ONEM2M
TECHNICAL SPECIFICATION

Document Number	TS-0008- V-1.0.1
Document Name:	CoAP Protocol Binding
Date:	2015-January-30
Abstract:	The specification will cover the protocol specific part of communication protocol used by oneM2M compliant systems as 'CoAP binding'

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

Contents

1 Scope

The present document will cover the protocol specific part of communication protocol used by oneM2M compliant systems as 'RESTful CoAP binding'.

The scope of this specification is (not limited to as shown below):

- Binding oneM2M primitives to CoAP messages
- Binding oneM2M Response Status Codes to CoAP Response Codes
- Defining behaviour of a CoAP Client and Server depending on oneM2M parameters

2 References

2.1 Normative 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 reference document (including any amendments) applies.

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: " Service Layer Core Protocol Specification".

[3] IETF draft-ietf-core-block-15: "Blockwise transfers in CoAP".

[4] oneM2M TS-0003: "Security Solutions".

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

2.2 Informative 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 reference document (including any amendments) applies.

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.

NOTE: Available http://member.onem2m.org/Static_pages/Others/Rules_Pages/oneM2M-Drafting-Rules-V1_0.doc.

3 Abbreviations and acronyms

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

DTLS	Datagram Transport Layer Security
HTTP	Hyper Text Transfer Protocol
IP	Internet Protocol
TCP	Transport Control Protocol
TLS	Transport Layer Security
TLV	Tag - Length - Value (data structure)

UDP	User Datagram Protocol
URI	Uniform Resource Identifier

4 Conventions

The keywords "Shall", "Shall not", "May", "Need not", "Should", "Should not" in the present document are to be interpreted as described in the oneM2M Drafting Rules [i.1].

5 Overview

The clause describes which features need to be supported in CoAP layer and introduces a message format and several features of CoAP used in this procotol binding specification.

5.1 Required Features

This clause explicitely specifies the required features of the CoAP layer for oneM2M to propery bind oneM2M primitives into CoAP messages.

- The 4-byte binary CoAP message header is defined in Section 3 of [1].
- Confirmable (CON), Acknowledgement (ACK) and Reset (RST) messages shall be supported. The Reset message is used to send a error message in response to a malformed Confirmable message in CoAP layer.
- GET, PUT, POST and DELETE methods shall be supported. oneM2M primitives map to these methods.
- A subset of Response Code specified in clause 6.2.4 shall be supported for oneM2M ***Response Status Code*** parameter mapping.
- The Uri-Host, Uri-Port, Uri-Path, and Uri-Query shall be supported.
- The Content-Type Option shall be used to indicate the media types of the payload.
- The Token Option may be used.
- Block-wise transfers feature may be supported to carry large payloads.
- Caching feature may be supported.

5.2 Introduction of CoAP

This clause describes a message format, and caching and block-wise transfers features which may be used to map oneM2M primitives to CoAP messages.

5.2.1 Message Format

This clause 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].

5.2.2 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.

5.2.2.1 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.

5.2.2.2 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.

5.2.3 Blockwise Transfers

CoAP Block [6] Options shall be used when CoAP endpoints 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.

6 CoAP Message Mapping

6.1 Introduction

When AE or CSE binds oneM2M primitives to CoAP messages, or binds CoAP messages to oneM2M primitives, it is required that

- AE shall host a CoAP client and should host a CoAP server or
- CSE shall host both a CoAP client and a CoAP server.

Basically single oneM2M request primitive is mapped to single CoAP request message, and single oneM2M response primitive is mapped to single CoAP response message. However, single oneM2M request/response primitive is mapped to multiple CoAP request/response messages respectively when CoAP block-wise transfers feature is used.

Mapping between CoAP message and oneM2M primitive shall be applied in the following cases:

- when the Originator sends a request primitive,
- when the Receiver receives a CoAP message(s),

- when the Receiver sends a response primitive,
- when the Originator receives a CoAP message(s).

The following sub-clauses specify how to map each oneM2M primitive parameter defined in [7] to a corresponding CoAP message field to compose a CoAP request/response message.

6.2 Primitive Mapping to CoAP Message

This clause describes where to map oneM2M parameters in a primitive to header, Option and payload fields in a CoAP message.

6.2.1 Header

This clause specifies how to configure CoAP header information.

- The Version field shall be configured as 1.
- The Type field shall be configured according to clause 6.3. The Reset message is used to send a error message in response to a malformed Confirmable message in CoAP layer.
- In case of a request, the Code field indicates CoAP Method. The oneM2M ***Operation*** parameter shall be mapped to a CoAP Method according to the table 6.2.1-1.
- In case of a response, the Code field indicates CoAP Response Code. The oneM2M ***Response Status Code*** parameter shall be mapped to CoAP Response Code as specified in clause 6.2.4.

The configurations of Token Length and Message ID are left to implementation.

Table 6.2.1-1: oneM2M Operation Parameter Mapping

oneM2M Operation Parameter	CoAP Method
CREATE	POST
RETRIEVE	GET
UPDATE	PUT
DELETE	DELETE
NOTIFY	POST

At the Receiver, CoAP request message with POST method shall be mapped to oneM2M CREATE or NOTIFY ***Operation*** parameter in accordance with the existence of ***Resource Type*** parameter. If ***Resource Type*** parameter exists then value of the ***Operation*** parameter is CREATE and if ***Resource Type*** parameter doesn't exist, the value of ***Operation*** parameter is NOTIFY.

6.2.2 Configuration of Token and Options

This clause describes configuration of Token and Options based on oneM2M parameters.

6.2.2.1 Token

Due to size limitation, Request Identifier is not mapped to Token option. However, Token may be used in CoAP layer to match a CoAP request and response.

6.2.2.2 Content Format Negotiation Options

The CoAP Accept Option may 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" shall be sent as a response, unless another error code takes precedence for this response.

Possible values for Content-Format and Accept options are listed below.

- application/xml
- application/json
- media types specified in clause 6.7 "oneM2M specific MIME media types" of [8].

6.2.2.3 URI Options

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

Host and port part of the address specified in ***pointOfAccess*** attribute of <*remoteCSE*> resource shall be mapped to Uri-Host and Uri-Port respectively.

To parameter shall be mapped to Uri-Path.

Resource Type, responseType element of ***Response Type***, ***Result Persistence***, ***Delivery Aggregation***, ***Result Content***, parameters of ***Filter Criteria***, and ***Discovery Result Type*** parameters shall be carried in Uri-Query Option in a short name form as specified in clause 8.2.2 [9].

6.2.2.4 Definition of New Options

This clause describes new CoAP Options used for binding several oneM2M request/response parameters. The table 6.2.2.4-1 contains definitions of the new CoAP Options and sub-clauses specify oneM2M parameter mapping with the newly defined CoAP Options in the table 6.2.2.4-1.

Table 6.2.2.4-1: Definition of New Options

No	C	U	N	R	Name	Format	Length	Default
256	X				oneM2M-FR	string	0-255	(None)
257	X				oneM2M-RQI	string	0-255	(None)
258					oneM2M-NM	string	0-255	(None)
259					oneM2M-OT	string	15	(None)
260					oneM2M-RQET	string	15	(None)
261					oneM2M-RSET	string	15	(None)
262					oneM2M-OET	string	15	(None)
263					oneM2M-RTURI	string	0-255	(None)
264					oneM2M-EC	uint	1	(None)
265					oneM2M-RSC	uint	2	(None)
266					oneM2M-GID	string	0-255	(None)

Note: table 6.2.4-1 follows the template used in section 5.10 Option Definitions of CoAP specification [1].

6.2.2.4.1 From

The ***From*** parameter shall be mapped to the oneM2M-FR Option.

6.2.2.4.2 Request Identifier

The ***Request Identifier*** parameter shall be mapped to the oneM2M-RQI Option.

6.2.2.4.3 Name

The ***Name*** parameter shall be mapped to the oneM2M-NM Option.

6.2.2.4.4 Originating Timestamp

The ***Originating Timestamp*** parameter shall be mapped to the oneM2M-OT Option.

6.2.2.4.5 Request Expiration Timestamp

The ***Request Expiration Timestamp*** parameter shall be mapped to the oneM2M-RQET Option.

6.2.2.4.6 Result Expiration Timestamp

The ***Request Expiration Timestamp*** parameter shall be mapped to the oneM2M-RSET Option

6.2.2.4.7 Operation Execution Time

The ***Operation Execution Time*** parameter shall be mapped to the oneM2M-OET Option.

6.2.2.4.8 notificationURI of Response Type

The notificationURI element of ***Response Type*** parameter shall be mapped to the oneM2M-RTURI Option.

6.2.2.4.9 Event Category

The ***Event Category*** parameter shall be mapped to the oneM2M-EC Option.

6.2.2.4.10 Response Status Code

The ***Response Status Code*** parameter shall be mapped to the oneM2M-RSC Option

6.2.2.4.11 Group Request Identifier

The ***Group Request Identifier*** parameter shall be mapped to the oneM2M-GID Option.

6.2.3 Payload

Content parameter shall be mapped to CoAP payload. Blockwise transfers mechanism may be used to deliver large size of ***Content*** parameter which is not fit into one CoAP message. Please refer to clause 6.5 for the detail information. If ***Content*** parameter contains URI and resource representation in a response to a create request, URI shall be mapped to Location-Path Option.

6.2.4 Response Codes Mapping

Table 6.2.4-1 defines a mapping between oneM2M ***Response Status Code*** parameter specified in [10] and CoAP Response Code.

In case of where multiple oneM2M ***Response Status Code*** parameters are mapped to single CoAP Status Code, ***Response Status Code*** parameter shall be specified in oneM2M-RSC Option.

Table 6.2.4-1 Mapping between oneM2M Response Status Code and CoAP Response Code

oneM2M Response Status Code	Description	Status Code of CoAP	Description
1000	ACCEPTED	None	Empty Acknowledgement Message shall be used.
2000	OK	2.05	Content
2001	CREATED	2.01	Created
2002	DELETED	2.02	Deleted
2004	CHANGED	2.04	Changed
4000	BAD_REQUEST	4.00	Bad Request
4004	NOT_FOUND	4.04	Not Found
4005	OPERATION_NOT_ALLOWED	4.05	Method Not Allowed
4008	REQUEST_TIMEOUT	4.04	Not Found
4101	SUBSCRIPTION_CREATOR_HAS_NO_PRIVILEGE	4.03	Forbidden
4102	CONTENTS_UNACCEPTABLE	4.00	Bad Request
4103	ACCESS_DENIED	4.03	Forbidden
4104	GROUP_REQUEST_IDENTIFIER_EXISTS	4.00	Bad Request
4105	CONFLICT	4.03	Forbidden
5000	INTERNAL_SERVER_ERROR	5.00	Internal Server Error
5001	NOT_IMPLEMENTED	5.01	Not Implemented
5103	TARGET_NOT_REACHABLE	4.04	Not Found
5105	NO_PRIVILEGE	4.03	Forbidden
5106	ALREADY_EXISTS	4.00	Bad Request
5203	TARGET_NOT_ SUBSCRIBABLE	4.03	Forbidden
5204	SUBSCRIPTION_VERIFICATION_INITIATION_FAILED	5.00	Internal Server Error
5205	SUBSCRIPTION_HOST_HAS_NO_PRIVILEGE	4.03	Forbidden
5206	NON_BLOCKING_REQUEST_NOT_SUPPORTED	5.00	Internal Server Error
6003	EXTENAL_OBJECT_NOT_REACHABLE	4.04	Not Found
6005	EXTENAL_OBJECT_NOT_FOUND	4.04	Not Found
6010	MAX_NUMBERF_OF_MEMBER_EXCEEDED	4.00	Bad Request
6011	MEMBER_TYPE_INCONSISTENT	4.00	Bad Request
6020	MGMT_SESSION_CANNOT_BE_ESTABLISHED	5.00	Internal Server Error
6021	MGMT_SESSION_ESTABLISHMENT _TIMEOUT	5.00	Internal Server Error
6022	INVALID_CMDTYPE	4.00	Bad Request
6023	INVALID_ARGUMENTS	4.00	Bad Request
6024	INSUFFICIENT_ARGUMENTS	4.00	Bad Request
6025	MGMT_CONVERSION_ERROR	5.00	Internal Server Error
6026	MGMT_CANCELATION_FAILURE	5.00	Internal Server Error
6028	ALREADY_COMPLETE	4.00	Bad Request
6029	COMMAND_NOT_CANCELLABLE	4.00	Bad Request

The Receiver decides the ***Response Status Code*** parameter using the combination of CoAP Response Code and oneM2M-RSC Option information.

6.3 Accessing Resources in CSEs

This clause describes the behavior of CoAP layer depending on ***Response Type*** parameter.

6.3.1 Blocking case

- If ***Response Type*** 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.

6.3.2 Non-Blocking Asynchronous case

- If ***Response Type*** 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.

6.3.3 Non-Blocking Synchronous case

- If ***Response Type*** 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.

6.4 Mapping rules of caching

This clause specifies how to enable or disable CoAP caching mechanism and how to use cached information.

If the CoAP end point supports caching mechanism by freshness, the CoAP server shall:

- set the Max-Age Option value to "0" (zero) to disable caching, in order to support complete oneM2M mapping; or
- set the Max-Age option value to another value (such as the default value), in order to use CoAP caching mechanism for constrained environment.

NOTE 1: In the second case, the new request from oneM2M layer can get the stored fresh response from CoAP client, not from CoAP server.

If the CoAP end point supports caching mechanism by validity:

- the CoAP server shall not present Etag in responses to disable caching, in order to support complete oneM2M mapping; or
- the CoAP server shall present Etag in responses, in order to use CoAP caching mechanism for constrained environment.

NOTE 2: In the second case, the new request from oneM2M layer can get the stored fresh response from CoAP server, not from oneM2M layer.

6.5 Usage of Blockwise Transfers

Using Block Options, large oneM2M 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 request primitives and the CoAP Block2 Option shall be used to define the size of the blocks used for oneM2M response responses. Refer [11] for further details.

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 in a hop-by-hop manner. For the details, see oneM2M security solution specification [4].

NOTE: Some provisioning schemes of oneM2M TS-0003 [4] enable the provisioning of end-to-end credentials, but protocols to establish security associations between non-adjacent nodes are not addressed.by oneM2M in the present release.

Annex A (informative): Example Procedures

A.1 AE Registration

The figure A.1-1 illustrates CoAP mapping of AE Registration procedure described in clause 7.3.5.2.1 of TS-0004 [12].

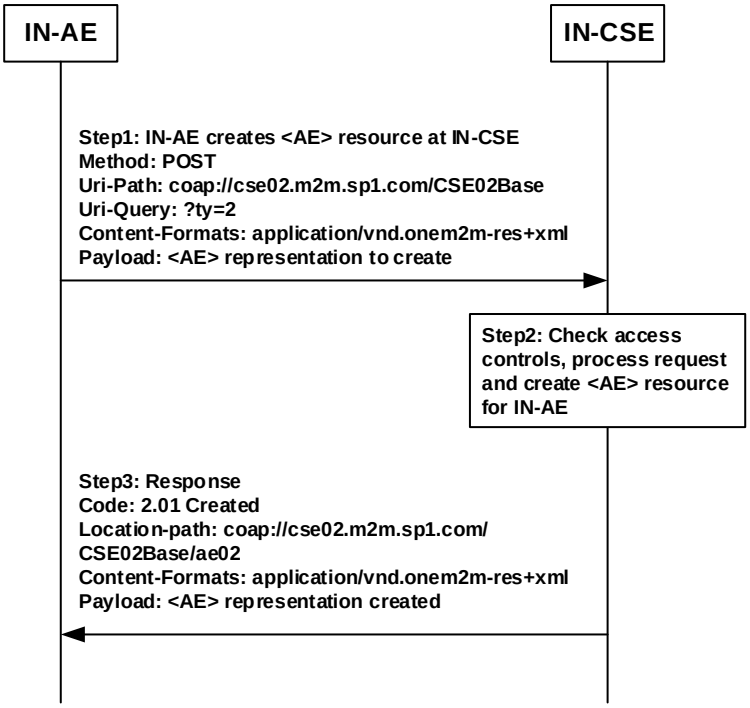

Figure A.1-1 Binding Example – AE Registration

History

Publication history		
V1.0.1	30 Jan 2015	Release 1 - Publication