ONEM2M
TECHNICAL SPECIFICATION

Document Number	oneM2M-TS-0008-CoAP Protocol Binding-V-0.5.0
Document Name:	CoAP Protocol Binding Technical Specification
Date:	2014-08-01
Abstract:	The specification will cover the protocol specific part of communication protocol used by oneM2M compliant systems as 'CoAP binding'

This Specification is provided for future development work within oneM2M only. The Partners accept no liability for any use of this Specification.

The present document has not been subject to any approval process by the oneM2M Partners Type 1. Published oneM2M specifications and reports for implementation should be obtained via the oneM2M Partners' Publications Offices.

About oneM2M

The purpose and goal of oneM2M is to develop technical specifications which address the need for a common M2M Service Layer that can be readily embedded within various hardware and software, and relied upon to connect the myriad of devices in the field with M2M application servers worldwide.

More information about oneM2M may be found at: http//www.oneM2M.org

Copyright Notification

Notice of Disclaimer & Limitation of Liability

The information provided in this document is directed solely to professionals who have the appropriate degree of experience to understand and interpret its contents in accordance with generally accepted engineering or other professional standards and applicable regulations. No recommendation as to products or vendors is made or should be implied.

NO REPRESENTATION OR WARRANTY IS MADE THAT THE INFORMATION IS TECHNICALLY ACCURATE OR SUFFICIENT OR CONFORMS TO ANY STATUTE, GOVERNMENTAL RULE OR REGULATION, AND FURTHER, NO REPRESENTATION OR WARRANTY IS MADE OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR AGAINST INFRINGEMENT OF INTELLECTUAL PROPERTY RIGHTS. NO oneM2M PARTNER TYPE 1 SHALL BE LIABLE, BEYOND THE AMOUNT OF ANY SUM RECEIVED IN PAYMENT BY THAT PARTNER FOR THIS DOCUMENT, WITH RESPECT TO ANY CLAIM, AND IN NO EVENT SHALL oneM2M BE LIABLE FOR LOST PROFITS OR OTHER INCIDENTAL OR CONSEQUENTIAL DAMAGES. oneM2M EXPRESSLY ADVISES ANY AND ALL USE OF OR RELIANCE UPON THIS INFORMATION PROVIDED IN THIS DOCUMENT IS AT THE RISK OF THE USER.

Contents

1 Scope

The specification will cover the protocol specific part of communication protocol used by oneM2M compliant systems as ‘RESTful CoAP binding’.

2 References

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references,only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

2.1 Normative references

The following referenced documents are necessary for the application of the present document.

[1] IETF RFC 7252: "The Constrained Application Protocol (CoAP)"

[2] oneM2M TS-0004: Protocol TS

[3] IETF draft: “Blockwise transfers in CoAP”, draft-ietf-core-block-15

[4] oneM2M Security Solutions Technical Specification

[5] IETF RFC 6347 "Datagram Transport Layer Security Version 1.2"

2.2 Informative references

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

[i.1] oneM2M Drafting Rules (http://member.onem2m.org/Static_pages/Others/Rules_Pages/oneM2M-Drafting-Rules-V1_0.doc)

3 Definitions, symbols, abbreviations and acronyms

3.1 Definitions

For the purposes of the present document, the following terms and definitions apply:

3.2 Symbols

For the purposes of the present document, the following symbols apply:

3.3 Abbreviations

For the purposes of the present document, the following abbreviations apply:

3.4 Acronyms

For the purposes of the present document, the following abbreviations apply:

4 Conventions

The key words “Shall”, ”Shall not”, “May”, ”Need not”, “Should”, ”Should not” in this document are to be interpreted as described in the oneM2M Drafting Rules [i.1]

5 Overview

This specification defines how to map oneM2M APIs into CoAP messages and vice versa.

5.1 Required Features

This section specifies required features from CoAP[1] to be properly mapped into oneM2M APIs:

- CoAP message types for message correlation, including CON, NON, ACK, and RST, shall be supported.
- GET, PUT, POST and DELETE methods shall be supported.
- Related response code shall be supported.
- CoAP defines a single set of options that are used in both requests and responses. Related options shall be supported.
- Other features of CoAP shall be supported, such as blockwise transfers.(TBD)

5.2 Message Format

This section specifies details about the CoAP [1] message format:

- CoAP message occupies the data section of one UDP datagram.
- CoAP message format supports a 4-byte fixed-size header.
- Fixed-size header is followed by a Token value of length 0 or 8 bytes.
- The Token value is followed by a sequence of zero or more CoAP Options in TLV format.
- CoAP Options are followed by the payload part.

For more details on the CoAP Message Format and the supported Header Fields, refer [1].

5.3 Caching

CoAP [1] supports caching of responses to fulfill future equivalent requests to the same resource. Caching is supported using freshness and validity information carried with CoAP [1] responses.

5.3.1 Freshness

- CoAP server shall use Max-Age CoAP Option to specify the explicit expiration time for the CoAP Response’s resource representation. This indicates that the response is not fresh after its age is greater than the specified number of seconds.

- Max-Age Option defaults to a value of 60 (seconds). In case, Max-Age Option is not present in the cacheable response, the response shall not be considered fresh after its age is greater than 60 seconds.
- The CoAP server shall set the Max-Age Option value to 0 (zero) to prevent or disable caching.
- The CoAP client, having a fresh stored response, can make new request matching the request for that stored response. In this case, the new response shall invalidate the old response.

5.3.2 Validity

- A CoAP endpoint with stored responses but not able to satisfy subsequent requests (for example, the response is not fresh), shall use the ETag Option to perform a conditional request to the CoAP server where the resource is hosted.
- If the cached response with the CoAP client is still valid, the server shall include the Max-Age Option in the response along with a code of 2.03 - Valid. This shall update the freshness of the cached response at the CoAP client.
- If the cached response with the CoAP client is not valid, the server shall respond with an updated representation of the resource with response code 2.05 – Content. The CoAP client shall use the updated response to satisfy request and may also replace/update the stored or cached response.

5.4 Blockwise Transfers

CoAP Block [3] Option shall be used for handling cases where oneM2M resource representations will need to transfer large payloads e.g. firmware, software updates. Instead of relying on IP fragmentation, CoAP Block Option shall be used for transferring multiple blocks of information in multiple request-response pairs.

Using Block Options, larger resource representations can be fragmented and reassembled by CoAP independently of the lower layers as well as the above application. The CoAP Block1 Option shall be used to define the size of the blocks used for oneM2M requests and the CoAP Block2 Option shall be used to define the size of the blocks used for oneM2M responses. Refer [3] for further details.

6 oneM2M Protocol Mapping

6.1 Primitive Mapping

6.1.1 Request primitive to CoAP Request

The oneM2M request operation shall be mapped to a CoAP Method according to the table 6.1.1.-1

The CoAP request shall be constructed using the selected CoAP method, selected options as described in options.

CoAP message includes the 8-bit Code. In case of a request, the Code field indicates the Request Method. The Code field is limited to indicate all oneM2M request code, so additional information should be carry via the CoAP payload field.

CoAP defines a single set of options that are used in both requests and responses. In case of a request, if CoAP defined options is limited to indicate all oneM2M options, additional information should be carry in the CoAP payload field.

Table 6.1.1-1: oneM2M Operation Mapping

oneM2M Operation	CoAP Method
Create	POST
Retrieve	GET
Update	PUT
Delete	DELETE
Notify	POST

6.1.2 CoAP Request to Request Primitive

The CoAP request shall be mapped to a oneM2M request primitive according to the table 6.1.2-1

Table 6.1.2-1: CoAP Method Mapping

CoAP Method	oneM2M Operation
POST	Create or Notify
GET	Retrieve
PUT	Update
DELETE	Delete

In the case of mapping POST to oneM2M operations, operations are derived from the **op** parameter:

If **op** parameter indicated as "**Create (C)**", the POST shall be mapped to Create;

If **op** parameter indicated as "**Notify (N)**", the POST shall be mapped to Notify.

The oneM2M request shall be constructed using the selected method, selected options as described in options.

As CoAP message Code and Options are limited, sometimes additional information is carried in CoAP payload field. In that case, oneM2M request shall be constructed using that additional information.

6.2 Configuration of Options and Query String

This clause describes which information needs configuring to which CoAP options or query string.

6.2.1 Content Format Negotiation Options

The CoAP Accept option can be used to indicate which Content-Format is acceptable to an Originator. If a Hosting CSE supports the Content-Format specified in Accept option of the request, the Hosting CSE shall respond with that

Content-Format. If the Hosting CSE doesn’t support the Content-Format specified in Accept option of the request, 4.06 "Not Acceptable" MUST be sent as a response, unless another error code takes precedence for this response.

Editor’s note: which content format supported in oneM2M needs to be clarified.

6.2.2 Token

Since Token option is used to match between a request and a response(s), the Token shall have one-to-one mapping with ri parameter (M2M-Request-ID).

6.2.3 URI Options

This clause describes how to configure CoAP Uri-Host, Uri-Port, Uri-Path, and Uri-Query.

When addressing a resource on more than 0 hop, the Registrar or Registree address is used and the host, port, path and query part of the address shall be used as the value for the Uri-host, Uri-Port, Uri-Path and Uri-Query CoAP options and ***to*** parameter is mapped to query string.

When addressing a local resource (i.e., 0 hop), the host, port, path and query part of the ***to*** parameter shall be used as the value for the Uri-host, Uri-Port, Uri-Path and Uri-Query CoAP options.

6.2.4 Query String

da, dr, ec, fc, fr, gid, nm, oet, ort, ret, rqt, rst, rc, ret, rp and ro (see core protocol specification [2]) shall be carried in query string.

6.3 Response Codes Mapping

6.3.1 Response Primitive to CoAP Response

The response primitive shall be correlated to the corresponding request primitive and the following rules shall be followed:

- The CoAP response shall be correlated to the CoAP request corresponding to the request primitive.
- If the response primitive has any resource representation, this shall be transported in the payload of the CoAP response.
- The status code of the response for successful and unsuccessful response shall be set according to the table below.

If the request is send from originator, Table 6.3.1-1 shall be used for mapping as successful cases, and Table 6.3.1-2 shall be used for unsuccessful cases:

Table 6.3.2-1 Successful Cases

Status Code	Status Code of CoAP
STATUS_CREATED	2.01 Created
STATUS_ DELETED	2.02 Deleted
STATUS_CHANGED	2.04 Changed
STATUS_CONTENT	2.05 Content
STATUS_ACCEPTED	ACK

Table 6.3.2-2 Unsuccessful Cases

Status Code	Status Code of CoAP
Location info not authorized	4.01 Unauthorized
Unsupported resource	5.01 Not Implemented
Unsupported attribute	5.01 Not Implemented
Cannot forward, target not reachable	5.05 Proxying Not Supported
Cannot forward, other reason TBD	5.00 Internal Server Error TBD
No privilege	4.01 Unauthorized
Create error - already exists	4.12 Precondition Failed
Create error - missing mandatory parameter	4.02 Bad Option
Does not exist	4.04 Not Found
Update error - unacceptable contents	4.06 Not Acceptable
Create delivery - not able to take on responsibility	4.01 Unauthorized
Create fanoutpoint - group request identifier exists	4.12 Precondition Failed
Retrieve fanoutpoint - group request identifier exists	4.12 Precondition Failed
Update fanoutpoint - group request identifier exists	4.12 Precondition Failed
Delete fanoutpoint - group request identifier exists	4.12 Precondition Failed
Create mgmtObj - memory shortage	4.13 Request Entity Too Large
Cancel execInstance - not cancellable	4.03 Forbidden
Cancel execInstance - already complete	4.00 Bad Request
Delete execInstance - not cancellable	4.03 Forbidden
Delete execInstance - already complete	4.00 Bad Request
Retrieve CSEBase - format error	4.15 Unsupported Content-Format
CMDH rules -non compliant	4.00 Bad Request

Editor’s note: This part is updated based on PRO-2014-0372-Status_Code_ Cleanup more status code will be added when oneM2M statue code defined.

6.3.2 CoAP Response Code to oneM2M Response Code

If the CoAP response code is in the range 2.01 to 2.05, then the response shall be considered as a successful case. Table 6.3.2-1 shall be used for successful cases mapping.

Table 6.3.2-1 Successful Cases

CoAP Response Code		oneM2M Response Code	Note
Success 2.xx	2.01 Created	STATUS_CREATED	
	2.02 Deleted	STATUS_ DELETED	
	2.03 Valid	-	Same as HTTP304 Not Modified
	2.04 Changed	STATUS_CHANGED	
	2.05 Content	STATUS_CONTENT	

If the CoAP response code is in the range of 4.00 to 4.15 or 5.00 to 5.05, then the response shall be considered as unsuccessful. Table 6.3.2-2 shall be used for unsuccessful cases mapping. Additional information about the error should be included.

Table 6.3.2-2 Successful Cases

CoAP Response Code		oneM2M Response Code	Note
Client error	4.00	Cancel execInstance - already complete Delete execInstance - already complete CMDH rules -non compliant	4.00 shall be used for multiple response, depends on parameters in additional information
	4.01	Location info not authorized No privilege Create delivery - not able to take on responsibility	4.01 shall be used for multiple response, depends on parameters in additional information
	4.02	Create error - missing mandatory parameter	
	4.03	Cancel execInstance - not cancellable Delete execInstance - not cancellable	4.03 shall be used for multiple response, depends on parameters in additional information
	4.04	Does not exist	
	4.05		TBD
	4.06	Update error - unacceptable contents	
	4.12	Create error - already exists Create fanoutpoint - group request identifier exists Retrieve fanoutpoint - group request	4.12 shall be used for multiple response, depends on parameters in additional information

		identifier exists Update fanoutpoint - group request identifier exists Delete fanoutpoint - group request identifier exists	
	4.13	Create mgmtObj - memory shortage	
	4.15	Retrieve CSEBase - format error	
Server error	5.00	Cannot forward, other reason TBD	
	5.01	Unsupported resource Unsupported attribute	
	5.02		TBD
	5.03		TBD
	5.04		TBD
	5.05	Cannot forward, target not reachable	

Editor’s note: This part is updated based on PRO-2014-0372-Status_Code_ Cleanup, more status code will be added when oneM2M statue code defined.

6.3.3 Additional Information

CoAP message includes the 8-bit Code. In case of a request, the Code field shall indicate the Request Method; in case of a response, the Code field shall indicate a Response Code.

The Code field is limited to indicate all oneM2M response code, so additional information shall be carried via the CoAP payload field.

6.4 Accessing Resources in CSE

6.4.1 Blocking case

- If rt parameter is configured as “blockingRequest“ (blocking case), the Originator (CoAP client) shall use the Confirmable Method for the resource to the Receiver (CoAP server).
- In case of successful processing of the request at the Receiver, the Receiver shall piggyback the response with an appropriate response code in the Acknowledgment message that acknowledges the Confirmable request.

6.4.2 Non-Blocking Asynchronous case

- If rt parameter is configured as “nonBlockingRequestAsynch“ (non-blocking asynchronous case), the Originator (CoAP client) shall use the Confirmable Method for the resource to the Receiver (CoAP server). Originator shall provide a unique Token value in the request.
- The Receiver shall provide acknowledgment of receipt of the request using Acknowledgment message.
- The Receiver, upon successful processing of the request, shall send an appropriate response in a separate Confirmable message. The Originator shall acknowledge the Confirmable response.

6.4.3 Non-Blocking Synchronous case

- If rt parameter is configured as “nonBlockingRequestSynch“ (non-blocking synchronous case), the Originator (CoAP client) shall use the Confirmable Method for the resource to the Receiver (CoAP server). Originator shall provide a unique Token value in the request.
- The Receiver shall provide an acknowledgment of receipt of the request using Acknowledgment message.
- The Receiver, after validating the request and before processing it fully, shall send an appropriate response including a reference in a separate Confirmable message. The Originator shall acknowledge the Confirmable response. Alternatively, if possible for the Receiver, the response can be piggy-backed with acknowledgment message in the previous step.
- The Originator can use the reference or the token to synchronously access or retrieve the resource. The Receiver, upon receipt of the request, shall respond with the current state of the resource.

Note: If the Receiver is a Transit CSE, the Receiver acts as CoAP client and CoAP server.

7 Security Consideration

CoAP itself does not provide protocol primitives for authentication or authorization; where this is required, it shall be provided by DTLS.

Just as HTTP is secured using Transport Layer Security (TLS) over TCP, CoAP shall be secured using Datagram TLS (DTLS) [5].

All CoAP messages shall be sent as DTLS “application data”. For matching an ACK or RST to a CON message or a RST to a NON message: The DTLS session shall be the same and the epoch shall be the same.

For matching a response to a request, the DTLS session shall be the same and the epoch shall be the same. The response to a DTLS secured request shall always be DTLS secured using the same security session and epoch.

OneM2M primitive parameters contained in CoAP messages may be protected by DTLS as hop-by-hop manner, not end-to-end. For the details, see clause 6.1 [4]

History

Publication history		
V1.1.1	<dd-Mmm-yyyy>	<Milestone>

Draft history (to be removed on publication)		
V.0.1.0	2014-02-05	Initial Draft Version
V.0.2.0	2014-04-10	Incorporates: PRO-2014-0023R04
V.0.3.0	2014-06-18	Incorporates: PRO-2014-0255R01, PRO-2014-0254
V.0.4.0	2014-07-27	Incorporates: PRO-2014-0319, PRO-2014-0318, PRO-2014-0317, 2014-0297R01, 2014-0264R01
V.0.4.1	2014-07-31	Incorporates: PRO-2014-0253R03, PRO-2014-0368R01, PRO-2014-0369R01, PRO-2014-386R03, PRO-2014-0382R01
V.0.5.0	2014-08-01	Incorporates: PRO-2014-0413R02