Internet Engineering Task Force
Internet-Draft
Intended status: Informational
Expires: April 26, 2012

Z. Cao, Ed.
China Mobile
Y. Ma
Hitachi R&D China
H. Deng
China Mobile

October 24, 2011

HTTP-COAP Proxy Discovery using Link-format
draft-cao-core-pd-00

Abstract

This document discusses the problem of HTTP-COAP proxy discovery and proposes a method of using Link-format to do the job.

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 April 26, 2012.

Copyright Notice

Copyright (c) 2011 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 a RESTful protocol designed for constrained devices. The ultimate goal of CoAP is to enable the "Web of Things" concept, which connects the smart sensor network with the global internet. Although CoAP has been implemented on various platforms, the rest of web is still dominated by HTTP. As a result, it is desirable to interconnect the HTTP and CoAP via some intermediary proxy. For example, the CoAP sensor client in the constrained network can access and update resources on the HTTP server, and also the HTTP client on the web can access and/or update resources on the CoAP server.

There are already some works discussing how to map HTTP to CoAP and vice versa. The basic mapping between HTTP and CoAP is described in Section 8 of [I-D.ietf-core-coap] . Further details of implementing the proxy, internal procedures and design choices are described in[I-D.castellani-core-http-mapping] .

Static configuration of HTTP-CoAP proxies is a straightforward way for the client to access the server. However, in many situations, static configuration is not enough to meet the requirements. For example, if the HTTP client would like to access a certain type of resource (temperature or humidity in a certain location, etc.), it is required that the client would find an appropriate proxy to serve the content.

1.1. Requirements Language

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 RFC 2119 [RFC2119].

2. Problem Formation

We divide the problem into two separated parts. The first is how a CoAP client discovers a proxy to access the HTTP server. For example, the CoAP sensors want to report or get some information to a Web server. In this case, the CoAP sensor only acts as a client. In static configuration, the CoAP client is configured via DHCP or RSRA. But in dynamic environment, a mechanism for dynamic configuration is desired. This document mainly discusses this aspect.

The other case is how a HTTP client discovers a proxy to access the CoAP server. For example, the HTTP client wants to access a certain type of information in the constrained network, and would discover the proxy to the exact constrained sensor. In this case, the HTTP

Client only accesss the sensor indirectly. In this case, the HTTP Client only needs to know the address or the domain name of the proxy node, and the proxy forwards the requests to the sensor node according to the sub-domain information or the path included in the URI within the request. But in this case, we believe that the DNS-SD infrastures are sufficient to handle this problem. For example, [I-D.vanderstok-core-bc] has described detailed considerations of a DNS-SD based proxy discovery method for Building Control use cases. So, in this document we will not talk about this direction.

3. Link-format Proxy Discovery

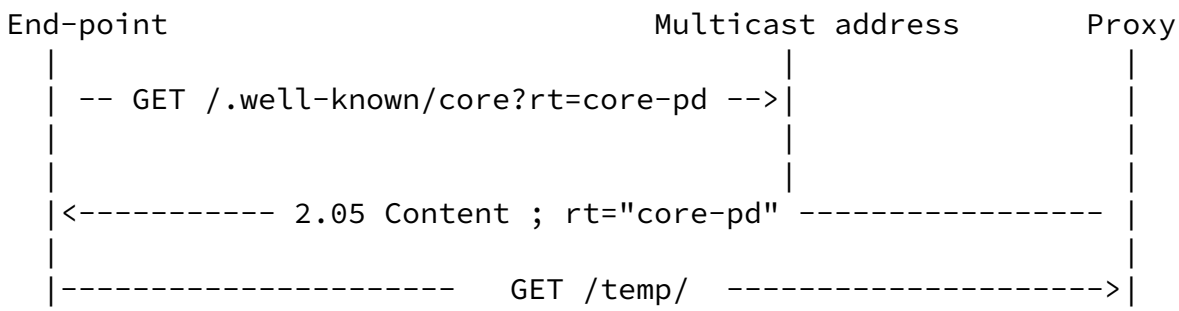
Before the CoAP sensor makes use of the CoAP-HTTP proxy, it must know the location of the proxy. There can be multiple ways to discover the proxy'ss location, including both static and dynamic methods. DHCP is one way to do that, and documented in another document. This document describes one way to discover the proxy by the CoRE link format [I-D.ietf-core-link-format].

Note: Think of the way the user is configured with the http proxy in the enterprise network.

Discovery is performed by sending a multicast GET request to /.well-known/core and including a Resource Type (rt) parameter [I-D.ietf-core-link-format] with the value "core-pd" in the query string. Upon success, the response will contain a payload with a link format entry for each proxy discovered. The multicast IP address used will depend on the scope required and the multicast

capabilities of the network. (If determined, IANA actions are required to assign a multicast address for this purpose)

The following example shows an end-point discover a locally available CoAP-HTTP proxy. The CoAP end-point sends a multicast GET request to the multicast address in the domain carrying a resource type "core-pd" indicating its discovery of a local proxy. Then the serving proxy responds the request with the rt="core-pd" and the address of the proxy is carried within the Content payload. Afterwards, the CoAP sensor initiates the data-plane communication with the proxy directly.



Req: GET coap://[ff02::1]/.well-known/core?rt=core-pd

Res: 2.05 Content
fe80::ff; rt="core-pd";

4. Acknowledgements

Some ideas in this document are according to the discussion between Zach Shelby on the problem.

5. IANA Considerations

If the ideas in this document is determined by the working group, IANA actions are required to assign a multicast address for the purpose of HTTP-CoAP proxy discovery.

6. Security Considerations

TBD.

7. References

7.1. Normative References

[I-D.castellani-core-http-mapping]
Castellani, A., Loreto, S., Rahman, A., Fossati, T., and E. Dijk, "Best practices for HTTP-CoAP mapping implementation", draft-castellani-core-http-mapping-01 (work in progress), July 2011.

[I-D.ietf-core-coap]
Shelby, Z., Hartke, K., Bormann, C., and B. Frank, "Constrained Application Protocol (CoAP)", draft-ietf-core-coap-07 (work in progress), July 2011.

[I-D.ietf-core-link-format]
Shelby, Z., "CoRE Link Format", draft-ietf-core-link-format-07 (work in progress), July 2011.

[I-D.vanderstok-core-bc]
Stok, P. and K. Lynn, "CoAP Utilization for Building Control", draft-vanderstok-core-bc-04 (work in progress), July 2011.

7.2. Informative References

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

Authors' Addresses

Zhen Cao (editor)
China Mobile
China,
China

Phone:
Email: zehn.cao@gmail.com

Yuanchen Ma
Hitachi R&D China

Phone:
Fax:
Email: ycma@hitachi.cn
URI:

Hui Deng
China Mobile

Phone:
Fax:
Email: denghui@chinamobile.com
URI: