 16-bit message ID size, CoAP supports the transmission of up to 265 messages/s between a sender and its destination endpoint. If CoAP is used in scenarios involving much greater latencies (e.g., deep space), the greater EXCHANGE_LIFETIME would significantly limit the CoAP message rate. Figure 9 provides the maximum possible message rates for message ID sizes of 16 and 24 bits, and a range of EXCHANGE_LIFETIME values.

Message ID	16 bits	24 bits
#Messages per EXCHANGE_LIFETIME	65,536	16,777,216

Message rate (messages/second)		
EXCHANGE_LIFETIME (s)	Message ID_16 bits	Message_ID 24 bits
247 (default)	265.3 (default)	67,924
500	131.1	33,554
1,000	65.5	16,777
1,500	43.7	11,184
2,000	32.8	8,388
2,500	26.2	6,710
3,000	21.8	5,592
3,500	18.7	4,793
4,000	16.4	4,194
4,500	14.6	3,728
5,000	13.1	3,355
5,500	11.9	3,050
6,000	10.9	2,796
6,500	10.1	2,581
7,000	9.4	2,396
7,500	8.7	2,237
10,000	6.6	1,677
20,000	3.3	838
30,000	2.2	559
40,000	1.6	419
50,000	1.3	335
60,000	1.1	279
70,000	0.9	239
80,000	0.8	209
90,000	0.7	186
100,000	0.7	167
110,000	0.6	152
120,000	0.5	139
130,000	0.5	129
140,000	0.5	119
150,000	0.4	111

Figure 9: Maximum CoAP message rate imposed by the Message ID size and EXCHANGE_LIFETIME, expressed in messages/s.

Authors' Addresses

Carles Gomez
UPC
C/Esteve Terradas, 7
08860 Castelldefels
Spain

Email: carles.gomez@upc.edu

Anna Calveras
UPC
C/Jordi Girona, 1-3
08034 Barcelona
Spain

Email: anna.calveras@upc.edu