core Internet-Draft Intended status: Informational Expires: January 10, 2013

CoAP Minimum Block Time draft-greevenbosch-core-block-minimum-time-00

Abstract

This document defines an "MinimumBlockTime" option for CoAP, which can be used to negotiate the minimum time between two subsequent block requests. It can be used for overload and congestion control.

Greevenbosch

Expires January 10, 2013

[Page 1]

Note

Discussion and suggestions for improvement are requested, and should be sent to core@ietf.org.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of <u>BCP 78</u> and <u>BCP 79</u>.

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 <u>http://datatracker.ietf.org/drafts/current/</u>.

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 10, 2013.

Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to <u>BCP 78</u> and the IETF Trust's Legal Provisions Relating to IETF Documents (<u>http://trustee.ietf.org/license-info</u>) 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

<u>1</u> .	Introduction														4
<u>2</u> .	Requirements notati	on													<u>5</u>
<u>3</u> .	Definitions														<u>6</u>
<u>4</u> .	The "MinimumBlockTi	me"	op	oti	lor	ı									7
<u>5</u> .	Legacy behaviour .														<u>8</u>
<u>6</u> .	Open issues														<u>9</u>
<u>7</u> .	Example														<u>10</u>
<u>8</u> .	Security Considerat	ions	5												<u>11</u>
<u>9</u> .	IANA Considerations														<u>12</u>
<u>10</u> .	Acknowledgements .														<u>13</u>
<u>11</u> .	Normative Reference	s.													<u>14</u>
Auth	nor's Address														<u>15</u>

1. Introduction

The Constrained Application Protocol (CoAP) [<u>I-D.ietf-core-coap</u>] is a RESTful protocol for constrained nodes and networks. In [<u>I-D.ietf-core-block</u>], the block mechanism for block-wise transmission of data is defined.

This document defines a "MinimumBlockTime" option, which can be used to negotiate the minimum time between two subsequent block requests.

Negotiating the minimum time between the block requests can be used to limit the associated traffic, providing a mechanism for congestion control. In addition, it allows very constrained servers to limit the number of requests they receive within a certain time period, preventing them from becoming overloaded.

2. Requirements notation

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 [<u>RFC2119</u>].

3. Definitions

Block transaction

The block-wise transmission of a single resource, between a unique client and server pair.

Two subsequent requests

In this document, the phrase "two subsequent requests" indicates two requests in a same block transaction, in which one follows the other, without other requests from the same block transaction in between. Notice that the difference between the block numbers of the two subsequent requests does not have to be one, although most often this will be the case.

Block time

The time between two subsequent requests in a block transaction.

Block speed

The multiplicative inverse of the block time.

4. The "MinimumBlockTime" option

Table 1: The "MinimumBlockTime" option

The "MinimumBlockTime" option is an elective option, which is used to negotiate the minimum time in milliseconds that a client needs to wait between sending two subsequent block requests.

In the remainder of this section, it is assumed that both the client and the server support the "MinimumBlockTime" option.

If the client includes a "Block1" or "Block2" option in its first request in a block transaction, it SHOULD include the "MinimumBlockTime" option in that first request too. The server SHOULD include the "MinimumBlockTime" option in its first block response.

In a block request, the option's value T_C indicates the minimum block time in ms that the client can support.

In a block response, the option's value T_S indicates the minimum block time in ms that the server can support.

The client SHALL wait at least T_S ms between sending two subsequent block requests.

The following MUST hold: T_S <= T_C.

The "MinimumBlockTime" option has a default value 0. A value $T_S=0$ indicates the server does not put any restrictions on the block speed. A value $T_C=0$ indicates that the client prefers to send the requests as quickly as possible.

5. Legacy behaviour

It is possible that either the client or server does not support the "MinimumBlockTime" option. If the client does not support the option, then obviously it cannot take the server's preference into account. Similarly if the server does not support the option, it cannot use it to restrict the block speed.

In either case, or their combination, the client will choose the block speed as it prefers. This corresponds to the case $T_S=0$.

To allow the server to distinguish between a client that supports the "MinimumBlockTime" option but wants to signal T_C=0, and a client that does not support the "MinimumBlockTime" option, it is RECOMMENDED for the compliant client to include option in the first request in a Block transaction, even when the client wants to signal $T_C=0$.

<u>6</u>. Open issues

For longer block transactions, there may be value in allowing updates of the block speed during a block transaction. We should consider whether increasing efficiency will justify the extra complexity.

7. Example

Figure 1 contains an example of a block transaction with the "MinimumBlockTime" option. The client indicates its supported minimum block time as 200ms. The associated block speed is too high for the server, so the server indicates a minimum block time of 300ms. The client obeys this value for the rest of the transaction.

CLIE	NT	SERVER
/	CON [MID=1234], GET, /status, N=0, T_C = 200	 >
300ms 	< ACK [MID=1234], 2.05 Content, N=0, T_S	= 300
	CON [MID=1235], GET, /status, N=1	>
300ms	< ACK [MID=1235], 2.05 Content, N=1	
	CON [MID=1234], GET, /status, N=2	>
300ms	< ACK [MID=1234], 2.05 Content, N=2	i
\	CON [MID=1235], GET, /status, N=3	>
:		:

Figure 1: Example of transaction with "MinimumBlockTime"

Greevenbosch Expires January 10, 2013 [Page 10]

8. Security Considerations

By modifying the value of the "MinimumBlockTime" option to a higher value, a man-in-the-middle could increase the time used to perform a block transaction. When the client encounters a response with a too high "MinimumBlockTime" value, it MAY abort the transaction, and try to reinitiate it. However, to prevent overloading the server, the client MUST limit the number of these reinitiations.

By decreasing the value of the "MinimumBlockTime" option, the man-inthe-middle can induce the client to send block requests at a speed too high for the server. The server should be prepared for this, for example by discarding requests that cannot be processed. This is similar to the case where the server or client does not support the "MinimumBlockTime" option.

Greevenbosch Expires January 10, 2013 [Page 11]

9. IANA Considerations

This draft adds the following option numbers to the CoAP Option Numbers registry of [<u>I-D.ietf-core-coap</u>].

> +----+ Number | Name | Reference | +----+ | TBD (elective) | MinimumBlockTime | [RFCXXXX] | +----+

> > Table 2: CoAP option numbers

<u>10</u>. Acknowledgements

The author would like to thank Kepeng Li for his ideas and feedback.

11. Normative References

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

[I-D.ietf-core-block]

Shelby, Z. and C. Bormann, "Constrained Application Protocol (CoAP)", draft-ietf-core-block-08 (work in progress), February 2012.

[I-D.ietf-core-coap]

Shelby, Z., Hartke, K., Bormann, C., and B. Frank, "Constrained Application Protocol (CoAP)", draft-ietf-core-coap-10 (work in progress), March 2012.

Greevenbosch Expires January 10, 2013 [Page 14]

Author's Address

Bert Greevenbosch Huawei Technologies Co., Ltd. Huawei Industrial Base Bantian, Longgang District Shenzhen 518129 P.R. China

Phone: +86-755-28978088 Email: bert.greevenbosch@huawei.com