Speeding Up CoAP Block-wise Transfer
draft-zcao-core-speedy-blocktran-00

Abstract

This document specifies a method to speed up block-wise transfer in CoAP. With this, the client can indicate its willingness to be responded with a sequence of blocks without issuing request for each block one by one.

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 May 11, 2019.

Copyright Notice

Copyright (c) 2018 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 (https://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
When performing the block-wise CoAP transfer as defined in [RFC7959], the Client needs to continuously send requests to the Server, using the BLOCK options to specify the exact segment that is expected. As a consequence, the Server can only acknowledge different blocks one by one according to the request. Such a design was a reasonable choice since the server can be implemented to be truly stateless and lightweight. However, there are some cases where the server is capable of keeping necessary states so that the transfer can be performed in a more efficiently and faster way, e.g., the server being a firmware upgrade device without constraints defined for CoAP. This document describes such a proposal that is used to speed up the CoAP block transfer.

2. Conventions used in this document

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].

3. Speedy Blockwise Transfer

3.1. The Speedy Block Option

We introduce a new option called 'BlockS', the format being specified in Figure 1. The client can include this option in the payload of the request to indicate its willingness to fetch the block content in a speedy way.
The value of the BlockS option is a variable-size unsigned integer. This integer value encodes these four fields: NUM, M, SIZE, and SPDYWND; NUM, M and SIZE are defined the same as [RFC7959].

NUM: the relative number of the block (NUM) within a sequence of blocks with the given size. (4, 12, 20 bit)

M: whether more blocks are following (M) (1-bit);

SIZE: the size of the block (SIZX) (3-bit), the actually size in octct equals to \((2^{(SZX+4)})\);

SPDYWND(8-bit): the allowance of maximum window size used for speed-up, for a total of SPDYWND messages, the Server will solicitate an acknowledgement from the client

if the client includes the BlockS option in its request, it indicates it will be willing and capable to receive a bunch of blocks without sending the request one by one. Since the client is fully aware of its capacity, it MUST indicate the SPDYWND value in the BlockS option explicitly.

Upon receiving a request with the BlockS option, the Server will check the SPDYWND value in the request. It will calculate how many blocks (according to the SIZE value) the requested content consists of, and if this value being smaller than the SPDYWND, it will send the first block by changing the SPDYWND to the actually size (in block).

The first block is piggyback in the Acknowledgement of the client request. for the next (SPDYWND-1) messages, the server will send the content in a NON message. for every SPDYWND responding messages, the server will send a CON message to solicitate an acknowledgement from the client to make sure that the client is still alive.

if the Server does not support the BlockS option, it will neglect this message. The client, under this case, needs to implement a timeout mechanism, so that it can switch to other block-wise transfer (Block1 or Block2) as specified in [RFC7959].
3.2. Example Follow-chart

In the following example, the client specifies its SPDYWND as 5, but the server finds out the representation only consists of 4 blocks, it will change the SPDYWND value to 4 in the BlockS option. The client will recognize this as an indication of actually length of the resource, continue to receive the content response until the M(More) bit being unset.

CLIENT | SERVER
--- | ---
CON [MID=1234], GET, T=0xFA /status, S:0/0/64/5 | ----->
<----ACK [MID=1234],T=0xFA, 2.05 Content, S:0/1/64/4
<----NON [MID=2000],T=0xFA, 2.05 Content, S:1/1/64/4
<----NON [MID=2001],T=0xFA, 2.05 Content, S:2/1/64/4
<----CON [MID=2002],T=0xFA, 2.05 Content, S:3/0/64/4
-------------- ACK [MID=2002] -------------->

Figure 2: Speedy Blockwise GET with Early Negotiation

In the following example, the client specifies its SPDYWND as 5, and the server finds out the representation consists of 8 blocks, it will keep the SPDYWND value to 5 in the BlockS option. The server will send a CON message every SPDYWND messages.
3.3. Retransmission Considerations

There is only one CON message every SPDYWND messages sending by the server. If the NON messages get lost during the communication, the server has no way to know the fact and will not retransmit it. Only after receiving a number of messages from the server, the Client can identify the "hole" due to packet loss. The client can send separate request per the missing sequence number in the whole block.

Figure 4 shows an example where the NUM 2 and NUM 6 blocks get lost. The client will send separate requests for block number 2 and 6 after receiving the final block where M=0.
4. Security Considerations

TBD

5. IANA Considerations

This document needs an assignment of the BlockS option value defined in Section.3.1.
6. Acknowledgments

TBD

7. References


Authors’ Addresses

Zhen Cao
Huawei
Xinxi Rd 3
Beijing 100085
China

Email: zhencao.ietf@gmail.com

Ke Jin
Huawei
Shenzhen
China

Email: jinke@huawei.com

Baicheng Fu
Huawei
Shenzhen
China

Email: 290275570@qq.com
Dacheng Zhang
Huawei
Beijing
China

Email: zhangdacheng@huawei.com