CoRE Working Group
Internet-Draft
Intended status: Standards Track
Expires: August 25, 2017

C. Groves
W. Yang
Huawei

February 21, 2017

Additional CoAP Binding and Observe Attributes
draft-groves-core-obsattr-00

Abstract

[I-D.ietf-core-dynlink] defines five CoAP Observaton attributes (minimum period, maximum period, band step, less than and greater than) to control when notifications are sent. These attributes are insufficient for some use cases. This document specifies additional attributes allowing for notification bands, initialization values, band step, sample number window and sample time window to allow for a wider range of use cases.

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 August 25, 2017.

Copyright Notice

Copyright (c) 2017 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. Requirements Language 2
2. Introduction 2
3. Terminology 4
4. Binding and Resource Observation Attributes 4
4.1. Initialization Value (iv) 5
4.2. Notification Band Minimum (bmn) 6
4.3. Notification Band Maximum (bmx) 6
4.4. Band Step (bst) 6
4.5. Sample Number Window 7
4.6. Sample Time Window 7
4.7. Interactions 8
4.8. Examples 9
4.8.1. Example 1 - Band Minimum and Maximum 9
4.8.2. Example 2 - Band Minimum and Maximum and Step 10
4.8.3. Example 3 - Band Minimum and Maximum, Step and Initialization Value 10
4.8.4. Example 4 - Step and Initialization Value 11
4.8.5. Example 5 - Band Minimum and Maximum, Band Step and Initial Value 11
4.8.6. Example 6 - Band Minimum and Sample Number Window . . 12
5. Security Considerations 12
6. IANA Considerations 12
7. Acknowledgements 12
8. Changelog 12
9. References 12
9.1. Normative References 12
9.2. Informative References 13
Authors' Addresses 13

1. Requirements Language

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 [RFC2119].

2. Introduction

[I-D.ietf-core-dynlink] defines five CoAP Binding and Observaton attributes (minimum period, maximum period, change step, less than and greater than) to control when notifications are sent. The currently defined attributes have characteristics that means some use cases cannot be supported. These are described below:

- o A transition across a less than (lt), or greater than (gt) limit or a change step (st) only generates one notification. This means that it is not possible to describe a case where multiple notifications are sent so long as the limit is exceeded.

- o The change step (st) value is not deterministic when setting the attribute. A client cannot set the initial value that the change step applies to. The change step is based on the initial value sent by the server. This means that a client cannot indicate that it wants the change step (st=10) to apply to a certain increment (e.g. 10, 20, 30) instead it relies on the initial value from the client. Thus a change step of 10 could result in reporting (11, 21, 31) or equally (15, 27, 53).

- o SenML allows for multiple values (records) to be reported for a resource. The current attributes do not allow a method for a client to request a particular number of records or sample time window.

In order to allow a more complete set of use cases to be supported this specification introduces several new attributes.

The notification band attributes "Notification Band Minimum" (bmn) and "Notification Band Maximum" (bmx) attributes allow a bounded or unbounded (based on a minimum or maximum) value range that may trigger multiple state synchronizations. This enables use cases where different ranges results in differing behaviour. For example: monitoring the temperature of machinery. Whilst the temperature is in the normal operating range only periodic observations are needed. However as the temperature moves to more abnormal ranges more frequent synchronization/reporting may be needed.

An "Initialization Value" (iv) attribute allows a seed value for the calculation of the change step to be specified. This allows use cases where synchronization occurs around a known value. For example: synchronization will occur based on the operating temperature set point of a machine. Without the initialization synchronization will occur around the first measured value.

A "Band Step" (bst) attribute defines a series of bands that will trigger state synchronization. This allows use cases where state synchronization is required against known levels. Rather than synchronizing based on a difference to a previous synchronization value, synchronization occurs against a fixed known level. For example: it allows state sychronisation for a sensor when it's value is between (5,10],(10,15],(15,20] etc.

The "Sample Number Window" (snw) attribute allows a number of state synchronizations to be queued before the actual queue synchronization occurs. Once the number of queued state synchronizations has reached a certain level then a single queue synchronization occurs with the multiple resource values related to individual state synchronizations included. This allows use cases where multiple resource values are required but frequent synchronization is not required as there is a need to minimise resource usage. For example: a meter may need to be recorded once an hour but the values only need to be synchronized once a day.

The "Sample Time Window" (stw) attribute as per the sample number window allows state synchronizations to be queued before the actual queue synchronization occurs. The queue synchronization occurs when the indicated period expires independent of the number of samples.

3. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this specification are to be interpreted as described in [RFC2119].

This specification requires readers to be familiar with all the terms and concepts that are discussed in [RFC5988] and [RFC6690]. This specification makes use of the following additional terminology:

Notification Band: A resource value range that results in state sychronization. The value range may be bounded by a minimum and maximum value or may be unbounded having either a minimum or maximum value.

4. Binding and Resource Observation Attributes

The attributes defined in this section are additional attributes that may be used as binding attributes (3.3/[I-D.ietf-core-dynlink]) and resource observation attributes (4.2/[I-D.ietf-core-dynlink]).

This specification introduces several new attributes:

Attribute Name	Parameter	Data Format
Initialialization Value	/{resource}?iv	xsd:decimal
Band Minimum Notification	/{resource}?bmn	xsd:decimal
Band Maximum Notification	/{resource}?bmx	xsd:decimal
Band Step	/{resource}?bst	xsd:decimal (>0)
Sample Number Window	/{resource}?snw	xsd:integer (>0)
Sample Time Window	/{resource}?stw	xsd:integer (>0)

Table 1: Resource Observation Attribute Summary

The attributes may only be included at most once in a query.

4.1. Initialization Value (iv)

The attribute indicates the initialization value to be used to determine when a change step is notified. As such it MUST only be present in a query when the change step (st) or band step (bst) attribute is present (see [I-D.ietf-core-dynlink]). If st or bst is not present then the receiver MUST return a CoAP error code 4.00 "Bad Request" (or equivalent).

Without iv, on reception of a query the synchronization initiator uses the current value for the observed resource as the initial value to which the change step is applied. The use of iv overrides this behaviour and the iv value is used for the initial value (STinit or BSTinit). A state synchronization occurs once the resource value differs from the initial value by the change step value (i.e. CurrVal >= STinit + ST or CurrVal <= STint - ST). The initial value is then set to the state synchronization value.

A state synchronization due to Pmax (or Pmin) does not cause an update of the initial value. However once the initial value is updated by a state synchronization due to the other attributes in the query then the normal behaviour defined by 3.3.7/[I-D.ietf-core-dynlink] occurs.

4.2. Notification Band Minimum (bmn)

This attribute defines the lower bound for the notification band. State synchronization occurs when the resource value is equal to or above the notification band minimum. This attribute is optional. If not present there is no minimum value for the band. If present bmn must be less than bmx if it is also present otherwise the receiver MUST return a CoAP error code 4.00 "Bad Request" (or equivalent).

4.3. Notification Band Maximum (bmx)

This attribute defines the upper bound for the notification band. State synchronization occurs when the resource value is equal to or less than the notification band maximum. This attribute is optional. If not present there is no maximum value for the band. If present bmx must be more than bmn if it is also present, otherwise the receiver MUST return a CoAP error code 4.00 "Bad Request" (or equivalent).

4.4. Band Step (bst)

Like change step (st) this attribute indicates how much the value of a resource SHOULD change before triggering a state synchronization. The difference however is that the values used for the band step calculation are based on a constant step rather than being based on the synchronized value.

The current resource value or the initialization value (if provided) is used a seed to determine the band thresholds.

For example: Given a bst=10 and an initialization value=25. This defines a series of band step thresholds: i.e. ..., (5,15],(15,25],(25,35], ...

When the resource value enters a new band step by exceeding the minimum threshold value and being less than or equal to the maximum threshold value for a band step then synchronisation occurs.

A new synchronization occurs whenever the value enters a new band step. If the value jumps across band steps e.g. from 13 to 27 only one synchronisation occurs.

The band step MUST be greater than zero otherwise the server MUST return a CoAP error code 4.00 "Bad Request" (or equivalent).

Change step (st) and band step (bst) MUST NOT occur together in the same query.

4.5. Sample Number Window

If queuing of a number of state synchronizations are required then the sample number window attribute is set to the desired size of the window. The attribute may be set with valid combinations of other binding/resource observation attributes. When a state synchronization is triggered due to the other attributes the resource value is added to the list of samples instead of resulting in an update of the source and destination resource (state synchronization). Only when the number of samples in the window reaches the sample number window is a state sycnhronisation peformed for the resource. The samples are then flushed from the window and the process is repeated.

The use of the sample number window attribute may require the use of a suitable content-format (such as SenML [I-D.ietf-core-senml]) that allows multiple values/data points to be specified during the state synchronization.

Consideration should also be given to the resource capacity (i.e. memory) of the CoAP server for storing data associated with the sample window. The sample number window should not exceed its capabilities. Even if the sample number window has not been reached, if resource (memory) consumption is an issue then state synchronization for the stored resource values SHOULD occur enabling resources to be freed.

The pmin and pmax attributes have an indirect effect on the overall state sychronization. Whilst pmin and pmax do not directly specify the period for the overall state sychronization the setting of pmin and pmax may trigger samples entering the sample window as thus affect the frequency of state synchronization.

4.6. Sample Time Window

If state synchronizations are to be queued during a certain period of time (in seconds) then the sample time window attribute is used. The attribute may be set with valid combinations of other binding/ resource observation attributes. On reception of a query with the stw attribute a timer (T1=0) is started. Whilst T1<stw when a state synchronization is triggered due to the other attributes, the resource value is added to the sample window instead of resulting in a state synchronization. When the time expires e.g. T1=stw the state sychronization for the resource occurs. The window is then flushed, T1 is re-started and the process is repeated.

The use of the sample time window attribute may require the use of a suitable content-format (such as SenML [I-D.ietf-core-senml]) that

allows multiple values/data points to be specified during the synchronization.

Consideration should also be given to the resource capacity (i.e. memory) of the CoAP server for storing data associated with the time window. Consideration should be given to that the expected frequency of adding resource values and length of the time doesn't exceed the memory capacity of the server. Even if the sample time window has not been reached, if resource (memory) consumption is an issue then state synchronization for the stored resource values SHOULD occur enabling resources to be freed.

4.7. Interactions

To enable an notification band at least bmn or bmx MUST be set. If both bmn and bmx are set then a finite band is specified. State synchronization occurs whenever the resource value is between bmn and bmx or is equal to bmn or bmx. If only one attribute bmx or bmn is set then the band has an open bound. That is all values above bmn or all values below bmx will be synchronized.

When using multiple resource bindings (e.g. multiple Observations of resource) with different bands, consideration should be given to the resolution of the resource value when setting sequential bands. For example: Given BandA (Abmn=10, Bbmx=20) and BandB (Bbmn=21, Bbmx=30). If the resource value returns an integer then notifications for values between and inclusive of 10 and 30 will be triggered. Whereas if the resolution is to one decimal point (0.1) then notifications for values 20.1 to 20.9 will not be triggered.

Note: The use of bmn and bmx allow for a synchronization whenever a change in the resource value occurs. Theoretically this could occur in-line with the server internal sample period for the determining the resource value. Implementors SHOULD consider the resolution needed before updating the resource, e.g. updating the resource when a temperature sensor value changes by 0.001 degree versus 1 degree.

If pmin and pmax are present in a query then they take precedence over the other parameters. Thus even if bmn and bmx are met if pmin has not been exceeded then no state synchronization occurs. Likewise if bmn and bmx have not been met and pmax time has expired then state synchronization occurs. The current value of the resource is used for the synchronization. If pmin time is exceeded and bmn and bmx are met then the current value of the resource is synchronized. If st is also included, a state synchronization resulting from pmin or pmax updates STinit with the synchronized value. If bst is included, a state synchronization resulting from pmin or pmax updates bstinit with the closest bst delta value as per Section 4.4.

It is an error to include a greater than (gt) or less than (lt) attribute in a query containing bmn or bmx.

If change step (st) is included in a query with bmn or bmx then state synchronization will occur whilst the resource value is in the notification band AND the resource value differs from STinit by the change step.

If band step (bst) is included in a query with bmn or bmx then state synchronization will occur whilst the resource value is in the notification band defined by bmn or bmx AND has entered a new bandstep band.

If bst is included in a query with a gt or lt attribute then state synchronizations occur only when the conditions described by bst AND gt or bst AND lt are met.

If bst is included in a query with a iv attribute then iv is used to calculate the band thresholds. Subsequent state synchronizations are as per Section 4.4.

If iv is included with the bmn and bmx or gt and lt attributes it has no affect on the synchronisation. If bst is also used then it used to calculate band step thresholds. If st is instead used then it is used to calculate the initial value.

The snw and stw attributes SHALL not be used in the same query together. Synchronizations based on pmin and pmax are added to the snw/stw sample window. In effect this overrides the pmin and pmax mechanism because resource state synchronizations will not occur between the source and destination resources based on these parameters. For snw the minimum period will be snw * pmin. The maximum period will be snw * pmax. For stw the state synchronization will occur after time stw and is not dependent on pmin and pmax (unless a state synchronization occurs due to memory constraints). The stw must be more than pmax if it is present, otherwise the pmax attribute becomes invalid.

4.8. Examples

4.8.1. Example 1 - Band Minimum and Maximum

Req: POST /bnd/ (Content-Format: application/link-format)
<coap://sensor.example.com/s/temperature>; rel="boundto";anchor="/a/temperature";bind="obs";pmin="10";pmax="60";bmn="20",bmx="40"

Res: 2.04 Changed

The above will result in a state synchronization through an Observe:

- o Every 60 seconds if the value is not between 20 and 40.

- o When the temperature is equal to or between 20 and 40 at least every 10 seconds.

4.8.2. Example 2 - Band Minimum and Maximum and Step

```
Req: POST /bnd/ (Content-Format: application/link-format)
<coap://sensor.example.com/s/temperature>; rel="boundto";anchor="/a/
temperature";bind="obs";pmin="10";pmax="60";bmn="20",bmx="40",st="5"

Res: 2.04 Changed
```

The above will result in:

- o STinit being set to the temperature value at the time of the POST.

- o A state synchronization through an Observe:

  - * Every 60 seconds if the value is not between 20 and 40 and if the value has not changed by 5

  - * When the temperature is equal to or between 20 and 40 and the value has changed by 5 from STinit at least every 10 seconds.

4.8.3. Example 3 - Band Minimum and Maximum, Step and Initialization Value

```
Req: POST /bnd/ (Content-Format: application/link-format)
<coap://sensor.example.com/s/temperature>; rel="boundto";anchor="/a/t
emperature";bind="obs";pmin="10";pmax="60";bmn="20",bmx="40",st="5",i
v="20"

Res: 2.04 Changed
```

The above will result in:

- o STinit being set to 20 due to iv.

- o A state synchronization through an Observe:

  - * Every 60 seconds if the value is not between 20 and 40 and if the value has not changed by 5 from 20

  - * When the temperature is equal to or between 20 and 40 and the value has changed by 5 from STinit at least every 10 seconds.

4.8.4. Example 4 - Step and Initialization Value

Req: POST /bnd/ (Content-Format: application/link-format)
<coap://sensor.example.com/s/temperature>; rel="boundto";anchor="/a/temperature";bind="obs";pmin="10";pmax="60";st="5",iv="20"

Res: 2.04 Changed

The above will result in:

- o STinit being set to 20 due to iv.

- o A state synchronization through an Observe:

  - * Every 60 seconds if the temperature does not differ from STinit by 5.

  - * When the temperature differs from STinit by 5 at least every 10 seconds.

4.8.5. Example 5 - Band Minimum and Maximum, Band Step and Initial Value

Req: POST /bnd/ (Content-Format: application/link-format)
<coap://sensor.example.com/s/temperature>; rel="boundto";anchor="/a/temperature";bind="obs";pmin="10";pmax="60";bmn="20",bmx="40",bst="5",iv="15"

Res: 2.04 Changed

The above will result in:

- o A series of bands being created of a width of 5 with the seed value 15. Given bmn="20" and bmx="40" this effectively means that bands with the following thresholds(15,20],(20,25],(25,30],(30,35],(35,40] are created.

- o A state synchronization through an Observe:

  - * Every 60 seconds if the value is not between 20 and 40 (inclusive) and if the value has not entered into a new band.

  - * When the temperature is equal to or between 20 and 40 and the value changes between the bands at least every 10 seconds.

4.8.6. Example 6 - Band Minimum and Sample Number Window

Req: POST /bnd/ (Content-Format: application/link-format)
<coap://sensor.example.com/s/temperature>; rel="boundto";anchor="/a/temperature";bind="obs";pmin="10";pmax="60";bmn="50";snw="5"

Res: 2.04 Changed

The above will result in:

- o A state sychronization added to the queue at pmax or whenever the value changes and is equal to or above 50.

- o A state sychronization through an Observe occuring once 5 synchronizations have been added to the queue resulting in multiple values being synchronized between the source and destination resources.

5. Security Considerations

As per 5/[I-D.ietf-core-dynlink].

6. IANA Considerations

None.

7. Acknowledgements

Michael Koster for discussions leading to the creation of these notification band and initialization attributes.

8. Changelog

draft-groves-core-intparam-00

- o Initial version

9. References

9.1. Normative References

[I-D.ietf-core-dynlink]
Shelby, Z., Vial, M., Koster, M., and C. Groves, "Dynamic Resource Linking for Constrained RESTful Environments", draft-ietf-core-dynlink-01 (work in progress), October 2016.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[RFC5988] Nottingham, M., "Web Linking", RFC 5988, DOI 10.17487/RFC5988, October 2010, <http://www.rfc-editor.org/info/rfc5988>.

[RFC6690] Shelby, Z., "Constrained RESTful Environments (CoRE) Link Format", RFC 6690, DOI 10.17487/RFC6690, August 2012, <http://www.rfc-editor.org/info/rfc6690>.

9.2. Informative References

[I-D.ietf-core-senml] Jennings, C., Shelby, Z., Arkko, J., Keranen, A., and C. Bormann, "Media Types for Sensor Measurement Lists (SenML)", draft-ietf-core-senml-04 (work in progress), October 2016.

Authors' Addresses

Christian Groves
Huawei
Australia

Email: Christian.Groves@mail01.huawei.com

Weiwei Yang
Huawei
P.R.China

Email: tommy@huawei.com