Abstract

CoAP is a RESTful application protocol for constrained nodes and networks. Through the Observe option, clients can observe changes in the state of resources and obtain a current representation of the last resource state. This document defines two new options for CoAP Observe so that a CoAP client can specify timing conditions when observing a resource on a CoAP server. As a result, the CoAP client is only informed about resource state changes when the timing conditions are met. This offers possibilities to extend network intelligence, enhance scalability, and optimize the lifetime and performance in order to address the requirements from the Constrained Nodes and Networks.

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 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."
1. Introduction

CoAP [RFC7252] is an Application Protocol for Constrained Nodes/Networks. The observe [I-D.ietf-core-observe] specification describes a protocol design so that a CoAP client and server can use the subject/observer design pattern to establish an observation relationship. When observe is used, the CoAP client will get a notification response whenever the state of the observed resource changed. However, in some scenarios, the CoAP client may only be interested in the state changes of the resource after a specified time interval, to avoid superfluous traffic. This memo defines two new CoAP options "Minimum-Interval" and "Maximum-Interval" that can be used to allow the CoAP client to set conditions for the
observation relationship, and only when such condition is met, the CoAP server will send the notification response with the latest state change. When such a condition fails, the CoAP server does not need to send the notification response.

1.1. Justification

A GET request that includes an Observe Option creates an observation relationship. When a server receives such a request, it first serves the request like a GET request without this option and, if the resulting response indicates success, establishes an observation relationship between the client and the target resource. The client is notified of resource state changes by additional responses sent in reply to the GET request to the client.

CoAP is used for Constrained Networks, especially used for transporting sensor data. Different sensor equipments have different properties, e.g. different change rates, different response time, etc. resulting in varying clients’ interests that differ from mere state changes. As such, when a client wants to collect information from a sensor, it does not want to receive too many notification messages within a short time period, if the state of a server resource changes too often. Also, if the resource’s representation does not change for a long time, the client wants to receive notifications in order to make sure that the observe relationship is still alive.

Consider the following example.

<table>
<thead>
<tr>
<th>CLIENT</th>
<th>SERVER</th>
</tr>
</thead>
</table>
| ------ | GET:/temperature, Observe:0 ------>
| <------ | 2.05 Content, Observe:5 Payload:22 --------|
| <------ | 2.05 Content, Observe:10 Payload:22.3 --------|
| <------ | 2.05 Content, Observe:15 Payload:22.6 --------|

Figure 1: GET request with observe

In this example, the sensor acts as a server, and it collects the resource data every 5 seconds. When the client observes a resource on the server, it will receive a response whenever the server updates the resource, that is to say, mostly every 5 seconds the client will receive a notification response. However, the client might be a
simple constrained device not too sensitive to the resource state change, so it may not want to receive notifications that often. One possible solution could be to alter the sensor’s configuration, e.g. to shorten the collecting frequency. However, the sensor should be able to provide services to many other clients, making it hard to find the best configuration that fits all clients’ requirements.

1.2. Terminology

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].

2. Motivation

The CoAP Observe Option gives clients the ability to observe changes in the state of resources. An observe relationship is established and whenever the state of the resource changes, the new representation is pushed to the observer. In some cases, the server resource changes too often, while the client does not want to receive notifications that often. The client just wants to receive notifications after a specific time interval.

Defining a standardized set of commonly used conditional observations has a number of advantages. In a well-defined way, clients can observe different resources conditionally. At the same time, these resources can clearly announce how they can be observed, facilitating machine processing of this information. Also, intermediaries can process multiple conditional observations, with the goal to alleviate the load on the constrained network and devices. In the absence of such a set of commonly used conditional observations, where every application developer can specify its own mechanisms, these advantages are lost.

In [I-D.ietf-core-interfaces], a mechanism is described to provide additional information to the Observe Option through the use of query parameters. It is possible to define a fixed set of query parameters to enable conditional observations. However, many more query parameters can be offered by a resource for different purposes. This complicates the automatic processing of conditional observations. Also embedding the query parameters in the URI encumbers processing at intermediaries. To alleviate this problem, this draft proposes to use CoAP options to specify timing-related conditions, that is, minimum time interval and maximum time interval, for the notifications. Using options ensures a compact representation and well-defined meaning. For resource value related conditions, e.g. larger than, smaller than, another mechanism such as query parameters
can be used to complement the Minimum-Interval and Maximum-Interval Options.

3. Conditional Observe Options

<table>
<thead>
<tr>
<th>No.</th>
<th>C</th>
<th>U</th>
<th>N</th>
<th>R</th>
<th>Name</th>
<th>Format</th>
<th>Length</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>TBD</td>
<td></td>
<td>x</td>
<td>-</td>
<td></td>
<td>Minimum-Interval</td>
<td>int</td>
<td>0-2 B</td>
<td>(none)</td>
</tr>
<tr>
<td>TBD</td>
<td></td>
<td>x</td>
<td>-</td>
<td></td>
<td>Maximum-Interval</td>
<td>int</td>
<td>0-2 B</td>
<td>(none)</td>
</tr>
</tbody>
</table>

Figure 2: Conditional Observe Options

This draft defines two condition options for observe, Minimum-Interval and Maximum-Interval. Both options are elective, and Proxy Unsafe (similar to the Observe Option).

The Minimum-Interval and Maximum-Interval options can only be present in a GET request message and its response message. They must be used together with the Observe Option, since they extend the meaning of the Observe Option. The Minimum-Interval/Maximum-Interval MUST be included in the first notification message if the conditional observation relationship is created successfully. If the server does not support the Minimum-Interval/Maximum-Interval option contained in the observe request, it will ignore the them.

The Minimum-Interval Option indicates the minimum time interval between two notification responses sent from the server, even if the resource representation changes. After sending the previous notification response, the server has to wait for the minimum time interval to expire, before sending the subsequent notification response, if the resource representation changes within the specified minimum time interval.

The Maximum-Interval Option indicates that the maximum time interval between two notification responses sent from the server, even if the resource representation does not change. After sending the previous notification response, if the resource representation does not change after the maximum time interval, the server needs to send the same resource representation immediately after timeout of the maximum time interval. This can be used to show the liveness of the server.

Basically, a similar procedure as described in the observe draft [I-D.ietf-core-observe] is followed, but extended with additional
behavior by taking the Minimum-Interval and Maximum-Interval Options into account. The exact semantics are described below.

The values of the Minimum-Interval and Maximum-Interval Options are measured in seconds. The range is from 1 second to $2^{16}$ seconds, that is, 65256 seconds, around 18 hours. There is no default value for the Minimum-Interval/Maximum-Interval option.

The Minimum-Interval Option and Maximum-Interval Option are elective, and may be present separately or together. If both of the options are included in a request, their relationship is "AND", meaning that the server will only send a notification when both conditions are fulfilled. Note that if both options are present, Maximum-Interval option should be greater than or equal to Minimum-Interval option. If the Minimum-Interval Option and Maximum-Interval Option are both present and have the same value, then the server should send notifications periodically, regardless whether the resource representation has changed or not.

4. Using the Condition Option

Whenever a client wants to initiate a Conditional Observation relationship, it sends a GET request with both an Observe option and at least one Conditional Observe Option. The Conditional Observe Options represent the minimum and maximum time interval conditions the client wants to apply to the observation relationship.

When a server receives such a request, it first services the request in the same way as described in [I-D.ietf-core-observe]. Next, if the server supports the Minimum-Interval or Maximum-Interval option, it analyzes the options to find the condition requested by the client. If the Minimum-Interval/Maximum-Interval option is supported, the relationship is stored and the client is informed about the successful establishment of the conditional relationship. This is done by sending a response containing both the Observe and Minimum-Interval/Maximum-Interval option, which implies that the client has now been added to the list of observers and will only be informed about state changes or resource states satisfying the conditions described in the Minimum-Interval/ Maximum-Interval option.

Since the Minimum-Interval/Maximum-Interval option is elective, an observe request that includes the option will automatically fall back to a basic observe request if the server does not support the Minimum-Interval/Maximum-Interval option. This implies that the client will now receive notifications as described in [I-D.ietf-core-observe] and that the client itself is responsible for
processing the resource state in the notifications in order to identify whether the condition of interest is fulfilled or not.

Whenever the state of a resource that supports conditional observations changes on the server, the server needs to check the established conditional relationships. Whenever the relationship condition(s) is(are) met, the server sends the notification response to the client that has established the relationship. In case the server is still waiting for a confirmable notification to be acknowledged or the 3 seconds on average for a non-confirmable notification to elapse, it MUST adhere to the behaviour specified in section 4.5 of [I-D.ietf-core-observe].

5. Examples

This section gives some short examples with message flows to illustrate the use of the Minimum-Interval and Maximum-Interval Options in an observe GET request.

The following example shows that the client sets the Minimum-Interval Option to 10 seconds. This means that the server shall wait at least 10s between sending notification responses, indicating changes in the state of the resource, to the client.
The next example shows that the client sets the Maximum-Interval Option to 60 seconds. The server will send notifications upon every state change, but will leave maximally 60s between subsequent notifications, even if they do not incur a state change.
The next example shows a client that sets both the Minimum-Interval Option and the Maximum-Interval Option to 30 seconds. As a result, the server sends notifications every 30 seconds, independent of whether the resource has changed or not.
Figure 5: Minimum-Interval and Maximum-Interval Options together

Further, in [CPSCOM], an evaluation can be found regarding the feasibility of implementing conditional observations on real constrained devices, together with a basic performance comparison between conditional observe (server-filtering) and normal observe in combination with client-side filtering.

6. Security Considerations

As the Minimum-Interval and Maximum-Interval options are used together with the Observe option, when it is used it must follow the security considerations as described in Observe draft [I-D.ietf-core-observe].

7. IANA Considerations

This draft adds the following option numbers to the CoAP Option Numbers registry of [RFC7252]
Table 3: Conditional Observe Option Numbers

<table>
<thead>
<tr>
<th>Number</th>
<th>Name</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>TBD</td>
<td>Minimal-Interval</td>
<td>[RFCXXXX]</td>
</tr>
<tr>
<td>TBD</td>
<td>Maximum-Interval</td>
<td>[RFCXXXX]</td>
</tr>
</tbody>
</table>

8. Acknowledgements

Thanks to the IoT6 European Project (STREP) of the 7th Framework Program (Grant 288445).

Thanks to Zach Shelby, Bert Greevenbosch for the discussions.

9. References

9.1. Normative References

[I-D.ietf-core-observe]


9.2. Informative References


[I-D.bormann-coap-misc]
Bormann, C. and K. Hartke, "Miscellaneous additions to CoAP", draft-bormann-coap-misc-13 (work in progress), March 2012.

[I-D.ietf-core-interfaces]

Appendix A. Change log

Changes in v05

- Simplified to contain only two options:
  - Minimal Interval
  - Maximum Interval

Changes in v04

- Updated draft to be consistent with observe draft:
  - (Took request URI into consideration for Cancellation.)
  - (Explicitly stated that the draft diverges from a MUST NOT defined in the observe draft)
  - Stated that a server should follow the text from the observe draft for an unacknowledged notification in regards to the transmission of new notifications and the cancellation of existing relationships.
  - Removed obsolete Pledge and Keep-Alive options due to decoupling the freshness of a representation from whether or not the client is still on the list of observers in observe v08. Instead, rely on server-side initiated CONfirmable messages (as defined in the Observe) for checking the existence of a relationship.
  - (Switched Reliability flag with Logic flag.)

- Updated source endpoint terminology.

Changes in v03

- Examples for most condition types

- Update the option number according to the new numbering scheme
o Added reference to paper validating implementation on constrained device

Changes in v02

o Restructured entire document

o Detailed description of the Condition Option and updated format of the Condition Option value

o Added more Condition Types

o New section on cancellation, updating and existence of conditional relationships

o New section on discovery

Authors’ Addresses

Shitao Li
Huawei Technologies
Huawei Base
101 Software Avenue, Yuhua District
Nanjing, Jiangsu  210012
China

Phone: +86-25-56624157
Email: lishitao@huawei.com

Kepeng Li
Huawei Technologies
Huawei Base
Bantian, Longgang
Shenzhen, Guangdong  518129
China

Phone: +86-755-28970231
Email: likepeng@huawei.com