core   B. Greevenbosch
Internet-Draft   Huawei Technologies
Intended status: Informational   June 12, 2012
Expires: December 14, 2012

                      CoAP Profile Description Format
                 draft-greevenbosch-core-profile-description-00

Abstract

   This document provides a profile description format, that can be used
   to express capabilities of a CoAP server.

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 December 14, 2012.

Copyright Notice

Table of Contents

1. 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 [RFC2119].

[2](#). **Introduction**

   The Constrained Application Protocol (CoAP) [I-D.ietf-core-coap] is a
   RESTful protocol for constrained nodes and networks.

   Often, a client first learns about a resource through the link format
   [I-D.ietf-core-link-format]. The link format only provides basic
   information, for example the resource URL. However, it would be good
   if the client could get more extensive information on the resources
   when required. This document defines a profile description format,
   which can be used to signal several parameters about resources and
   their servers.

   One of the main features of the CoAP protocol is the ability to
   include CoAP options. These options can be either elective or
   critical. A message with an unsupported critical option will be
   rejected, whereas unsupported elective options will be ignored.

   Even though it is well defined how the server must respond to
   unsupported options, it is useful for the client to know which
   options are supported in advance. In this way, it can determine
   which options are viable to use in a transaction, and which features
   cannot be exploited. This specification allows signalling of the
   supported options by the resource.

   Another important feature of a server is which media types it
   supports. This document allows signalling of that information too.

   It is anticipated that more profile descriptions will become
   available in the future.

3. The Profile Interface

   The profile description can be obtained through a link with interface
   type "core#p" (see also [I-D.shelby-core-interfaces]). The URI of
   the profile description should be equal to the URI of the associated
   resource, but with a "/p" suffix. For example, if the client wants
   to get profile description on a server witn URI "www.example.org", it
   can send a GET request for "coap://www.example.org/p".

   If the profile of a particular resource is available, it should use
   the same naming convention. For example, the profile of the sensor
   "coap://www.example.org/s/tmp" (on the same server) can be acquired
   with a GET to: "coap://www.example.org/s/tmp/p".

[4](). The Profile Format

The profile description is formatted as a JSON document. It consists of several profile fields, each of which has associated parameters.

[4.1](). The Options "op" profile field

The Options "op" profile field contains a list of options that are supported by a resource. It has the format depicted in Figure 1, where op1, op2, ... are integers representing the option numbers of the supported options, as defined in [[I-D.ietf-core-coap]()] and/or through IANA. The option numbers MUST appear in numerical order, starting with the lowest number.

```
                    "op":[op1,op2,...]
```

Figure 1: "op" syntax

When including the "op" profile field in the profile description of a resource, all option numbers of the CoAP options supported by that resource MUST be included. Options that are not supported by the resource MUST NOT be included in the "op" profile field.

If the "op" profile field is available, the receiving party SHALL assume a non-listed option is not supported by the associated resource.

[4.2](). The Media-Types "mt" profile field

The Media-Types "mt" profile field indicates which media-types are supported. It has the format depicted in Figure 2, where mt1, mt2, ... are integers representing the media-type numbers of the supported media-types, as defined in [[I-D.ietf-core-coap]()] and/or through IANA. The media-type numbers MUST appear in numerical order, starting with the lowest number.

```
                    "mt":[mt1,mt2,...]
```

Figure 2: "mt" syntax

When including the "mt" profile field in the profile description of a resource, all media-types of the CoAP options supported by that resource MUST be included. Media-types that are not supported by the resource MUST NOT be included in the "mt" profile field.

If the "mt" profile field is available, the receiving party SHALL assume a non-listed media-type is not supported by the associated resource.

5. Usage with servers, resources and batches

   A CoAP server may cater for multiple resources. In addition, the
   document [I-D.shelby-core-interfaces] provides a batch mechanism that
   allows querying/manipulating multiple resources at once.

   The "/p" URI suffix allows obtaining profile description on each of
   these three levels. For example, a GET to "http://www.example.org/p"
   would return profile description about the complete server, whereas a
   GET to "http://www.example.org/s/tmp/p" only returns profile
   description about the temperature sensor "tmp". Lastly, a GET to
   "http://www.example.org/s/p", where "/s" is a batch of sensors, would
   return common profile description for all these sensors.

   The hierarchical nature of these three entities implies that profiles
   can only become more restrictive when going to a lower level. For
   example, a server may support the block option, whereas one of its
   resources does not. In that case, the profile of the server would
   indicate block option support, whereas the profile of the resource
   would not do so.

6. Forward compatibility

   To allow addition of new profile fields in the future, unknown
   profile fields SHOULD be ignored by the client.

7. **Examples**

   The following is an example of a camera sensor at
   "coap://www.example.org/s/cam", that supports the "Content-Type" (1),
   "Proxy-Uri" (3), "ETag" (4), "Uri-Host" (5), "Uri-Port" (7), "Uri-
   Path" (9), "Token" (11) and "Block2" (17) options.

   The supported media types are "text/plan" (0), "application/
   link-format" (40) and "application/json" (50).

```
                Req: GET /s/cam/p

                Res: 2.05 Content (application/json)
                {
                  "op":[1,3,4,5,7,9,11,17],
                  "mt":[0,40,50]
                }
```

8. **Open topics**

   o How to signal the client profile?

   o Which other profile data needs signalling?

   o A natural media type for a camera would be JPEG. Therefore the
      "image/jpeg" media type may need CoAP registration.

   o Currently, support of observe can be signalled in the link format
      as well as through the mechanism described here.

   o Section 5 describes the hierchical nature of servers, batches and
      sensors, and how the profiles fit in this hierarchy. It is the
      question how to handle sensors that do not provide profile data.
      One option could be to assume profile description is inherited,
      unless separately specified. This may be reasonable as often the
      server provides most functionality to all of its resources.
      However, if we don't allow inheritance of the profile description,
      it would be needed to specify the profile description for all
      resources separately.

   o Is it needed to signal the profile description on a resource
      basis, or would it be enough to have only one, associated with the
      server?

   o Fix the order in which the profile fields must appear?

   o Make a distinction between "critical" and "elective" profile
      fields?

9. Security Considerations

For general CoAP security considerations see [I-D.ietf-core-coap].

In an unprotected environment, an attacker can change the profile
description. For example, the list of supported options may be
changed. This could cause the client to make a wrong decision on
which mechanisms to use. However, such a threat is normal in
environments that lack secure authentication.

10. IANA Considerations

o A registry for profile fields as well as possible values needs to
   be set up.

o The "core#p" interface must be registered in due time.

11. Acknowledgements

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

12. References

12.1. Normative References

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

12.2. Informative References

[I-D.ietf-core-coap]
           Shelby, Z., Hartke, K., Bormann, C., and B. Frank,
           "Constrained Application Protocol (CoAP)",
           draft-ietf-core-coap-09 (work in progress), March 2012.

[I-D.ietf-core-link-format]
           Shelby, Z., "CoRE Link Format",
           draft-ietf-core-link-format-12 (work in progress),
           May 2012.

[I-D.shelby-core-interfaces]
           Shelby, Z., "CoRE Link Format",
           draft-shelby-core-interfaces-02 (work in progress),
           March 2012.

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