

CoRE Working Group
Internet-Draft
Intended status: Informational
Expires: September 11, 2016

B. Silverajan
Tampere University of Technology
March 10, 2016

CoAP Protocol Negotiation
draft-silverajan-core-coap-protocol-negotiation-02

Abstract

CoAP has been standardised as an application-level REST-based protocol. This document introduces a way forward for CoAP clients and servers to exchange resource representations when multiple transports exist at an endpoint, by agreeing upon alternate locations as well as transport and protocol configurations.

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, 2016.

Copyright Notice

Copyright (c) 2016 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	2
2. Rationale	2
3. Goals	3
4. New Link Attribute and Relation types	4
5. Examples	4
6. IANA Considerations	6
7. Security Considerations	6
8. Acknowledgements	6
9. References	6
9.1. Normative References	6
9.2. Informative References	6
Appendix A. Change Log	7
A.1. From -01 to -02	7
A.2. From -00 to -01	7
Author's Address	7

1. Introduction

In the Constrained Application Protocol (CoAP) [RFC7252], resources are uniquely represented by Uniform Resource Identifiers (URIs). Using URIs, CoAP endpoints, such as clients, origin servers and proxies, are able to exchange representations using REST-based methods. A URI in CoAP serves two purposes. Firstly, it functions as a locator, by specifying the network location of the endpoint hosting the resource, and the underlying transport used by CoAP for accessing the resource representation. Secondly, it identifies the name of the specific resource found at that endpoint together with its namespace, or resource path.

This draft proposes a new link format attribute as well as a new link relation type that together enable an origin server to serve a resource from other protocol configurations or endpoints. CoAP clients then interact with an origin server's CoRE resource discovery interface to obtain a set of links describing alternate locations of resources.

2. Rationale

Ongoing activity and discussion in CoRE has revealed the need to convey CoAP messages over not just UDP and DTLS, but also over alternative transports such as SMS [I-D.becker-core-coap-sms-gprs], TCP [I-D.ietf-core-coap-tcp-tls] and WebSockets [I-D.savolainen-core-coap-websockets]. The underlying transport to be used by CoAP is identified by the scheme component of a new URI format, as described in [I-D.silverajan-core-coap-alternative-transports].

Working group discussions and feedback for the new URI format's design criteria indicated eventually that two sets of requirements for CoAP over alternative transports were deemed to be important: The first focuses on how to express location information within the new URI format in order to reach the origin server hosting the CoAP resource, the alternative transport used as well as the path and resource name that uniquely identifies the specific resource within the server. The scope of [I-D.silverajan-core-coap-alternative-transports] is focused towards this first set of requirements, as well as an analysis of transport layer properties.

The second set of requirements pertains to accessing CoAP resources when multiple transports are present at a CoAP endpoint, by separating endpoint location information from the identification of a CoAP resource. By doing so, both CoAP clients can better discern if the same CoAP resource representation can continue to be retrieved from a CoAP server over other transports. The multiple transport problem cannot be directly solved by simply introducing a new URI format. Therefore, [I-D.silverajan-core-coap-alternative-transports] provides a categorization of CoAP nodes based on their ability to use multiple transports to convey CoAP messages, whilst its main emphasis is in providing guidance for implementing support for CoAP over an alternative transport. Instead, the issue of multiple transports for a CoAP resource is addressed in this document.

3. Goals

Should an origin server wish to serve a resource over multiple transports, a single CoAP URI cannot be used to express the identity of the resource independently of alternate underlying transports or protocol configurations. Similarly, if the server wishes to serve representations of the resource from a different endpoint and path, the URI mechanism is incapable of capturing the relationship between these alternate representations or locations.

However, providing a way to express such relationships would be useful in the following cases:

1. CoAP clients interacting with Type T1 or T2 CoAP origin servers (see Section 3 of [I-D.silverajan-core-coap-alternative-transports]) either before or during an ongoing transaction to communicate using CoAP over a different protocol configuration or alternative transport.
2. Avoiding URI aliases [WWWArchv1], where a single resource is represented with multiple URIs, without describing relations among the alternate representations.

3. Allowing intermediate nodes such as CoAP-based proxies to intelligently cache and respond to CoAP clients with the same resource representation requested over alternative transports or server endpoints.
4. Ability to separate the CoAP resource paths from web-based CoAP endpoint path in a URI.

4. New Link Attribute and Relation types

A CoAP server wishing to allow interactions with resources from multiple locations or transports can do so by specifying the Transport Type "tt" link attribute, which is an opaque string. Multiple transport types can be included in the value of this parameter, each separated by a space. In such cases, transport types appear in a prioritised list, with the most preferred transport type by the CoAP server specified first and the lowest priority transport type last.

At the same time, each transport type supported by the server is also described with an "altloc" link relation type. The "altloc" relation type specifies a URI (containing the URI scheme, authority and optionally path) providing an alternate endpoint location up to but not including the resource path of a representation.

Both "tt" and "altloc" are optional CoAP features. If supported, they occur at the granularity level of an origin server, ie. they cannot be applied selectively on some resources only. Therefore "altloc" is always anchored at the root resource ("/"). Additionally, the "tt" link attribute and "altloc" relation type can be ignored by unsupported CoAP clients.

(TBD: As type T1 nodes may not have all transports active at all times, should a lifetime value be reflected in server responses?)

5. Examples

Example 1 shows a CoAP server returning all transport types and the alternate resource locations to a CoAP client performing a CoAP Request to `./well-known/core`

In this case, the server supplies two different locations to interact with resources using CoAP over TCP. At the same time, the path to the WebSocket endpoint is provided in addition to the FQDN of the server, for using CoAP over WebSockets.

```
REQ: GET /.well-known/core

RES: 2.05 Content
</sensors>;ct=40;title="Sensor Index", tt="tcp ws sms",
</sensors/temp>;rt="temperature-c";if="sensor",
</sensors/light>;rt="light-lux";if="sensor",
<coap+tcp://server.example.com/>;rel="altloc",
<coap+tcp://server.example.net/>;rel="altloc",
<coap+ws://server.example.com/ws-endpoint/>;rel="altloc",
<coap+sms://001234567/>;rel="altloc"
```

Figure 1: Example of Server response

Example 2 shows a CoAP client actively soliciting a CoAP server for all supported transport types and protocol configurations.

```
REQ: GET /.well-known/core?tt=*

RES: 2.05 Content
</sensors>;tt="tcp sms ws"
<coap+tcp://server.example.com/>;rel="altloc",
<coap+tcp://server.example.net/>;rel="altloc",
<coap+ws://server.example.com/ws-endpoint/>;rel="altloc",
<coap+sms://001234567/>;rel="altloc"
```

Figure 2: CoAP client discovering transports supported by a CoAP server.

Example 3 shows a CoAP client explicitly soliciting support for a specific transport type using a query filter parameter.

```
REQ: GET /.well-known/core?tt=sms

RES: 2.05 Content
</sensors>;tt="tcp sms ws"
<coap+sms://001234567/>;rel="altloc"
```

Figure 3: CoAP client looking for a specific transport to use with a CoAP server.

6. IANA Considerations

New link attributes and link relations need to be registered.

7. Security Considerations

Probably lots. (TBD)

8. Acknowledgements

Thanks to Klaus Hartke for comments and reviewing this draft, and Teemu Savolainen for initial discussions about protocol negotiations and lifetime values.

9. References

9.1. Normative References

[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.becker-core-coap-sms-gprs]
Becker, M., Li, K., Kuladinithi, K., and T. Poetsch, "Transport of CoAP over SMS", draft-becker-core-coap-sms-gprs-05 (work in progress), August 2014.

[I-D.ietf-core-coap-tcp-tls]
Bormann, C., Lemay, S., Technologies, Z., and H. Tschofenig, "A TCP and TLS Transport for the Constrained Application Protocol (CoAP)", draft-ietf-core-coap-tcp-tls-01 (work in progress), November 2015.

[I-D.savolainen-core-coap-websockets]
Savolainen, T., Hartke, K., and B. Silverajan, "CoAP over WebSockets", draft-savolainen-core-coap-websockets-05 (work in progress), October 2015.

[I-D.silverajan-core-coap-alternative-transports]
Silverajan, B. and T. Savolainen, "CoAP Communication with Alternative Transports", draft-silverajan-core-coap-alternative-transports-09 (work in progress), December 2015.

[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>>.

[WWWArchv1] <http://www.w3.org/TR/webarch/#uri-aliases>, "Architecture of the World Wide Web, Volume One", December 2004.

Appendix A. Change Log

A.1. From -01 to -02

Document lifetime extension, references updated

A.2. From -00 to -01

Reworked "Introduction" section, added "Rationale", and "Goals" sections.

Author's Address

Bilhanan Silverajan
Tampere University of Technology
Korkeakoulunkatu 10
FI-33720 Tampere
Finland

Email: bilhanan.silverajan@tut.fi