CORE
Internet-Draft
Intended status: Standards Track
Expires: October 16, 2020

M. Boucadair
Orange
J. Shallow

April 14, 2020

New Constrained Application Protocol (CoAP) Block-Wise Transfer Options

draft-bosh-core-new-block-00

Abstract

This document specifies new Constrained Application Protocol (CoAP) Block-Wise transfer options: Block3 and Block4 options. These options are similar to the CoAP Block1 and Block2 options, but enable faster transmissions of big blocks of data with less packet interchanges as well as supporting faster recovery should any of the Blocks get lost in transmission.

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 October 16, 2020.

Table of Contents

1. Introduction

1.1. Existing Block-Wise Transfer Options

The Constrained Application Protocol (CoAP) [RFC7252], although inspired by HTTP, was designed to use UDP instead of TCP. The message layer of CoAP over UDP includes support for reliable delivery, simple congestion control, and flow control. [RFC7959] introduced the CoAP Block1 and Block2 options to handle data records that cannot fit in a single IP packet, so not having to rely on IP fragmentation.

The CoAP Block1 and Block2 options work well in environments where there are no or minimal packet losses. They operate synchronously where each block has to be requested and can only ask for (or send) the next block when the request for the previous Block has completed.

Packet, and hence Block transmission rate, is controlled by Round Trip Times.

There is a requirement for these Blocks of data to be transmitted under network conditions where there may be transient packet loss such as when a network is subject to a Distributed Denial Of Service (DDoS) attack and there is a need for DDoS mitigation agents need to communicate with each other (e.g., [I-D.ietf-dots-telemetry]). As a reminder, [RFC7959] recommends use of Confirmable (CON) responses to handle potential packet loss; which does not work with a flooded pipe DDoS situation.

1.2. New Block-Wise Transfer Options

This document introduces the CoAP Block3 and Block4 options. These options are similar in operation to the CoAP Block1 and Block2 options respectively, but enable faster transmissions of big blocks of data with less packet interchanges as well as supporting faster recovery should any of the Blocks get lost in transmission.

The faster transmissions occur as all the Blocks can be transmitted serially (as are IP fragmented packets) without having to wait for an acknowledgement from the remote CoAP peer. Recovery of missing Blocks is faster in that multiple missing Blocks can be requested in a single packet.

Non-Confirmable (NON) request usage of Block3 and Non-Confirmable response usage of Block4 enable the faster transmissions of the blocks of the body message as there is no need to wait for the responses. Note that the same performance benefits can be applied to Confirmable messages if the value of NSTART is increased from 1 (Section 4.7 of [RFC7252]). Some sample examples with Confirmable messages are provided in Appendix A.

A CoAP endpoint can acknowledge all or a subset of the blocks. Concretely, the receiving CoAP endpoint informs the CoAP endpoint sender about all blocks that have been received. The CoAP endpoint sender will then retransmit only the blocks that have been lost in transmission.

Only the deviation from Block1 and Block2 options are specified. Pointers to appropriate [RFC7959] sections are provided.

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

The terms "payload" and "body" are defined in [RFC7959]. The term "payload" is thus used for the content of a single CoAP message (i.e., a single block being transferred), while the term "body" is used for the entire resource representation that is being transferred in a block-wise fashion.

3. The Block3 and Block4 Options

3.1. Properties of Block3 ad Block4 Options

The properties of Block3 and Block4 options are shown in Table 1. The formatting of this table follows the one used in Table 4 of [RFC7252] (Section 5.10). The C, U, N, and R columns indicate the properties Critical, Unsafe, NoCacheKey, and Repeatable defined in Section 5.4 of [RFC7252]. Only C and R columns are marked for Block3 and Block4 options.

Number	C	U	N	R	Name	Format	Length	Default
TBA1	x			x	Block3	uint	0-7	(none)
TBA2	x			x	Block4	uint	0-7	(none)

Table 1: CoAP Block3 and Block4 Option Properties

The Block3 option pertains to the request payload, and the Block4 option pertains to the response payload. The Content-Format option applies to the body, not to the payload.

For the methods defined in [RFC7252] and [RFC8132], Block3 is useful with the payload-bearing POST, PUT, and PATCH requests and their responses. Block4 is useful with GET, POST, PUT, and FETCH requests and their payload-bearing responses (2.01, 2.02, 2.04, and 2.05) (Section 5.5 of [RFC7252]).

To indicate support for Block4 responses, the CoAP client MUST include the Block4 option in a GET or FETCH request so that the server knows that the client supports this functionality. Otherwise, the server would use the Block2 option (if supported) to send back a message body that is larger than can fit into a single IP packet [RFC7959].

Where Block3 option is present in a request or Block4 option in a response (i.e., in that message to the payload of which it pertains), it indicates a block-wise transfer and describes how this specific block-wise payload forms part of the entire body being transferred (referred to as "descriptive usage"). Where it is present in the opposite direction, it provides additional control on how that payload will be formed or was processed (referred to as "control usage").

Implementation of either block option is intended to be optional. However, when it is present in a CoAP message, it MUST be processed (or the message rejected); therefore, it is identified as a Critical option.

The Block3 and Block4 options are safe to forward. That is, a CoAP proxy that does not understand the Block3 and Block4 options should forward the options on.

Both Block3 and Block4 options are repeatable when requesting re-transmissions of missing Blocks but not otherwise. Otherwise, any request carrying multiple Block3 (or Block4) options MUST be handled following the procedure specified in Section 5.4.5 of [RFC7252].

PROBING_RATE parameter in CoAP indicates the average data rate that must not be exceeded by a CoAP endpoint in sending to a peer endpoint that does not respond. The body of blocks will be subjected to probing rate.

3.2. Structure of Block3 and Block4 Options

The structure of Block3 and Block4 options follows the structure defined in Section 2.2 of [RFC7959] with two additional fields:

- The Block ID (BID) which associates all the Blocks that make up the large item of data that is being transferred.

- The "All" bit (called, A-bit) used when acknowledging all the blocks.

As such, five items of information may need to be transferred in a Block3 or Block4 option:

- the size of the block (size exponent (SZX)),

- whether more blocks are following (More (M)),

- the relative number of the block (Block Number (NUM)) within a sequence of blocks with the given size,

- whether this is acknowledging all the blocks successfully received (A), and

- the Block identifier number (BID) that is common to the sequence of blocks with the same given size. The BID is different for each set of sequence of blocks.

The value of the Block3 or Block4 option is a variable-size (0 to 7 byte) unsigned integer (uint) (Section 3.2 of [RFC7252]). This integer value encodes the aforementioned five fields as shown in Figure 1. Note that, due to the CoAP uint-encoding rules, when all of NUM, M, SZX, A, and BID happen to be zero, a zero-byte integer will be sent.

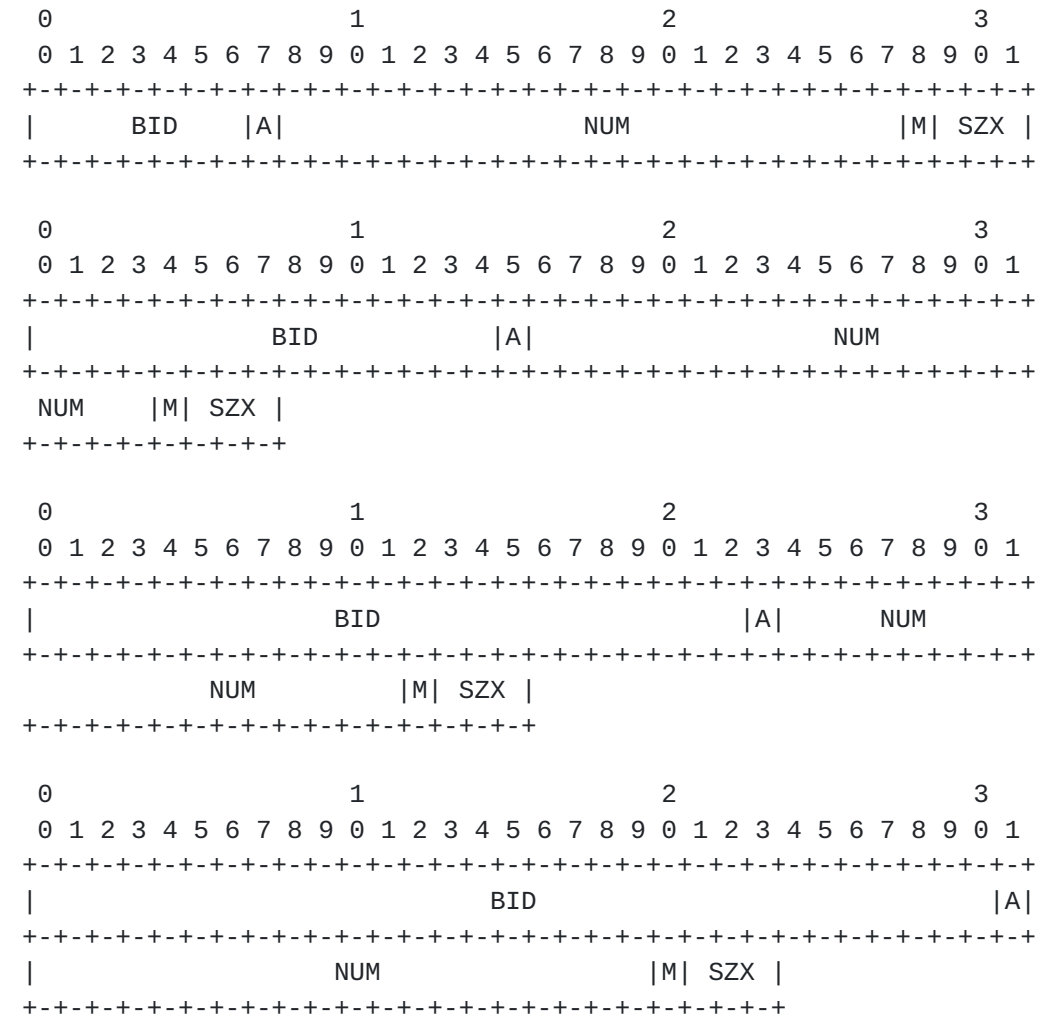

Figure 1: Structure of Block3 and Block4 Options

The twenty-fourth least significant bit is the A-bit ("val & 0x1000000").

The option value shifted right by 25 ("(val >> 25)")(the BID field) is the Block identifier that identifies which sequence of blocks this particular block is in.

The current transfer is about the "size" bytes starting at byte "NUM << (SZX + 4)".

Within the option value of a Block3 or Block4 option, the meaning of the option fields is defined below. Note that the block size (SZX, size exponent), the M-bit, and the NUM fields are defined in Section 2.2 of [RFC7959], but are provided below for the reader's convenience:

A: All Flag. This bit is only set in a response packet to indicate that this response refers to all of the blocks in the body. The A-bit MUST be unset in all other cases.

BID: Block Identifier. This block identifier is the same for all of the blocks in the body of data that is being transferred. It is used when a particular block needs to be re-transmitted.

This value MUST be different for distinct sets of blocks of data and SHOULD be incremented whenever a new body of data is being transmitted for a CoAP session between peers. The initial BID value SHOULD be randomly generated.

NUM: Block Number, indicating the block number being requested or provided. Block number '0' indicates the first block of a body (i.e., starting with the first byte of the body).

M: More Flag ("not last block"). For descriptive usage, this flag, if unset, indicates that the payload in this message is the last block in the body; when set, it indicates that there are one or more additional blocks available.

When a Block4 option is used in a request to retrieve a specific block number ("control usage"), the M-bit MUST be sent as zero and ignored on reception. In a Block3 option in a response, the M-bit is used to indicate atomicity, similar to Block1 option ([RFC7959]).

SZX: Block Size. The block size is represented as a three-bit unsigned integer indicating the size of a block to the power of two. Thus, block size = 2**(SZX + 4). The allowed values of SZX are 0 to 6, i.e., the minimum block size is 2**(0+4) = 16 and the

maximum is 2**(6+4) = 1024. The value 7 for SZX (which would indicate a block size of 2048) is used as a BERT option in [RFC8323].

There is no default value for the Block3 and Block4 options. Absence of one of these options is equivalent to an option value of 0 with respect to the value of NUM, M, A, and BID that could be given in the option, i.e., it indicates that the current block is the first and only block of the transfer (block number is set to 0, M-bit is unset, A-bit is unset, and BID is set to 0). However, in contrast to the explicit value 0, which would indicate an SZX of 0, and thus a size value of 16 bytes, there is no specific explicit size implied by the absence of the option -- the size is left unspecified. (As for any uint, the explicit value 0 is efficiently indicated by a zero-length option; this, therefore, is different in semantics from the absence of the option).

3.3. Working with Observe

As the blocks of the body are sent without waiting for acknowledgement of the individual blocks, the Observe value [RFC7641] MUST be the same for all the blocks of the same body.

Likewise, the Tokens MUST all have the same value for all the blocks of the same body. This is so that if any of the blocks gets lost during transmission (including the first one), the receiving CoAP endpoint can take the appropriate decisions (implementation-specific).

3.4. Working with Size1 and Size2 Options

[RFC7959] defines two CoAP options, Size1 for indicating the size of the representation transferred in requests, and Size2 for indicating the size of the representation transferred in responses.

It is RECOMMENDED that the Size1 option is used with the Block3 option and that the Size2 option is used with the Block4 option.

3.5. Working with Etag Option

The Etag option defined in Section 5.10.6 of [RFC7252] applies to the whole representation of the resource, and thus to the body of the response.

4. Caching Considerations

The Block3 and Block4 options are part of the cache key. As such, a CoAP proxy that does not understand the Block3 and Block4 options must follow the recommendations in Section 5.7.1 of [RFC7252] for caching.

This specification does not require a proxy to obtain the complete representation before it serves parts of it to the client. Otherwise, the considerations discussed in Section 2.10 of [RFC7959] apply for the Block3 and Block4 options (with Block3 substituted for Block1 and Block4 substituted for Block2) for proxies that support Block3 and Block4 options.

A proxy that supports Block3 and Block4 options MUST be prepared to receive a GET message indicating one or more missing blocks. The proxy can serve from its cache missing blocks that are available in its cache. If one more requested blocks are not available locally, the proxy MUST update the GET request with the blocks that it served locally, and then forward the request to the next hop. When the proxy replies from its local cache, it MUST use the same Token value as in the received request.

How long a CoAP endpoint (or proxy) keeps the body in its cache is implementation-specific (e.g., it may be based on Max-Age).

5. HTTP-Mapping Considerations

As a reminder, the basic normative requirements on HTTP/CoAP mappings are defined in Section 10 of [RFC7252]. The implementation guidelines for HTTP/CoAP mappings are elaborated in [RFC8075].

The rules defined in Section 5 of [RFC7959] are to be followed.

6. Examples of Selective Block Recovery

This section provides some sample flows to illustrate the use of Block3 and Block4 options. The following conventions are used in the following sub-sections:

```
   T: Token value
   O: Observe Option value
   M: Message ID
  B3: Block3 option values BID/All/NUM/More/SZX
  B4: Block3 option values BID/All/NUM/More/SZX
   \: Trimming long lines
[[]]: Comments
---X: Message loss
```

6.1. Block3 Option: Non-Confirmable Example

Figure 2 depicts an example of a NON PUT request conveying Block3 option. All the blocks are received by the server; hence the A-bit is set in the 2.05 message sent by the server to the client.

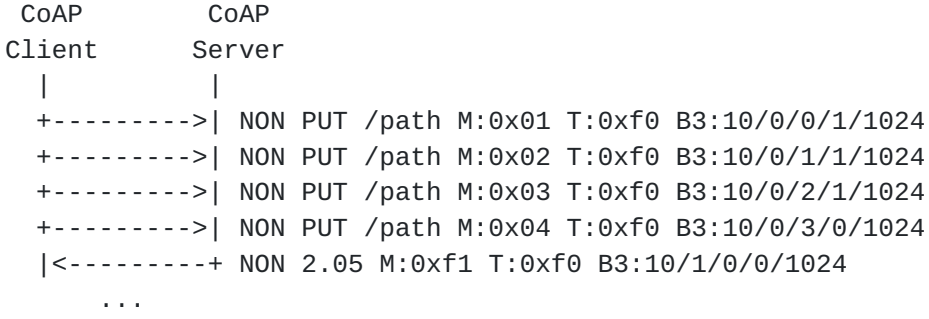

Figure 2: Example of NON Request with Block3 Option (Without Loss)

Consider now a scenario where a new body of data is to be sent by the client, but some blocks are dropped in transmission as illustrated in Figure 3.

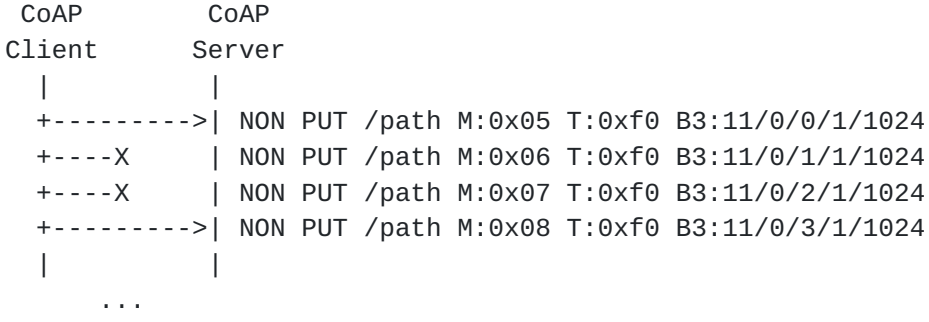

Figure 3: Example of NON Request with Block3 Option (With Loss)

The server realizes that some blocks are missing and asks for the missing ones in one go (Figure 4). It does so by indicating which blocks have been received.

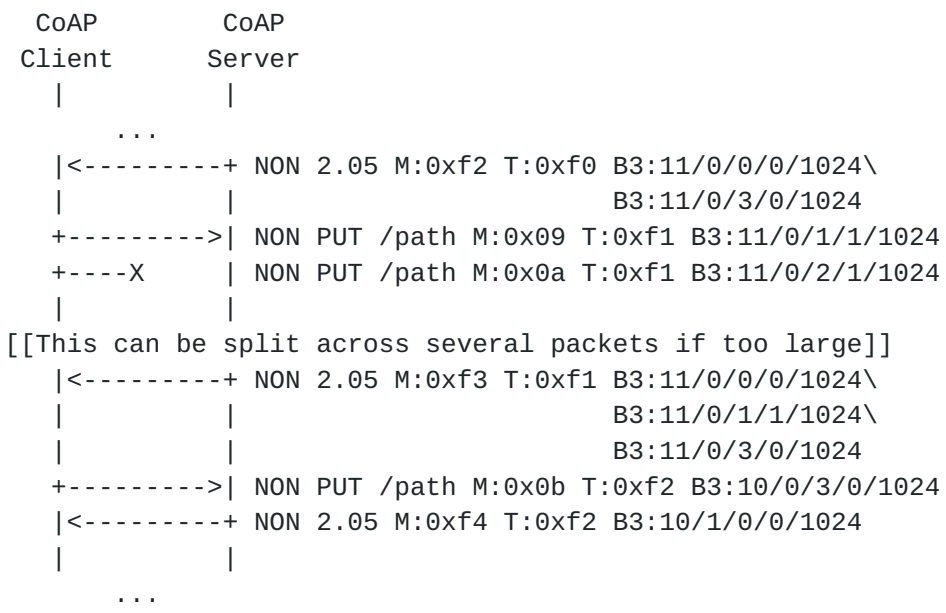

Figure 4: Example of NON Request with Block3 Option (Blocks Recovery)

Under high levels of traffic loss, the client can elect not to retry sending missing blocks of data. This decision is implementation-specific.

6.2. Block4 Option: Non-Confirmable Example

Figure 5 illustrates the example of Block4 option. The client sends a NON GET carrying an Observe and a Block4 options. The Block4 option indciates a size hint (1024 bytes). This request is replied by the serer using four (4) blocks that are transmitetted to the client without any loss. Each of these blocks carries a Block4 option. The same process is repeated when an Observe is triggered, but no loss is experienced by any of of the notification bloks.

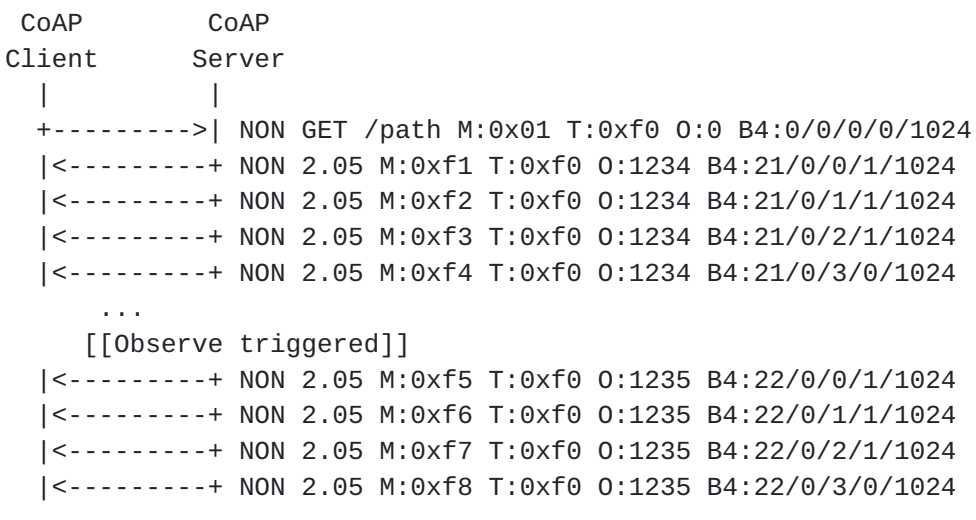

Figure 5: Example of NON Notifications with Block4 Option (Without Loss)

Figure 6 shows the example of an Observe that is triggered but for which some notification blocks are lost. The client detects the missing blocks and request their retransmission. It does so by indicating the blocks that were successfully received.

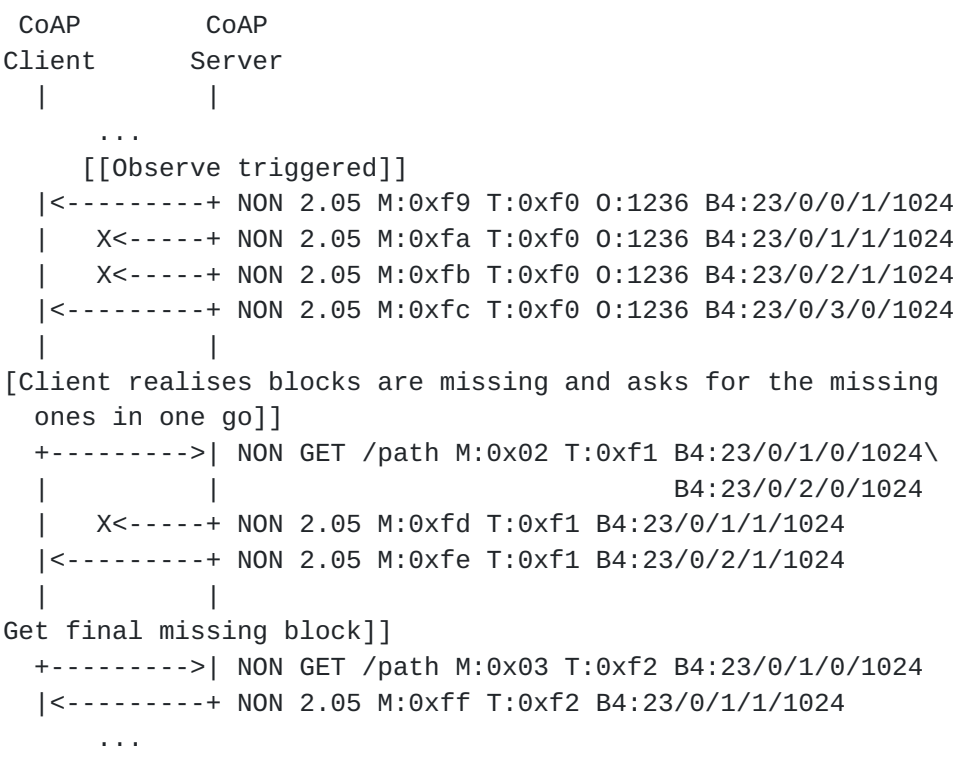

Figure 6: Example of NON Notifications with Block4 Option (Blocks Recovery)

Under high levels of traffic loss, the client can elect not to retry getting missing blocks of data. This decision is implementation-specific.

7. IANA Considerations

IANA is requested to add the following entries to the "CoAP Option Numbers" sub-registry available at https://www.iana.org/assignments/core-parameters/core-parameters.xhtml#option-numbers:

Number	Name	Reference
TBA1	Block3	[RFCXXXX]
TBA2	Block4	[RFCXXXX]

Table 2: CoAP Block3 and Block4 Option Numbers

This document suggests XX and XX as a values to be assigned for the new option numbers.

8. Security Considerations

Security considerations discussed in Section 9 of [RFC7959] should be taken into account.

[[discuss iof any security issues related to the incremental BID values. Lifetime of a BID (pointer to RFC8200)]]

9. Acknowledgements

Thanks to Achim Kraus, Christian Amsuess, Carsten Bormann, Jim Schaad for the comments on the mailing list.

Some text from [RFC7959] is reused for readers convenience.

10. References

10.1. 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>.

[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>.

[RFC8075] Castellani, A., Loreto, S., Rahman, A., Fossati, T., and E. Dijk, "Guidelines for Mapping Implementations: HTTP to the Constrained Application Protocol (CoAP)", RFC 8075, DOI 10.17487/RFC8075, February 2017, <https://www.rfc-editor.org/info/rfc8075>.

[RFC8132] van der Stok, P., Bormann, C., and A. Sehgal, "PATCH and FETCH Methods for the Constrained Application Protocol (CoAP)", RFC 8132, DOI 10.17487/RFC8132, April 2017, <https://www.rfc-editor.org/info/rfc8132>.

[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>.

10.2. Informative References

[I-D.ietf-dots-telemetry] Boucadair, M., Reddy.K, T., Doron, E., and c. chenmeiling, "Distributed Denial-of-Service Open Threat Signaling (DOTS) Telemetry", draft-ietf-dots-telemetry-06 (work in progress), April 2020.

Appendix A. Examples with Confirmable Messages

These examples assume NSTART has been increased to at least 4.

A.1. Block3 Option

Let's now consider the use Block3 option with a CON request as shown in Figure 7. All the blocks are acknowledged (ACK).

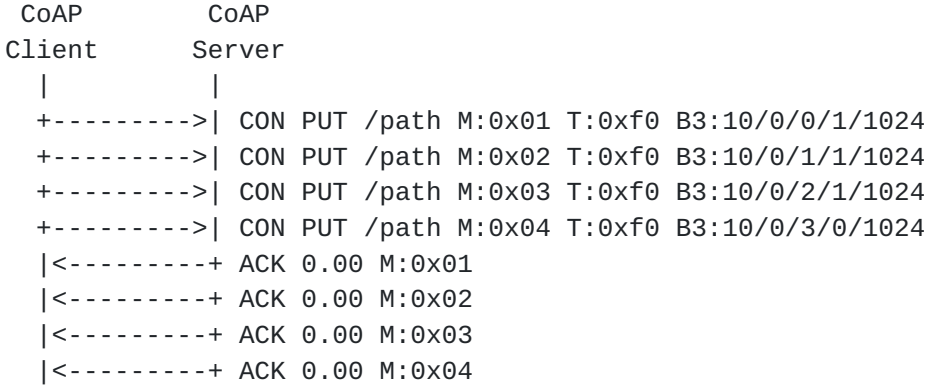

Figure 7: Example of CON Request with Block3 Option (Without Loss)

Now, suppose that a new body of data is to sent but with some blocks dropped in transmission as illustrated in Figure 8. The client will retry sedning blocks for which no ACK was received.

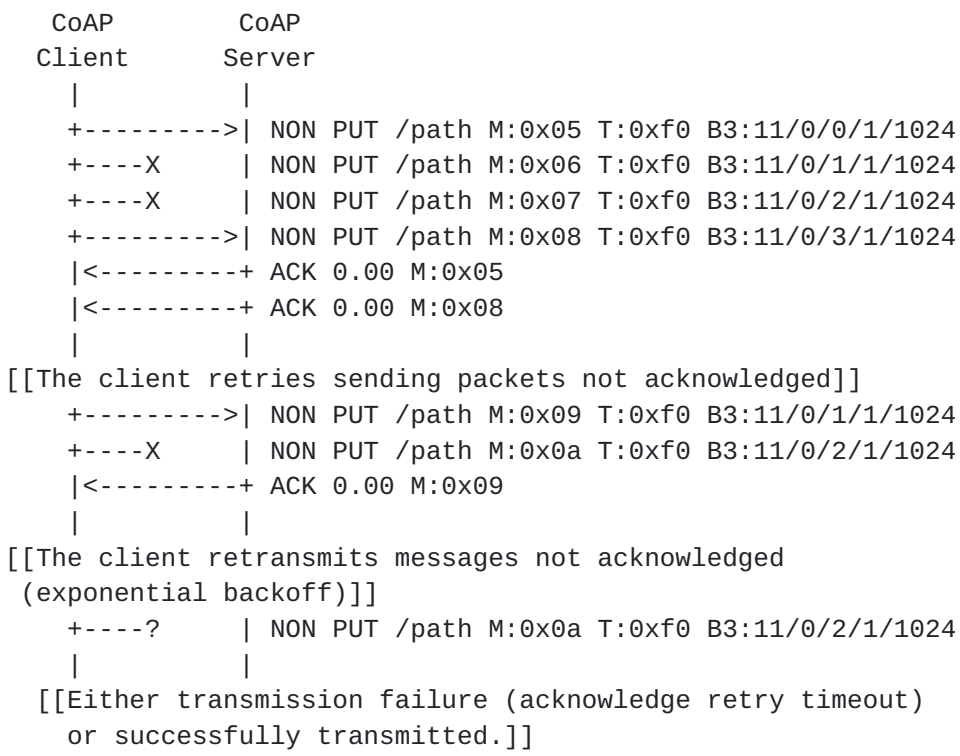

Figure 8: Example of CON Request with Block3 Option (Blocks Recovery)

It is implementation dependent as to whether a CoAP session is terminated following acknowledge retry timeout, or whether the CoAP session continues to be used under such adverse traffic conditions.

If there is likely to be the possibility of network transient losses, then the use of Non-Confirmable traffic should be considered.

A.2. Block4 Option

An exmaple of the use of Block4 option with Confirmable messages is shown in Figure 9.

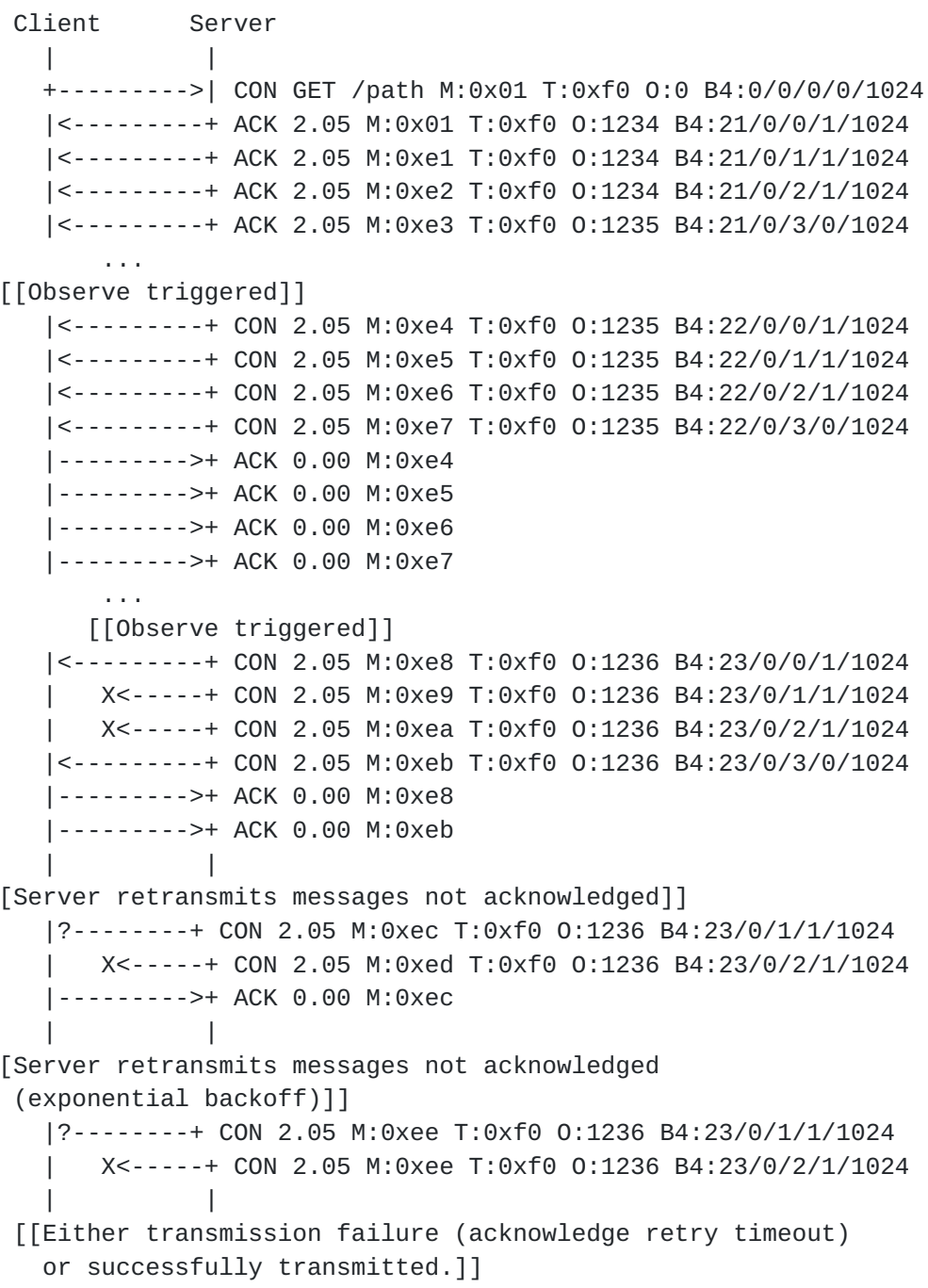

Figure 9: Example of CON Notifications with Block4 Option

It is implementation-dependent as to whether a CoAP session is terminated following acknowledge retry timeout, or whether the CoAP session continues to be used under such adverse traffic conditions.

If there is likely to be the possibility of network transient losses, then the use of Non-Confirmable traffic should be considered.

Authors' Addresses

Mohamed Boucadair
Orange
Rennes 35000
France

Email: mohamed.boucadair@orange.com

Jon Shallow
United Kingdom

Email: supjps-ietf@jpshallow.com