CoRE Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 11, 2017

C. Groves
W. Yang
Huawei

March 10, 2017

SenML Options
draft-groves-core-senml-options-00

Abstract

SenML [I-D.ietf-core-senml] defines an initial set of base and regular attributes which are tied to a particular version of SenML. SenML also allows the definition of additional attributes by extending the syntax with a new label. Allowing the extension of attributes brings the problem of how do endpoints negotiate whether the new attribute can be used or not? This document discusses the issue and proposes some potential solutions to this issue.

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 11, 2017.

Table of Contents

1. Introduction

SenML [I-D.ietf-core-senml] defines an initial set of base and regular attributes which are tied to a particular version of SenML. SenML also allows the definition of additional attributes by extending the defined syntax with a new label. Allowing the extension of attributes brings the problem of how do endpoints negotiate whether the new attribute can be used or not?

For example: A CoAP client issues a GET that indicates support of SenML through the use of an CoAP Accept option. A CoAP server supports the SenML attributes defined in [I-D.ietf-core-senml] and in addition supports the Base Time Offset (BTO) [I-D.groves-core-senml-bto] attribute. The server responds using the BTO attribute.

```
[ {"bn": "urn:dev:ow:10e2073a01080063",
      "bt": 1320067464,
      "bto": 10,
      "bu": "%RH",
      "v": 21.2},
      { "v": 21.3},
      { "v": 21.4},
      { "v": 21.4},
      { "v": 21.5},
      { "v": 21.5},
      { "v": 21.5},
      { "v": 21.6},
      { "v": 21.7},
      { "v": 21.5},
   ...
```

Figure 1: Response with SenML using base time offset

As the CoAP client does not understand the "bto" attribute it will ignore the attribute. This means that the time information is lost for each of the SenML records. Whereas if the Server had not used "bto" the client would have been able to understand the information.

This is mainly a problem when the server provides a response to a message (i.e. GET) rather than when a client uses the SenML media type in a message (i.e. PUT). In this later case the client can modify its behaviour and not use an attribute based on an error response from the server.

A solution is needed to prevent incompatible attributes from being used.

2. Terminology

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].

See [I-D.ietf-core-senml] for further definitions.

3. Solution space analysis

The extension of protocols in a compatible manner is not a new problem. Some approaches to handle the are discussed below. The discussion below is not meant to be treatise on protocol extension but to highlight potential solutions.

3.1. Version Numbering

A version number is used to indicate when changes have been made to a syntax. In some protocols a major version is used to indicate non-backward compatible changes and a minor version is used to indicate backwards compatible changes. SenML does use a version number however it is an integer (no major and minor). It is used to indicate the version number of the media type format. The version does not appear in the media type format name. However presumably the proposed registrations (e.g. senml+json) imply the use of version number 10 (see sect.4.1/[I-D.ietf-core-senml]). If in the future there is a new version how does the endpoint negotiate which SenML version to use? A simple solution would be to create a new media type name incorporating the new version e.g. "senml_v11+json". A CoAP endpoint could then use an Accept option to indicate support for "senml+json" and/or "senml_v11+json".

This method does generally mean that for each new attribute and version the endpoints must at least understand all previously defined attributes. Although major versions in some protocols do not maintain backwards compatibility.

SenML does indicate that for certain representations i.e. EXI representation (application/senml+exi) that the schemaID number must be updated when the syntax is updated for a new attribute. This is effectively a version mechanism but the other representation formats do not follow this approach.

However the current SenML draft does allow the definition of additional attributes without increasing the SenML version number. Indeed there is a trend at the IETF that protocol versions change very rarely. Instead updates are incorporated via option or extension mechanisms.

Therefore it seems that an approach of utilising a version number for each additional new attribute does not seem appropriate for SenML.

3.2. Mandatory / Optional Indication

Some protocols use a mechanism to indicate whether a parameter is mandatory or optional to understand. This is based on the assumption that a sending endpoint knows the functions that a parameter/s relate to and can indicate whether the receiving endpoint must understand the parameter/s to implement the function. A receiving endpoint will anaylse the received parameters and if it does not understand the parameter it will check the mandatory/option tag to see what it should do. If the parameter is marked as mandatory then the receiving endpoint will generate an error. If the parameter is

marked as optional then the receiving endpoint will continue processing in knowledge that it doesn't need the parameter.

CoAP uses a mechanism similar to this for Options. By looking at the option number an endpoint can determine whether it is critical or not.

SenML does indicate that the version indicates the mandatory to understand attributes (sect. 4.3/[I-D.ietf-core-senml]). SenML also indicates that some attributes (i.e. base attributes) are optional but this is in context of "optional to be used" rather than "optional to be understood".

Whilst a method could be defined to indicate in SenML whether an attribute is mandatory or optional, its not clear that it would be useful. Given the number of use cases where CoAP can be used a server may not know which information in a SenML pack is relevant for a client. E.g. Whilst a server may return time information associated with a record it doesn't actually know whether it is useful to the client. The usefulness is application specific to the client.

Therefore it seems this approach is also not appropriate for SenML.

3.3. Options Mechanism

Some protocols allow the optional parts of the protocol to be negotiated during the initial protocol negotiation. For example the SIP protocol has the OPTIONS method [RFC3261]. The CLUE protocol [I-D.ietf-clue-protocol] also defines an extension method where an OPTIONS message is used to negotiate the protocol extensions. The benefits of such an approach is that two endpoints can negotiate which extensions they will use in a session ensuring compatible communications.

However these approaches assume an application level session where there are establishment, communication and release phases. SenML is a media type format primarly defined to be used with the HTTP and CoAP protocols. These protocols don't follow a session based approach.

HTTP/1.1 does have the OPTIONS method [RFC2616] however the use is largely undocumented. CoAP does not have an equivalent method.

CoAP does have a method for negotiating signalling through "Signaling Option Numbers" (sect.4.2/[I-D.ietf-core-coap-tcp-tls]). This however is more used to negotiate the properties of the signalling

connection than any elements of the CoAP payload (not withstanding the Blockwise transfer).

CoAP does have the OPTIONs mechanism allowing for the definition of optionality functionality associated with a CoAP message. It also defines the concept of critical and elective options. Two options related to Content-type/Content-format are "Content-Format" and "Accept". These options allow endpoints to indicate the media types are using or wish to use. HTTP also uses these options.

CoAP also allows a per message exchange of what is supported for a particular resource. This seems more appropriate than a protocol level negotiation of the support of SenML attributes.

This functionality is very close to what needs to be acheived for negotiating SenML attributes.

3.4. Media Type Definition and Parameters

Potentially the media type name could be used to indicate versions or extensions. This may be appropriate where there are seldom changes that affect the whole media type. However it may become unwieldy if the media type name is used to define combinations of SenML attributes, e.g. given 3 extensions a, b and c you could end up with media names / content formats for a, b, c, a+b, a+c, b+c, a+b+c. The problem gets worse each time an extension is added. It is made even worse because Table 7/[I-D.ietf-core-senml] defines 8 different content formats for SenML that would need to be updated. To allow combinations of these parameters on the media types defined in SenML it would need 56 Content-format code points. The content format range 0..255 for IETF specifications isn't particularly large.

[RFC6838] allows for the registration of media type parameters. This allows further companion information to be included along with the media type. Charset is a common parameter (See [RFC3023]). This information could be used to provide version or option information associated with a media type.

This appears to be a good solution to indicate if additional SenML attributes are supported in a media type. Whilst HTTP supports media type parameters, CoAP does not support media type parameters or extensions (i.e. see sect.10.2.2/[RFC7252]). Meaning that parameters cannot be used as a common approach for HTTP and CoAP. However this solution is used in section 16.9.1/[I-D.ietf-cose-msg] which takes the approach of defining an optional parameter for the "application/cose". It then assigns multiple CoAP Content-Formats for the values associated with the optional parameter (see sect.16.10/[I-D.ietf-cose-msg]).

4. Solution Proposal

There doesn't appear to be one outstanding approach for negotiating SenML attributes common to both HTTP and CoAP. The authors believe that a hybrid approach as per [I-D.ietf-cose-msg] is needed in order to be able to negotiate which SenML extension attributes are used.

The first part would be the definition of a optional media-type parameter that allows an endpoint utilising HTTP to indicate the SenML extension attributes that it is using or accepts. This could be in the form of a comma separated string list of SenML labels from those registered in the SenML label registry. Only attributes NOT defined in [I-D.ietf-core-senml] would be allowed.

e.g. Content-type: application/senml+json; ext=abc,xyz

In order to allow this functionality in the base version of SenML an optional parameter would be needed in the media type registrations in sect.11.3/[I-D.ietf-core-senml].

i.e. o Optional parameter: SenML extensions

This parameter indicates which SenML extensions are associated with the media type. The parameter is defined by the following ABNF:

```
SenML-ext = "ext" "=" <"> senml-label *("," senml-label) <">
; Note: this follows the {{RFC2616}} quoted-string form.
; senml-label is the label string from the list of IANA
; registered SenML labels.
; Only non-{{I-D.ietf-core-senml}} labels are allowed.
```

For example: ext="a,b,c";

If the group decides that there will only ever be a small number of SenML extensions then the simplest approach would be to follow [I-D.ietf-cose-msg] and define multiple CoAP content formats associated with potential extensions. This would be done in which ever document defines the SenML extension. For example [I-D.groves-core-senml-bto] would add the following to the IANA considerations section:

Media Type	Encoding	ID	Reference
application/senml+json; ext="bto"		TBD	TBD
application/sensml+json; ext="bto"		TBD	TBD
application/senml+cbor; ext="bto"		TBD	TBD
application/sensml+cbor; ext="bto"		TBD	TBD
application/senml+xml; ext="bto"		TBD	TBD
application/sensml+xml; ext="bto"		TBD	TBD
application/senml+exi; ext="bto"		TBD	TBD
application/sensml+exi; ext="bto"		TBD	TBD

Table 1: New CoAP content formats

Text would also need to be added to [I-D.ietf-core-senml] to describe the procedure for support of the media type parameter in CoAP.

If issuing potentially a large number of content-format numbers is problematic a separate approach could be taken. A new CoAP option could be defined to allow media-type parameters to be carried in CoAP messages when then Content-Format or Accept options are used. As the Content-Format and Accept options may be used in the same request (with two different media types) two new options would be required, one for the media-type parameters associated with the Content-Format Option and one for the media-type parameters associated with the Accept option.

Editor's note: Alternatives could include defining the option for both CoAP and HTTP. However this would likely mean that the option would become specific to SenML extensions rather than a general mechanism for carrying media type parameters.

As CoAP only allows a single Content-Format to be carried in the Content-Format and Accept options it would be straight forward to define an option that allows media-type parameters to be carried. One complication is that the encoding and syntax of the media-type parameters is up to the media-type definition. It could be a string, integer, binary, etc. Therefore the option value would need to be an opaque sequence of bytes. If the scope was limited to SenML then the option format would be a narrowed to a string of labels.

As multiple parameters could be defined for a media-type the mechanism must allow multiple media type parameter to be signalled in a CLUE message. One possible method is to define the option value syntax to allow multiple parameter to be specified as a single parameter value. Alternatively multiple instances of the option could be used in the CoAP message. This method is indicated in the IANA registration by allowing the option multiple times.

If a new generic option is defined it's not clear that [I-D.ietf-core-senml] would be the best place to define the option. If the scope is limited then [I-D.ietf-core-senml] would be appropriate. Whichever draft defines the options it would need to define them for registration with IANA along the lines of:

Number	Name	Reference
TBD	AMTP	TBD
TBD	CFMTP	TBD

Table 2: New CoAP Option Numbers

4.1. Accept Media Type Parameter (AMTP) Option

o The meaning of the option in a request: It indicates the media-type parameters associated with the content-format (media-type) specified in the Accept option.

Note: Some content-formats contain media-type parameters as part of the content-format ID registrations. The AMTP option SHALL not be used with these CoAP content-formats.

o The meaning of the option in a response: Not used.

o Whether the option is critical or elective: Critical as per the Accept option.

o Whether the option is Safe-to-Forward: Safe as per the Accept option.

Note: Potentially it could be unsafe to forward an opaque byte sequence that the proxy does not understand. However processing this option should only be done within the context of the media-type specified by the Accept option.

o The format and length of the option's value: A variable length opaque sequence of bytes. The encoding of the bytes is defined as per the syntax for the parameters in the media-type definition document.

o Whether the option must occur at most once or whether it can occur multiple times: Multiple times. Each instance containing a seperate media-type parameter. Whether the option can be included multiple times for the one media type parameter is dependent on whether the media-type definition allows for multiple instances of the one media type parameter.

o Default value: None unless the media-type indicated in the accept option defines a default parameter/s value.

4.2. Content-Format Media-Type Parameter Option

o The meaning of the option in a request: When used together with the Content-Format option it indicates the media-type parameters associated with the content-format (media-type) specified in the content-format option.

Note: Some content-formats contain media-type parameters as part of the content-format ID registrations. The CFMTP option SHALL not be used with these CoAP content-formats.

o The meaning of the option in a response: When used together with the Content-Format option it indicates the media-type parameters associated with the content-format (media-type) specified in the content-format option.

o Whether the option is critical or elective: As per the Content-Format option it is elective.

o Whether the option is Safe-to-Forward: Safe as per the Content-format option.

Note: Potentially it could be unsafe to forward an opaque byte sequence that the proxy does not understand. However processing this option should only be done within the context of the media-type specified by the Content-Format options.

o The format and length of the option's value: A variable length opaque sequence of bytes. The encoding of the bytes is defined as per the syntax for the parameters in the media-type definition document.

o Whether the option must occur at most once or whether it can occur multiple times: Multiple times. Each instance containing a seperate media-type parameter. Whether the option can be included multiple times for the one media type parameter is dependent on whether the media-type definition allows for multiple instances of the one media type parameter.

o Default value: None unless the media-type indicated in the content-format option defines a default parameter/s value.

5. Security Considerations

SenML security issues are described in [I-D.ietf-core-senml]. Some extra considerations are indicated above.

6. IANA Considerations

Section 4 discusses potential IANA registrations.

7. Acknowledgements

TBD

8. Changelog

TBD

9. References

9.1. Normative References

[I-D.groves-core-senml-bto]
Groves, C. and W. Yang, "SenML Base Time Offset Attribute", draft-groves-core-senml-bto-00 (work in progress), October 2016.

[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.

[I-D.ietf-cose-msg]
Schaad, J., "CBOR Object Signing and Encryption (COSE)", draft-ietf-cose-msg-24 (work in progress), November 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>.

[RFC2616] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, DOI 10.17487/RFC2616, June 1999, <http://www.rfc-editor.org/info/rfc2616>.

[RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, DOI 10.17487/RFC3261, June 2002, <http://www.rfc-editor.org/info/rfc3261>.

[RFC6838] Freed, N., Klensin, J., and T. Hansen, "Media Type Specifications and Registration Procedures", BCP 13, RFC 6838, DOI 10.17487/RFC6838, January 2013, <http://www.rfc-editor.org/info/rfc6838>.

[RFC7252] Shelby, Z., Hartke, K., and C. Bormann, "The Constrained Application Protocol (CoAP)", RFC 7252, DOI 10.17487/RFC7252, June 2014, <http://www.rfc-editor.org/info/rfc7252>.

9.2. Informative References

[I-D.ietf-clue-protocol] Presta, R. and S. Romano, "CLUE protocol", draft-ietf-clue-protocol-13 (work in progress), February 2017.

[I-D.ietf-core-coap-tcp-tls] Bormann, C., Lemay, S., Tschofenig, H., Hartke, K., Silverajan, B., and B. Raymor, "CoAP (Constrained Application Protocol) over TCP, TLS, and WebSockets", draft-ietf-core-coap-tcp-tls-07 (work in progress), March 2017.

[RFC3023] Murata, M., St. Laurent, S., and D. Kohn, "XML Media Types", RFC 3023, DOI 10.17487/RFC3023, January 2001, <http://www.rfc-editor.org/info/rfc3023>.

Authors' Addresses

Christian Groves
Huawei
Australia

Email: Christian.Groves@mail01.huawei.com

Weiwei Yang
Huawei
P.R.China

Email: tommy@huawei.com