core
Internet-Draft
Intended status: Standards Track
Expires: September 13, 2012

Shi Tao. Li
Huawei Technologies
J. Hoebeke
IBBT-IBCN/UGent

March 12, 2012

Conditional observe in CoAP
draft-li-core-conditional-observe-01

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. This document defines a new mechanism in CoAP Observe so that a CoAP client can conditionally observe a resource on a CoAP server, only being informed about state changes meeting a specific condition.

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."

This Internet-Draft will expire on September 13, 2012.

Copyright Notice

Copyright (c) 2012 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 (http://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 the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction

CoAP [I-D.ietf-core-coap] 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 can get a notification response whenever the state of the observed resource changed. However, in some scenarios, the CoAP client may only care parts of the state change of the resource, other state changes might be meaningless. This inability to suppress additional notification results in superfluous traffic. This memo defines a new CoAP option "Condition" that can be used to allow the CoAP client to condition the observe request, and only when such condition is met, the CoAP server shall 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 services 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. But different sensor equipments have different properties, e.g. different data unit, different response time. When a client wants to collect information from a sensor, it does not want to receive useless information, it hopes that the sensor only responses with what the client wants.

Consider the following example.

```
      CLIENT                                                      SERVER
        |                                                            |
        | 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 quite simple equipment not too sensitive to the resource state change, so it may not want to receive the notification that often. One possible solution could be to change the sensor's parameter, 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. A notification relationship is established and whenever the state of the resource changes, the new representation is pushed to the observer. In many cases, an observer will typically be interested in state changes that satisfy a specific condition. In addition, similar conditional observations will prove useful for many different resources. For example, being informed when the state of a resource exceeds a specific value.

Defining an agreed 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, intermediate machines such as a smart proxy, can process multiple conditional observations, having as 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.shelby-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. To alleviate this problem, this draft proposes to implement frequently occurring conditional observations through the use of a new CoAP Condtion Option, having a compact representation and well-defined meaning. For other specific conditional observations, another mechanism such as query parameters can be used to complement the Condition Option.

3. Condition types

This section lists some possible conditions that can be used to extend the observe request.

Minimum/Maximum Period: the minimum/maximum time in seconds between notifications

Step: how much the value of a resource should change before sending a new notification

Periodic: periodic interval with which new notification should be sent

Range: a notification is sent when the value of the resource lies under a specific range.

4. Overview of Operation

Whenever a client wants to initiate a Conditional Observation relationship, it sends a GET request with a Observe and Condition Option (see section 5 for the Condition Option definition). The Condition Option includes the condition required by the client such as the minimum response time or the minimum step change (see section 6 for the definition of condition type). When a server receives such a request, it first services the request the same way as described in [I-D.ietf-core-observe]. Next, if the server supports the Condition Option, it needs to analyze the Condition Option to find the condition requested by the client. If the condition is supported, the client is informed about the successful establishment of the conditional relationship.

Whenever the resource state changes on the server, it needs to check the established conditional relationships. Whenever the condition is met, the server shall send the notification response to the client that has established the relationship. If not met, the server does not need to send any response to the client.

5. Definition of Condition Option

Type	C/E	Name	Data type	Length	Default
22	E	Condition	uint	1-3 B	

table 1: Condition Option number

6. Using the Condition option

The Condition Option is used to indicate the condition requirement requested by a CoAP client. It must be used together with the Observe Option and represents the condition the client wants to apply to the observation relationship. The server needs to follow the general procedure as described in observe draft [I-D.ietf-core-observe], but takes the Condition Option into account.

Since the condition Option is elective, an observe request that includes the Condition Option will automatically fall back to a basic observe request if the server does not support the Condition Option. There is no default value for the Condition Option.

The Condition option may occur more than once, when multiple Condition Options are present in an observe GET request, it means that the initiator has multiple condition requirements, and the first condition option in the request has the highest priority.

If multiple Condition Options with the same condition type are present in a request, their priority is the same, and the relationship is "AND".

The size of the Condition Option is not fixed and can range from 1 to 3 bytes. The value carried is in a specific format consisting of three fields: condition type, method and condition value.

```
           0
           0 1 2 3 4 5 6 7 8 9
          +-+-+-+-+-+-+-+-+-+-+
          | TYPE  | M | VAL   |
          +-+-+-+-+-+-+-+-+-+-+

           0                   1
           0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7
          +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
          | TYPE  | M |         VAL           |
          +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

           0                   1                   2
           0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
          +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
          | TYPE  | M |         VAL                           |
          +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

table 2: Condition Option value

TYPE: condition type. The condition type is a 4 bit integer indicating the type of the condition used in the observe request. Every value of TYPE represents the type ID of a specific condition type. For example, "1" represents the minium response time, "2" represent the step. Below is the definition of identified condition types.

M: method. The method is a 2 bit integer indicating the method used in the condition. Below is the further definition of the method attribute.

VAL: condition value. The condition value is a variable-size (4,12, or 20 bit) unsigned integer indicating the value of the condition. The value can range from 0 to 2^4 (16), from 0 to 2^12 (4096) or from 0 to 2^20 (1048576).

Condition type	Id.
minimum response time	1
maximum response time	2
step	3
range	4
periodic	5

table 3: Condition type

Minimum response time: When present, it indicates that the condition required by the initiator is the minimum response time. This condition indicates the minimum time the server should wait between sending notification responses.

Maximum response time: When present, it indicates that the condition required by the initiator is the maximum response time. This condition indicates the maximum time the server should wait between sending notification responses.

Step: When present, it indicates that the condition required by the initiator is the change step. This condition indicates the minium state change of a resource before the server can send a new notification response.

Range: When present, it indicates that the condition required by the initiator is the state change range. This condition indicates that only when the state value is under the range of this condition, the server shall send a new notification response. The value and the method attributes decide the range of the condition, e.g, set the method to ">", and value to "20", means that the range is bigger than 20, so only the state value bigger than 20, that the server shall send a new notification response to the client.

Periodic: When present, it indicates that the condition required by the initiator is the periodic response time . This condition indicates the periodic interval with which new notifications should be sent.

Method	Id.
=	0
>	1
<	2

table 4: Condition method

7. Examples

This section gives a number of short examples with message flows to illustrate the use of Condition Option in a observe GET request.

The first example (Figure 2) shows that the client set the Conditon Option to 1/0/10, it means that the server shill wait at least 10s between sending notification responses to the client .

```
   CLIENT                                                      SERVER
     |                                                           |
     |     GET/temperature, observe:0,Condition:1/0/10----->     |0s
     |                                                           |
     |                                                           |
     | <------ 2.05Content,observe:5,payload:22                22|5s
     |                                                           |
     |                                                       22.4|10s
     |                                                           |
     | <------ 2.05Content,observe:15,payload:22.5             22.5|15s
     |                                                           |
     |                                                       23  |20s
     |                                                           |
     |  <------ 2.05Content,observe:25,payload:22.8            22.8|25s
     |                                                           |
```

Figure 2: Condition Option with value 1/0/10

This example (Figure 3) shows the client set the Conditon Option value to 3/0/1, it means that the server will send the notification response to the client if the resource state change bigger than 1.

```
      CLIENT                                                     SERVER
        |                                                          |
        | GET/temperature, observe:0,Condition:3/0/1       ----->  |
        |                                                          |
        |                                                          |
        | <------ 2.05Content,observe:5,payload:22                 |22
        |                                                          |
        |                                                          |22.5
        |                                                          |
        | <------ 2.05Content,observe:20,payload:23.2              |23.2
        |                                                          |
        |                                                          |23.6
        |                                                          |
        |  <------ 2.05Content,observe:35,payload:24.3             |24.3
```

Figure 3: Condition Option with value 3/0/1

This example (Figure 4) shows the client set the Conditon Option value to 4/1/5, it means that the server will send the notification response to the client only if the resource value bigger than 5.

```
      CLIENT                                                     SERVER
        |                                                          |
        | GET/temperature, observe:0,Condition:4/1/5        -----> |
        |                                                          |
        |                                                          |
        | <------ 2.05Content,observe:5,payload:4                  |4
        |                                                          |
        |                                                          |3
        |                                                          |
        | <------ 2.05Content,observe:25,payload:6                 |6
        |                                                          |
        |                                                          |4.5
        |                                                          |
        |  <------ 2.05Content,observe:25,payload:7                |7
```

Figure 4: Condition Option with value 4/1/5

This example (Figure 5) shows the client adds two range Conditon Options in the request , one sets to 4/1/5, another one sets to 4/2/15 it means that the range is within 5 and 15.

```
      CLIENT                                                     SERVER
        |                                                           |
        | GET/temperature, observe:0,Condition:4/1/5,               |
                                   Condition:4/2/15       ----->|
        |                                                           |
        |                                                           |
        | <------ 2.05Content,observe:5,payload:4                   |4
        |                                                           |
        |                                                           |3
        |                                                           |
        | <------ 2.05Content,observe:25,payload:12                 |12
        |                                                           |
        |                                                           |16
        |                                                           |
        |  <------ 2.05Content,observe:25,payload:14                |14
```

Figure 5: Two Condition Option with the same condition type

8. Security Considerations

As the Condition Option is 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].

9. IANA Considerations

9.1. Condition option registry

This draft adds the following option number to the CoAP Option Numbers registry of [I-D.ietf-core-coap]

Number	Name	Reference
22	Condition	[RFCXXXX]

table 5: Condition Option number

9.2. Condition type registry

The Condition types defined in this draft are identifed by a string, such as "step". In order to minimze the overhead of using these condition type, this document defines a registry for the condition types to be used in CoAP and assigns each a numeric identifier.

Each entry in the registry must include the condition type registered with IANA, the numeric identifier in the range 0-15 to be used for that condition type in CoAP, and a reference to a document defining the usage of that condition type.

Initial entries in this registry are as follows:

Condition type	Id.	Reference
minimum response time	1	[RFCXXXX]
maximum response time	2	[RFCXXXX]
step	3	[RFCXXXX]
range	4	[RFCXXXX]
periodic	5	[RFCXXXX]

table 6: Condition Option type

9.3. Condition method registry

The condition method used in this draft are as follow:

Method	Id.	Reference
=	0	[RFCXXXX]
>	1	[RFCXXXX]
<	2	[RFCXXXX]

table 7: Condition method

10. Further Considerations

10.1. Resource Discovery

The Condition Option enables the establishment of well-defined set of conditional observations. It is equally important for a resource to be able to announce in a well-defined way which conditional observations it supports. Clients can then discover these capabilities and process them automatically.

By providing a value to the "obs" attribute defined in [I-D.ietf-core-observe], the conditional capabilities of a resource can be described and discovered. In order to describe which of the 2^4 possible condition types a resource supports, a 16-bit value can be used where a bit-value of 1 at position X (from right to left) indicates that the condition type X is supported.

11. Acknowledgements

12. Normative References

[I-D.ietf-core-coap]
Frank, B., Bormann, C., Hartke, K., and Z. Shelby, "Constrained Application Protocol (CoAP)", draft-ietf-core-coap-08 (work in progress), October 2011.

[I-D.ietf-core-link-format]
Shelby, Z., "CoRE Link Format", draft-ietf-core-link-format-11 (work in progress), January 2012.

[I-D.ietf-core-observe]
Hartke, K., "Observing Resources in CoAP", draft-ietf-core-observe-04 (work in progress), February 2012.

[I-D.shelby-core-interfaces]
Shelby, Z., "CoRE Interfaces", draft-shelby-core-interfaces-01 (work in progress), January 2012.

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

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

Jeroen Hoebeke
IBBT-IBCN/UGent
Department of Information Technology
Internet Based Communication Networks and Services (IBCN)
Ghent University - IBBT
Gaston Crommenlaan 8 bus 201
Ghent B-9050
Belgium

Phone: +32-9-3314954
Email: jeroen.hoebeke@intec.ugent.be