ETSI TS 103 104 V1.1.1 (2013-04)



Machine-to-Machine communications (M2M); Interoperability Test Specification for CoAP Binding of ETSI M2M Primitives

Reference DTS/M2M-00019

2

Keywords

interoperability, testing

ETSI

650 Route des Lucioles F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C Association à but non lucratif enregistrée à la Sous-Préfecture de Grasse (06) N° 7803/88

Important notice

Individual copies of the present document can be downloaded from: http://www.etsi.org

The present document may be made available in more than one electronic version or in print. In any case of existing or perceived difference in contents between such versions, the reference version is the Portable Document Format (PDF). In case of dispute, the reference shall be the printing on ETSI printers of the PDF version kept on a specific network drive within ETSI Secretariat.

Users of the present document should be aware that the document may be subject to revision or change of status. Information on the current status of this and other ETSI documents is available at http://portal.etsi.org/tb/status/status.asp

If you find errors in the present document, please send your comment to one of the following services: http://portal.etsi.org/chaircor/ETSI_support.asp

Copyright Notification

No part may be reproduced except as authorized by written permission. The copyright and the foregoing restriction extend to reproduction in all media.

> © European Telecommunications Standards Institute 2013. All rights reserved.

DECT[™], **PLUGTESTS[™]**, **UMTS[™]** and the ETSI logo are Trade Marks of ETSI registered for the benefit of its Members. **3GPP**[™] and **LTE**[™] are Trade Marks of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners.

GSM® and the GSM logo are Trade Marks registered and owned by the GSM Association.

Contents

Intelle	ntellectual Property Rights		
Forew	vord	4	
1	Scope	5	
2	References	5	
2.1	Normative references	5	
2.2	Informative references	5	
2	Definitions and althousistics	C	
3 2 1	Definitions	0	
3.2	Abbreviations	6	
4	Conventions	6	
4.1	The Test Description proforma	6	
4.2	Test Description naming convention	7	
5	Test Description Summary	7	
51	CoAP Binding for M2M REST Resources		
5.2	Additional CoAP		
· · -			
6	Basic Configuration		
6.1	Resources offered by servers under test	9	
0.2 6.2	M2M Access Control	13	
0.5 6 /	CoAP settings	13	
0.4	COAL settings		
7	Test Configurations		
7.1	Basic M2M CoAP (M2M_CFG_01)		
7.2	M2M CoAP Multihop (M2M_CFG_02)		
7.3	Basic CoAP I (CoAP_CFG_01)		
7.4 7.5	Test Configuration 3 (CoAP_CFG_02)	13	
1.5			
8	Test Descriptions	16	
8.1	CoAP Binding for M2M REST Resources		
8.1.1	ApplicationCreateRequest	16	
8.1.2	ApplicationKetrieveRequest	10 17	
0.1.5 8 1 <i>1</i>	ApplicationOpulateRequest	17 17	
815	SubscriptionNotifyRequest		
8.1.6	Subscription DeleteRequest		
8.1.7	ApplicationDeleteRequest		
8.1.8	TargetID containing several path segments	20	
8.1.9	TargetID containing several query options	21	
8.1.10	TargetID using partial addressing		
8.1.11	Announcement	23	
8.1.12	Multihop retrieval using Proxy-Uri and aPoC		
8.1.13	Multihop retrieval using m2mPocs		
8.2 8 2 1	Additional CoAP protocol		
0.2.1	CoRF Link Format	20 15	
823	Blockwise transfers	43 50	
8.2.4	Observing Resources		
J			
Anne	x A (informative): Bibliography	62	
Histor	ry	63	

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: "Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards", which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (http://ipr.etsi.org).

4

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Foreword

This Technical Specification (TS) has been produced by ETSI Technical Committee Machine-to-Machine communications (M2M).

1 Scope

The present document specifies Interoperability Test Descriptions (TDs) for the CoAP binding as specified in Annex D of TS 102 921 [5]. The Test Descriptions cover the CoAP protocol specification where relevant. The purpose of the interoperability testing is to prove that end-to-end functionality between devices such as:

- D' and GSCL
- or D' and GSCL and NSCL and NA

and using CoAP as underlying application layer, is as required by the standard(s) on which those devices are based.

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 reference document (including any amendments) applies.

Referenced documents which are not found to be publicly available in the expected location might be found at http://docbox.etsi.org/Reference.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

2.1 Normative references

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

[1]	draft-ietf-core-coap-12: "Constrained Application Protocol (CoAP)".
NOTE:	Available at: <u>http://tools.ietf.org/html/draft-ietf-core-coap-12</u> .
[2]	IETF RFC 6690: "Constrained RESTful Environments (CoRE) Link Forma".
[3]	draft-ietf-core-observe-08: "Observing Resources in CoAP".
NOTE:	Available at: <u>http://tools.ietf.org/html/draft-ietf-core-observe-08</u> .
[4]	draft-ietf-core-block-10: "Blockwise transfers in CoAP".
[5]	ETSI TS 102 921: "Machine-to-Machine communications (M2M); mIa, dIa and mId interfaces".
[6]	ETSI TS 102 690: "Machine-to-Machine communications (M2M); Functional architecture".

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.

Not applicable.

3 Definitions and abbreviations

3.1 Definitions

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

Device' (D'): Hosts DA that communicates to a GSCL using the dIa reference point

application Point of Contract (aPoC): URI that identifies how requests are re-targeted

3.2 Abbreviations

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

ACK	Acknowledgement
aPoC	application Point of Contract
CFG	ConFigGuration
CoAP	Constrained Application Protocol
CON	Confirmable
CORE	Constrained RESTful Environment
D'	Device'
DA	Device Application
dIa	device application Interface
EUT	Equipment Under Test
GSCL	Gateway SCL
mIa	M2M application Interface
mId	M2M device Interface
NA	Network Application
NON	Non-Confirmable
NSCL	Network SCL
OBS	OBServe
RST	Reset
SCL	Service Capability Layer
TD	Test Description
UDP	User Datagram Protocol
URI	Universal Resource Identifier
XML	eXtensible Markup Language

4 Conventions

4.1 The Test Description proforma

The test descriptions are provided in proforma tables. The following different types of test operator actions are considered during the test execution:

- A **stimulus** corresponds to an event that enforces an EUT to proceed with a specific protocol action, like sending a message for instance.
- A **verify** consists of verifying that the EUT behaves according to the expected behaviour (for instance the EUT behaviour shows that it receives the expected message).
- A configure corresponds to an action to modify the EUT configuration.
- A **check** ensures the receipt of protocol messages on reference points, with valid content. This "check" event type corresponds to the method called 'interoperability testing with conformance check'.

4.2 Test Description naming convention

TD/ <root>/<gr>/<nn></nn></gr></root>			1
<root> = root</root>	COAP	Constrained Application Protocol	
	M2M_COAP	CoAP Binding for M2M	
<gr> = group</gr>	CORE	Core protocol	
	LINK	CoRE Link Format	
	BLOCK	Blockwise transfers	
	OBS	Observing Ressources	1
<nn> = sequential number</nn>		01 to 99	7

Table 1: TD naming convention

7

5 Test Description Summary

5.1 CoAP Binding for M2M REST Resources

Table 2: CoAP Binding for M2M REST Resources

-		
1	TD_M2M_COAP_01	M2M DA registers to its local SCL via an applicationCreateRequest (CoAP POST)
2	TD_M2M_COAP_02	M2M DA retrieves application resource via an applicationRetrieveRequest (CoAP GET)
3	TD_M2M_COAP_03	M2M DA updates attribute in application resource via an applicationUpdateRequest (CoAP PUT)
4	TD_M2M_COAP_04	M2M DA creates a subscription to application resource via subscriptionCreateRequest (CoAP POST)
5	TD_M2M_COAP_05	M2M GSCL sends notification(s) via subscriptionNotifyRequest (CoAP POST)
6	TD_M2M_COAP_06	M2M DA cancels subscription via an subscriptionDeleteRequest (CoAP DELETE)
7	TD_M2M_COAP_07	M2M DA de-registers by deleting application resource via an applicationDeleteRequest (CoAP DELETE)
8	TD_M2M_COAP_08	Handle contentInstanceRetrieveRequest with targetID containing several path segments
9	TD_M2M_COAP_09	Handle contentInstanceRetrieveRequest with targetID containing several query options
10	TD_M2M_COAP_10	Handle contentInstanceRetrieveRequest with targetID using partial addressing
11	TD_M2M_COAP_11	M2M DA registration with Announcement
12	TD_M2M_COAP_12	M2M NA multi-hop resource retrieval using Proxy-URI (CoAP proxy)
13	TD_M2M_COAP_13	M2M NA multi-hop resource retrieval using m2mPocs (M2M proxy)

5.2 Additional CoAP

Table 3: CoAP Tests

1	TD COAP CORE 01	Perform GET transaction (CON mode)
2	TD COAP CORE 02	Perform POST transaction (CON mode)
3	TD COAP CORE 03	Perform PUT transaction (CON mode)
4	TD COAP CORE 04	Perform DELETE transaction (CON mode)
5	TD COAP CORE 05	Perform GET transaction (NON mode)
6	TD COAP CORE 06	Perform POST transaction (NON mode)
7	TD COAP CORE 07	Perform PLIT transaction (NON mode)
2 8	TD COAP CORE 08	Perform DELETE transaction (NON mode)
0		Perform GET transaction with delayed response (CON mode, no piggyback)
9	TD_COAP_CORE_09	Perform GET transaction containing Taken option (CON mode)
10	TD_COAP_CORE_10	Perform CET transaction containing token option with a concrete reaponed (CON mode)
11	TD_COAP_CORE_11	Perform GET transaction containing token option with a separate response (CON mode)
12	TD_COAP_CORE_12	Perform GET transaction not containing Token option (CON mode)
13	TD_COAP_CORE_13	Perform GET transaction containing several URI-Path options (CON mode)
14	TD_COAP_CORE_14	Perform GET transaction containing several URI-Query options (CON mode)
15	TD_COAP_CORE_15	Perform GE1 transaction (CON mode, piggybacked response) in a lossy context
16	TD_COAP_CORE_16	Perform GET transaction (CON mode, delayed response) in a lossy context
17	TD_COAP_CORE_17	Perform GET transaction with a separate response (NON mode)
18	TD_COAP_CORE_18	Perform POST transaction with responses containing several Location-Path options (CON mode)
19	TD_COAP_CORE_19	Perform POST transaction with responses containing several Location-Query options (CON mode)
20	TD_COAP_CORE 20	Perform GET transaction containing the Accept option (CON mode)
21	TD_COAP_CORE 21	Perform GET transaction containing the ETag option (CON mode)
22	TD_COAP_CORE_22	Perform GET transaction with responses containing the ETag option and requests
22		Derform DUT transaction with reasonance containing the If Nane Match antion (CON mode)
23	TD_COAP_CORE_23	Perform POT transaction with responses containing the in-None-Match option (CON mode)
24	TD_COAP_CORE_24	(Reverse Proxy in CON mode)
25	TD_COAP_CORE_25	Perform POST transaction with responses containing several Location- Query (Reverse proxy)
26	TD_COAP_CORE_26	Perform GET transaction containing the Accept option (CON mode) (Reverse proxy)
27	TD_COAP_CORE_27	Perform GET transaction with responses containing the ETag option and requests
1		containing the If-Match option (CON mode) (Reverse proxy)
28	TD_COAP_CORE_28	containing the If-Match option (CON mode) (Reverse proxy) Perform GET transaction with responses containing the ETag option and requests
28 11	TD_COAP_CORE_28	containing the If-Match option (CON mode) (Reverse proxy)Perform GET transaction with responses containing the ETag option and requests containing the If-None-Match option (CON mode) (Reverse proxy)Perform GET transaction with responses containing the Max-Age option (Reverse proxy)
28 11 9	TD_COAP_CORE_28	containing the If-Match option (CON mode) (Reverse proxy) Perform GET transaction with responses containing the ETag option and requests containing the If-None-Match option (CON mode) (Reverse proxy) Perform GET transaction with responses containing the Max-Age option (Reverse proxy)
28 11 9 30	TD_COAP_CORE_28 TD_COAP_CORE_29 TD_COAP_LINK_01	containing the If-Match option (CON mode) (Reverse proxy) Perform GET transaction with responses containing the ETag option and requests containing the If-None-Match option (CON mode) (Reverse proxy) Perform GET transaction with responses containing the Max-Age option (Reverse proxy) Access to well-known interface for resource discovery
28 11 9 30 31	TD_COAP_CORE_28 TD_COAP_CORE_29 TD_COAP_LINK_01 TD_COAP_LINK_02	containing the If-Match option (CON mode) (Reverse proxy) Perform GET transaction with responses containing the ETag option and requests containing the If-None-Match option (CON mode) (Reverse proxy) Perform GET transaction with responses containing the Max-Age option (Reverse proxy) Access to well-known interface for resource discovery Use filtered requests for limiting discovery results
28 11 9 30 31 32	TD_COAP_CORE_28 TD_COAP_CORE_29 TD_COAP_LINK_01 TD_COAP_LINK_02 TD_COAP_LINK_03	containing the If-Match option (CON mode) (Reverse proxy) Perform GET transaction with responses containing the ETag option and requests containing the If-None-Match option (CON mode) (Reverse proxy) Perform GET transaction with responses containing the Max-Age option (Reverse proxy) Access to well-known interface for resource discovery Use filtered requests for limiting discovery results Handle empty prefix value strings
28 11 9 30 31 32 33	TD_COAP_CORE_28 TD_COAP_CORE_29 TD_COAP_LINK_01 TD_COAP_LINK_02 TD_COAP_LINK_03 TD_COAP_LINK_04	containing the If-Match option (CON mode) (Reverse proxy) Perform GET transaction with responses containing the ETag option and requests containing the If-None-Match option (CON mode) (Reverse proxy) Perform GET transaction with responses containing the Max-Age option (Reverse proxy) Access to well-known interface for resource discovery Use filtered requests for limiting discovery results Handle empty prefix value strings Filter discovery results in presence of multiple rt attributes
28 11 9 30 31 32 33 34	TD_COAP_CORE_28 TD_COAP_CORE_29 TD_COAP_LINK_01 TD_COAP_LINK_02 TD_COAP_LINK_03 TD_COAP_LINK_04 TD_COAP_LINK_05	containing the If-Match option (CON mode) (Reverse proxy) Perform GET transaction with responses containing the ETag option and requests containing the If-None-Match option (CON mode) (Reverse proxy) Perform GET transaction with responses containing the Max-Age option (Reverse proxy) Access to well-known interface for resource discovery Use filtered requests for limiting discovery results Handle empty prefix value strings Filter discovery results in presence of multiple rt attributes Filter discovery results using if attribute and prefix value strings
28 11 9 30 31 32 33 34 35	TD_COAP_CORE_28 TD_COAP_CORE_29 TD_COAP_LINK_01 TD_COAP_LINK_02 TD_COAP_LINK_03 TD_COAP_LINK_04 TD_COAP_LINK_05 TD_COAP_LINK_06	containing the If-Match option (CON mode) (Reverse proxy) Perform GET transaction with responses containing the ETag option and requests containing the If-None-Match option (CON mode) (Reverse proxy) Perform GET transaction with responses containing the Max-Age option (Reverse proxy) Access to well-known interface for resource discovery Use filtered requests for limiting discovery results Handle empty prefix value strings Filter discovery results in presence of multiple rt attributes Filter discovery results using if attribute and prefix value strings Filter discovery results using sz attribute and prefix value strings
28 11 9 30 31 32 33 34 35 36	TD_COAP_CORE_28 TD_COAP_CORE_29 TD_COAP_LINK_01 TD_COAP_LINK_02 TD_COAP_LINK_03 TD_COAP_LINK_04 TD_COAP_LINK_05 TD_COAP_LINK_06 TD_COAP_LINK_07	containing the If-Match option (CON mode) (Reverse proxy) Perform GET transaction with responses containing the ETag option and requests containing the If-None-Match option (CON mode) (Reverse proxy) Perform GET transaction with responses containing the Max-Age option (Reverse proxy) Access to well-known interface for resource discovery Use filtered requests for limiting discovery results Handle empty prefix value strings Filter discovery results in presence of multiple rt attributes Filter discovery results using if attribute and prefix value strings Filter discovery results using sz attribute and prefix value strings Filter discovery results using href attribute and complete value strings
28 11 9 30 31 32 33 34 35 36 37	TD_COAP_CORE_28 TD_COAP_CORE_29 TD_COAP_LINK_01 TD_COAP_LINK_02 TD_COAP_LINK_03 TD_COAP_LINK_04 TD_COAP_LINK_05 TD_COAP_LINK_06 TD_COAP_LINK_07 TD_COAP_LINK_08	containing the If-Match option (CON mode) (Reverse proxy) Perform GET transaction with responses containing the ETag option and requests containing the If-None-Match option (CON mode) (Reverse proxy) Perform GET transaction with responses containing the Max-Age option (Reverse proxy) Access to well-known interface for resource discovery Use filtered requests for limiting discovery results Handle empty prefix value strings Filter discovery results in presence of multiple rt attributes Filter discovery results using if attribute and prefix value strings Filter discovery results using sz attribute and prefix value strings Filter discovery results using href attribute and complete value strings Filter discovery results using href attribute and prefix value strings
28 11 9 30 31 32 33 34 35 36 37 38	TD_COAP_CORE_28 TD_COAP_CORE_29 TD_COAP_LINK_01 TD_COAP_LINK_02 TD_COAP_LINK_03 TD_COAP_LINK_04 TD_COAP_LINK_05 TD_COAP_LINK_06 TD_COAP_LINK_07 TD_COAP_LINK_08 TD_COAP_LINK_09	containing the If-Match option (CON mode) (Reverse proxy) Perform GET transaction with responses containing the ETag option and requests containing the If-None-Match option (CON mode) (Reverse proxy) Perform GET transaction with responses containing the Max-Age option (Reverse proxy) Access to well-known interface for resource discovery Use filtered requests for limiting discovery results Handle empty prefix value strings Filter discovery results in presence of multiple rt attributes Filter discovery results using if attribute and prefix value strings Filter discovery results using href attribute and complete value strings Filter discovery results using href attribute and prefix value strings Filter discovery results using href attribute and prefix value strings Filter discovery results using href attribute and prefix value strings Filter discovery results using href attribute and prefix value strings Filter discovery results using href attribute and prefix value strings Filter discovery results using href attribute and prefix value strings
28 11 9 30 31 32 33 34 35 36 37 38 39	TD_COAP_CORE_28 TD_COAP_CORE_29 TD_COAP_LINK_01 TD_COAP_LINK_02 TD_COAP_LINK_03 TD_COAP_LINK_04 TD_COAP_LINK_05 TD_COAP_LINK_06 TD_COAP_LINK_07 TD_COAP_LINK_08 TD_COAP_LINK_09	containing the If-Match option (CON mode) (Reverse proxy) Perform GET transaction with responses containing the ETag option and requests containing the If-None-Match option (CON mode) (Reverse proxy) Perform GET transaction with responses containing the Max-Age option (Reverse proxy) Access to well-known interface for resource discovery Use filtered requests for limiting discovery results Handle empty prefix value strings Filter discovery results in presence of multiple rt attributes Filter discovery results using if attribute and prefix value strings Filter discovery results using href attribute and prefix value strings Filter discovery results using href attribute and prefix value strings Filter discovery results using href attribute and prefix value strings Filter discovery results using href attribute and prefix value strings Filter discovery results using href attribute and prefix value strings Filter discovery results using href attribute and prefix value strings Filter discovery results using href attribute and prefix value strings Filter discovery results using href attribute and prefix value strings Filter discovery results using href attribute and prefix value strings Arrange link descriptions hierarchically Handle an alternate link
28 11 9 30 31 32 33 34 35 36 37 38 39 40	TD_COAP_CORE_28 TD_COAP_CORE_29 TD_COAP_LINK_01 TD_COAP_LINK_02 TD_COAP_LINK_03 TD_COAP_LINK_03 TD_COAP_LINK_04 TD_COAP_LINK_05 TD_COAP_LINK_06 TD_COAP_LINK_07 TD_COAP_LINK_08 TD_COAP_LINK_08 TD_COAP_LINK_09 TD_COAP_LINK_10	containing the If-Match option (CON mode) (Reverse proxy) Perform GET transaction with responses containing the ETag option and requests containing the If-None-Match option (CON mode) (Reverse proxy) Perform GET transaction with responses containing the Max-Age option (Reverse proxy) Access to well-known interface for resource discovery Use filtered requests for limiting discovery results Handle empty prefix value strings Filter discovery results in presence of multiple rt attributes Filter discovery results using if attribute and prefix value strings Filter discovery results using sz attribute and prefix value strings Filter discovery results using href attribute and prefix value strings Filter discovery results using href attribute and prefix value strings Filter discovery results using href attribute and prefix value strings Filter discovery results using href attribute and prefix value strings Filter discovery results using href attribute and prefix value strings Filter discovery results using href attribute and prefix value strings Arrange link descriptions hierarchically Handle an alternate link Handle GET blockwise transfer for large resource (early negotiation)
28 11 9 30 31 32 33 34 35 36 37 38 39 40 41	TD_COAP_CORE_28 TD_COAP_CORE_29 TD_COAP_LINK_01 TD_COAP_LINK_02 TD_COAP_LINK_03 TD_COAP_LINK_03 TD_COAP_LINK_04 TD_COAP_LINK_05 TD_COAP_LINK_06 TD_COAP_LINK_06 TD_COAP_LINK_06 TD_COAP_LINK_07 TD_COAP_LINK_08 TD_COAP_LINK_09 TD_COAP_LINK_10 TD_COAP_LINK_02	containing the If-Match option (CON mode) (Reverse proxy) Perform GET transaction with responses containing the ETag option and requests containing the If-None-Match option (CON mode) (Reverse proxy) Perform GET transaction with responses containing the Max-Age option (Reverse proxy) Access to well-known interface for resource discovery Use filtered requests for limiting discovery results Handle empty prefix value strings Filter discovery results in presence of multiple rt attributes Filter discovery results using if attribute and prefix value strings Filter discovery results using sz attribute and prefix value strings Filter discovery results using href attribute and prefix value strings Filter discovery results using href attribute and prefix value strings Filter discovery results using href attribute and prefix value strings Filter discovery results using href attribute and prefix value strings Filter discovery results using href attribute and prefix value strings Filter discovery results using href attribute and prefix value strings Arrange link descriptions hierarchically Handle an alternate link Handle GET blockwise transfer for large resource (early negotiation) Handle GET blockwise transfer for large resource (late negotiation)
28 11 9 30 31 32 33 34 35 36 37 38 39 40 41 42	TD_COAP_CORE_28 TD_COAP_CORE_29 TD_COAP_LINK_01 TD_COAP_LINK_02 TD_COAP_LINK_03 TD_COAP_LINK_03 TD_COAP_LINK_04 TD_COAP_LINK_05 TD_COAP_LINK_06 TD_COAP_LINK_06 TD_COAP_LINK_06 TD_COAP_LINK_07 TD_COAP_LINK_08 TD_COAP_LINK_09 TD_COAP_LINK_10 TD_COAP_LINK_10 TD_COAP_BLOCK_01 TD_COAP_BLOCK_02	containing the If-Match option (CON mode) (Reverse proxy) Perform GET transaction with responses containing the ETag option and requests containing the If-None-Match option (CON mode) (Reverse proxy) Perform GET transaction with responses containing the Max-Age option (Reverse proxy) Access to well-known interface for resource discovery Use filtered requests for limiting discovery results Handle empty prefix value strings Filter discovery results in presence of multiple rt attributes Filter discovery results using if attribute and prefix value strings Filter discovery results using href attribute and prefix value strings Filter discovery results using href attribute and prefix value strings Filter discovery results using href attribute and prefix value strings Filter discovery results using href attribute and prefix value strings Filter discovery results using href attribute and prefix value strings Filter discovery results using href attribute and prefix value strings Filter discovery results using href attribute and prefix value strings Arrange link descriptions hierarchically Handle an alternate link Handle GET blockwise transfer for large resource (early negotiation) Handle GET blockwise transfer for large resource (late negotiation) Handle PUT blockwise transfer for large resource
28 11 9 30 31 32 33 34 35 36 37 38 39 40 41 42 43	TD_COAP_CORE_28 TD_COAP_CORE_29 TD_COAP_LINK_01 TD_COAP_LINK_01 TD_COAP_LINK_02 TD_COAP_LINK_03 TD_COAP_LINK_03 TD_COAP_LINK_03 TD_COAP_LINK_04 TD_COAP_LINK_05 TD_COAP_LINK_06 TD_COAP_LINK_06 TD_COAP_LINK_07 TD_COAP_LINK_08 TD_COAP_LINK_09 TD_COAP_LINK_10 TD_COAP_BLOCK_01 TD_COAP_BLOCK_02 TD_COAP_BLOCK_03 TD_COAP_BLOCK_04	containing the If-Match option (CON mode) (Reverse proxy) Perform GET transaction with responses containing the ETag option and requests containing the If-None-Match option (CON mode) (Reverse proxy) Perform GET transaction with responses containing the Max-Age option (Reverse proxy) Access to well-known interface for resource discovery Use filtered requests for limiting discovery results Handle empty prefix value strings Filter discovery results in presence of multiple rt attributes Filter discovery results using if attribute and prefix value strings Filter discovery results using href attribute and prefix value strings Filter discovery results using href attribute and prefix value strings Filter discovery results using href attribute and prefix value strings Filter discovery results using href attribute and prefix value strings Filter discovery results using href attribute and prefix value strings Filter discovery results using href attribute and prefix value strings Filter discovery results using href attribute and prefix value strings Arrange link descriptions hierarchically Handle an alternate link Handle GET blockwise transfer for large resource (early negotiation) Handle PUT blockwise transfer for large resource Handle POST blockwise transfer for large resource
28 11 9 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44	TD_COAP_CORE_28 TD_COAP_CORE_29 TD_COAP_LINK_01 TD_COAP_LINK_01 TD_COAP_LINK_02 TD_COAP_LINK_03 TD_COAP_LINK_03 TD_COAP_LINK_04 TD_COAP_LINK_05 TD_COAP_LINK_06 TD_COAP_LINK_06 TD_COAP_LINK_07 TD_COAP_LINK_08 TD_COAP_LINK_09 TD_COAP_LINK_10 TD_COAP_BLOCK_01 TD_COAP_BLOCK_02 TD_COAP_BLOCK_03 TD_COAP_BLOCK_04	containing the If-Match option (CON mode) (Reverse proxy)Perform GET transaction with responses containing the ETag option and requests containing the If-None-Match option (CON mode) (Reverse proxy)Perform GET transaction with responses containing the Max-Age option (Reverse proxy)Access to well-known interface for resource discoveryUse filtered requests for limiting discovery resultsHandle empty prefix value stringsFilter discovery results in presence of multiple rt attributesFilter discovery results using if attribute and prefix value stringsFilter discovery results using href attribute and prefix value stringsFilter discovery results using href attribute and prefix value stringsFilter discovery results using href attribute and prefix value stringsFilter discovery results using href attribute and prefix value stringsFilter discovery results using href attribute and prefix value stringsFilter discovery results using href attribute and prefix value stringsFilter discovery results using href attribute and prefix value stringsFilter discovery results using href attribute and prefix value stringsHandle an alternate linkHandle GET blockwise transfer for large resource (early negotiation)Handle GET blockwise transfer for large resourceHandle POST blockwise transfer for large resourceHandle resource observation with CON messages
28 11 9 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45	TD_COAP_CORE_28 TD_COAP_CORE_29 TD_COAP_LINK_01 TD_COAP_LINK_01 TD_COAP_LINK_02 TD_COAP_LINK_03 TD_COAP_LINK_03 TD_COAP_LINK_03 TD_COAP_LINK_04 TD_COAP_LINK_05 TD_COAP_LINK_06 TD_COAP_LINK_06 TD_COAP_LINK_07 TD_COAP_LINK_08 TD_COAP_LINK_09 TD_COAP_LINK_10 TD_COAP_BLOCK_01 TD_COAP_BLOCK_02 TD_COAP_BLOCK_03 TD_COAP_BLOCK_04 TD_COAP_OBS_01 TD_COAP_OBS_02	containing the If-Match option (CON mode) (Reverse proxy)Perform GET transaction with responses containing the ETag option and requests containing the If-None-Match option (CON mode) (Reverse proxy)Perform GET transaction with responses containing the Max-Age option (Reverse proxy)Access to well-known interface for resource discoveryUse filtered requests for limiting discovery resultsHandle empty prefix value stringsFilter discovery results in presence of multiple rt attributesFilter discovery results using if attribute and prefix value stringsFilter discovery results using sz attribute and prefix value stringsFilter discovery results using href attribute and prefix value stringsFilter discovery results using href attribute and prefix value stringsFilter discovery results using href attribute and prefix value stringsFilter discovery results using href attribute and prefix value stringsFilter discovery results using href attribute and prefix value stringsFilter discovery results using href attribute and prefix value stringsHandle an alternate linkHandle GET blockwise transfer for large resource (early negotiation)Handle GET blockwise transfer for large resourceHandle POST blockwise transfer for large resourceHandle resource observation with CON messagesHandle resource observation with CON messages
28 11 9 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46	TD_COAP_CORE_28 TD_COAP_CORE_29 TD_COAP_LINK_01 TD_COAP_LINK_01 TD_COAP_LINK_01 TD_COAP_LINK_02 TD_COAP_LINK_03 TD_COAP_LINK_03 TD_COAP_LINK_04 TD_COAP_LINK_05 TD_COAP_LINK_06 TD_COAP_LINK_06 TD_COAP_LINK_07 TD_COAP_LINK_08 TD_COAP_LINK_09 TD_COAP_LINK_10 TD_COAP_BLOCK_01 TD_COAP_BLOCK_02 TD_COAP_BLOCK_03 TD_COAP_BLOCK_04 TD_COAP_OBS_01 TD_COAP_OBS_02 TD_COAP_OBS_03	containing the If-Match option (CON mode) (Reverse proxy)Perform GET transaction with responses containing the ETag option and requests containing the If-None-Match option (CON mode) (Reverse proxy)Perform GET transaction with responses containing the Max-Age option (Reverse proxy)Access to well-known interface for resource discoveryUse filtered requests for limiting discovery resultsHandle empty prefix value stringsFilter discovery results in presence of multiple rt attributesFilter discovery results using if attribute and prefix value stringsFilter discovery results using sz attribute and prefix value stringsFilter discovery results using href attribute and prefix value stringsFilter discovery results using href attribute and prefix value stringsFilter discovery results using href attribute and prefix value stringsFilter discovery results using href attribute and prefix value stringsFilter discovery results using href attribute and prefix value stringsFilter discovery results using href attribute and prefix value stringsFilter discovery results using href attribute and prefix value stringsHandle an alternate linkHandle GET blockwise transfer for large resource (early negotiation)Handle GET blockwise transfer for large resourceHandle POST blockwise transfer for large resourceHandle resource observation with NON messagesHandle resource observation with NON messages
28 11 9 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47	TD_COAP_CORE_28 TD_COAP_CORE_29 TD_COAP_LINK_01 TD_COAP_LINK_01 TD_COAP_LINK_01 TD_COAP_LINK_02 TD_COAP_LINK_03 TD_COAP_LINK_03 TD_COAP_LINK_03 TD_COAP_LINK_05 TD_COAP_LINK_06 TD_COAP_LINK_06 TD_COAP_LINK_07 TD_COAP_LINK_08 TD_COAP_LINK_09 TD_COAP_LINK_10 TD_COAP_BLOCK_01 TD_COAP_BLOCK_02 TD_COAP_BLOCK_03 TD_COAP_BLOCK_04 TD_COAP_OBS_01 TD_COAP_OBS_03 TD_COAP_OBS_04	containing the If-Match option (CON mode) (Reverse proxy) Perform GET transaction with responses containing the ETag option and requests containing the If-None-Match option (CON mode) (Reverse proxy) Perform GET transaction with responses containing the Max-Age option (Reverse proxy) Access to well-known interface for resource discovery Use filtered requests for limiting discovery results Handle empty prefix value strings Filter discovery results in presence of multiple rt attributes Filter discovery results using if attribute and prefix value strings Filter discovery results using sz attribute and prefix value strings Filter discovery results using href attribute and prefix value strings Filter discovery results using href attribute and prefix value strings Filter discovery results using href attribute and prefix value strings Filter discovery results using href attribute and prefix value strings Filter discovery results using href attribute and prefix value strings Arrange link descriptions hierarchically Handle GET blockwise transfer for large resource (early negotiation) Handle PUT blockwise transfer for large resource Handle POST blockwise transfer for large resource Handle POST blockwise transfer for large resource Handle resource observation with CON messages Stop resource observation <tr< td=""></tr<>
28 11 9 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48	TD_COAP_CORE_28 TD_COAP_CORE_29 TD_COAP_LINK_01 TD_COAP_LINK_01 TD_COAP_LINK_01 TD_COAP_LINK_02 TD_COAP_LINK_03 TD_COAP_LINK_03 TD_COAP_LINK_05 TD_COAP_LINK_06 TD_COAP_LINK_06 TD_COAP_LINK_06 TD_COAP_LINK_07 TD_COAP_LINK_08 TD_COAP_LINK_09 TD_COAP_LINK_01 TD_COAP_LINK_03 TD_COAP_BLOCK_01 TD_COAP_BLOCK_02 TD_COAP_BLOCK_03 TD_COAP_BLOCK_04 TD_COAP_OBS_01 TD_COAP_OBS_03 TD_COAP_OBS_04 TD_COAP_OBS_05	containing the If-Match option (CON mode) (Reverse proxy) Perform GET transaction with responses containing the ETag option and requests containing the If-None-Match option (CON mode) (Reverse proxy) Perform GET transaction with responses containing the Max-Age option (Reverse proxy) Access to well-known interface for resource discovery Use filtered requests for limiting discovery results Handle empty prefix value strings Filter discovery results in presence of multiple rt attributes Filter discovery results using if attribute and prefix value strings Filter discovery results using sz attribute and prefix value strings Filter discovery results using hef attribute and prefix value strings Filter discovery results using hef attribute and prefix value strings Filter discovery results using here attribute and prefix value strings Filter discovery results using here fattribute and prefix value strings Filter discovery results using here fattribute and prefix value strings Filter discovery results using here fattribute and prefix value strings Filter discovery results using for large resource (early negotiation) Handle an alternate link Handle GET blockwise transfer for large resource Handle POST blockwise transfer for large resource Handle resource observation with CON messages Handle resource observation with NON mess
28 111 9 30 31 32 33 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49	TD_COAP_CORE_28 TD_COAP_CORE_29 TD_COAP_LINK_01 TD_COAP_LINK_02 TD_COAP_LINK_03 TD_COAP_LINK_03 TD_COAP_LINK_04 TD_COAP_LINK_05 TD_COAP_LINK_06 TD_COAP_LINK_06 TD_COAP_LINK_06 TD_COAP_LINK_07 TD_COAP_LINK_08 TD_COAP_LINK_07 TD_COAP_LINK_07 TD_COAP_LINK_07 TD_COAP_LINK_07 TD_COAP_LINK_07 TD_COAP_BLOCK_01 TD_COAP_BLOCK_02 TD_COAP_BLOCK_03 TD_COAP_BLOCK_04 TD_COAP_OBS_01 TD_COAP_OBS_02 TD_COAP_OBS_03 TD_COAP_OBS_05 TD_COAP_OBS_06	containing the If-Match option (CON mode) (Reverse proxy) Perform GET transaction with responses containing the ETag option and requests containing the If-None-Match option (CON mode) (Reverse proxy) Perform GET transaction with responses containing the Max-Age option (Reverse proxy) Access to well-known interface for resource discovery Use filtered requests for limiting discovery results Handle empty prefix value strings Filter discovery results in presence of multiple rt attributes Filter discovery results using if attribute and prefix value strings Filter discovery results using href attribute and prefix value strings Filter discovery results using href attribute and prefix value strings Filter discovery results using href attribute and prefix value strings Filter discovery results using href attribute and prefix value strings Filter discovery results using href attribute and prefix value strings Filter discovery results using href attribute and prefix value strings Handle an alternate link Handle GET blockwise transfer for large resource (early negotiation) Handle GET blockwise transfer for large resource Handle POST blockwise transfer for large resource Handle POST blockwise transfer for large resource Handle resource observation with NON messages Stop resource observation (Max-Age)
28 11 9 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50	TD_COAP_CORE_28 TD_COAP_CORE_29 TD_COAP_LINK_01 TD_COAP_LINK_02 TD_COAP_LINK_03 TD_COAP_LINK_03 TD_COAP_LINK_03 TD_COAP_LINK_03 TD_COAP_LINK_03 TD_COAP_LINK_04 TD_COAP_LINK_05 TD_COAP_LINK_06 TD_COAP_LINK_07 TD_COAP_LINK_08 TD_COAP_LINK_08 TD_COAP_LINK_07 TD_COAP_LINK_07 TD_COAP_LINK_07 TD_COAP_LINK_03 TD_COAP_BLOCK_02 TD_COAP_BLOCK_02 TD_COAP_BLOCK_03 TD_COAP_BLOCK_04 TD_COAP_OBS_01 TD_COAP_OBS_02 TD_COAP_OBS_03 TD_COAP_OBS_05 TD_COAP_OBS_06 TD_COAP_OBS_07	containing the If-Match option (CON mode) (Reverse proxy) Perform GET transaction with responses containing the ETag option and requests containing the If-None-Match option (CON mode) (Reverse proxy) Perform GET transaction with responses containing the Max-Age option (Reverse proxy) Access to well-known interface for resource discovery Use filtered requests for limiting discovery results Handle empty prefix value strings Filter discovery results in presence of multiple rt attributes Filter discovery results using if attribute and prefix value strings Filter discovery results using href attribute and prefix value strings Filter discovery results using href attribute and prefix value strings Filter discovery results using href attribute and prefix value strings Filter discovery results using href attribute and prefix value strings Filter discovery results using href attribute and prefix value strings Filter discovery results using href attribute and prefix value strings Handle a alternate link Handle GET blockwise transfer for large resource (early negotiation) Handle GET blockwise transfer for large resource Handle POST blockwise transfer for large resource Handle resource observation with NON messages Stop resource observation Client detection of deregistration (Max-Age) Server d
28 11 9 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 42 43 44 45 46 47 48 49 50 51	TD_COAP_CORE_28 TD_COAP_CORE_29 TD_COAP_LINK_01 TD_COAP_LINK_02 TD_COAP_LINK_03 TD_COAP_LINK_03 TD_COAP_LINK_03 TD_COAP_LINK_03 TD_COAP_LINK_03 TD_COAP_LINK_03 TD_COAP_LINK_03 TD_COAP_LINK_05 TD_COAP_LINK_06 TD_COAP_LINK_07 TD_COAP_LINK_08 TD_COAP_LINK_07 TD_COAP_LINK_07 TD_COAP_LINK_03 TD_COAP_LINK_03 TD_COAP_BLOCK_02 TD_COAP_BLOCK_02 TD_COAP_BLOCK_03 TD_COAP_BLOCK_04 TD_COAP_OBS_01 TD_COAP_OBS_02 TD_COAP_OBS_03 TD_COAP_OBS_05 TD_COAP_OBS_07 TD_COAP_OBS_08	containing the If-Match option (CON mode) (Reverse proxy) Perform GET transaction with responses containing the ETag option and requests containing the If-None-Match option (CON mode) (Reverse proxy) Perform GET transaction with responses containing the Max-Age option (Reverse proxy) Access to well-known interface for resource discovery Use filtered requests for limiting discovery results Handle empty prefix value strings Filter discovery results in presence of multiple rt attributes Filter discovery results using if attribute and prefix value strings Filter discovery results using sz attribute and prefix value strings Filter discovery results using href attribute and prefix value strings Filter discovery results using href attribute and prefix value strings Filter discovery results using href attribute and prefix value strings Filter discovery results using href attribute and prefix value strings Filter discovery results using from large resource (early negotiation) Handle an alternate link Handle GET blockwise transfer for large resource (late negotiation) Handle PUT blockwise transfer for large resource Handle POST blockwise transfer for large resource Handle resource observation with CON messages Handle resource observation (Max-Age) Server detection of deregistration (Max-Age) <t< td=""></t<>
28 11 9 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52	TD_COAP_CORE_28 TD_COAP_CORE_29 TD_COAP_LINK_01 TD_COAP_LINK_02 TD_COAP_LINK_03 TD_COAP_LINK_03 TD_COAP_LINK_03 TD_COAP_LINK_03 TD_COAP_LINK_03 TD_COAP_LINK_05 TD_COAP_LINK_06 TD_COAP_LINK_06 TD_COAP_LINK_07 TD_COAP_LINK_08 TD_COAP_LINK_09 TD_COAP_LINK_08 TD_COAP_LINK_07 TD_COAP_LINK_03 TD_COAP_BLOCK_01 TD_COAP_BLOCK_02 TD_COAP_BLOCK_03 TD_COAP_BLOCK_04 TD_COAP_OBS_01 TD_COAP_OBS_02 TD_COAP_OBS_03 TD_COAP_OBS_05 TD_COAP_OBS_07 TD_COAP_OBS_08 TD_COAP_OBS_08	containing the If-Match option (CON mode) (Reverse proxy) Perform GET transaction with responses containing the ETag option and requests containing the If-None-Match option (CON mode) (Reverse proxy) Perform GET transaction with responses containing the Max-Age option (Reverse proxy) Access to well-known interface for resource discovery Use filtered requests for limiting discovery results Handle empty prefix value strings Filter discovery results in presence of multiple rt attributes Filter discovery results using if attribute and prefix value strings Filter discovery results using href attribute and prefix value strings Filter discovery results using href attribute and prefix value strings Filter discovery results using href attribute and prefix value strings Filter discovery results using href attribute and prefix value strings Arrange link descriptions hierarchically Handle an alternate link Handle GET blockwise transfer for large resource (early negotiation) Handle PUT blockwise transfer for large resource Handle resource observation with CON messages Handle resource observation with NON messages Stop resource observation Client detection of deregistration (Max-Age) Server detection of deregistration (client OFF) Server cleans the observers li

6 Basic Configuration

6.1 Resources offered by servers under test

In order to ease test setup and execution, CoAP servers are requested to support the following resources and primitives:

Subject	Primitive		
Application management	applicationCreateRequest / Response		
	applicationRetrieveRequest / Response		
	applicationUpdateRequest / Response		
	applicationDeleteRequest / Response		
Subscription management	subscriptionCreateRequest / Response		
	subscriptionNotifyRequest / Response		
	subscriptionDeleteRequest / Response		
Content management	contentInstanceCreateRequest / Response		
	containerCreateRequest/Response		
	contentInstanceRetrieveRequest / Response		
Announcement management	applicationAnncCreateRequest / Response		
PoC management	m2mPocCreateRequest / Response		

Table 4: M2M Primitives

M2M Primitive	Resource	Resource Representation
applicationCreateRequest	<app></app>	xml version="1.0"?
		<tns:application< td=""></tns:application<>
		<pre>xmlns:tns="http://uri.etsi.org/m2m"</pre>
		tns:id="app"/>
applicationCreateResponse	<app></app>	xml version="1.0"?
		<tns:application< td=""></tns:application<>
		xmins.ths="http://uri.etsi.org/m2m" ths.id="app">
		<pre><li< td=""></li<></pre>
		<pre>/tng:application></pre>
application	Cappo	<pre> </pre>
application retriever esponse	<upp></upp>	<tns:application< td=""></tns:application<>
		<pre>xmlns:tns="http://uri.etsi.org/m2m" tns:id="app"></pre>
		<tns:applicationstatus>ONLINE</tns:applicationstatus>
		tus>
		<tns:expirationtime>2012-11-</tns:expirationtime>
		19T18:39:05
		<tns:lastmodifiedtime>2012-11-</tns:lastmodifiedtime>
		12T19:59:05
		<tns:containersreference></tns:containersreference>
		/gsclBase/applications/app/containers
		(gsclBase/applications/app/groups
		<tns:accessrightsreference></tns:accessrightsreference>
		/gsclBase/applications/app/accessRights
		<tns:subscriptionsreference>/</tns:subscriptionsreference>
		/gsclBase/applications/app/subscriptions
applicationUpdateRequest	<app></app>	xml version="1.0"?
		<tns:application< td=""></tns:application<>
		xmlns:tns="http://uri.etsi.org/m2m">
		<tns:apoc>coap://DA_IP_Address:Port</tns:apoc>
ann lianting ha data Dang ang a	(
applicationUpdateResponse	<app></app>	<pre></pre>
		<pre>vmlng:tng="bttp://uri_etgi_org/m2m"></pre>
		<pre>ctng:evpirationTime>2012-10-</pre>
		25T13:13:04
		/tns:application>
applicationCreateRequest	<app_ann></app_ann>	xml version="1.0"?
		<tns:application< td=""></tns:application<>
		xmlns:tns="http://uri.etsi.org/m2m"
		tns:id="app_ann">
		<tns:announceto></tns:announceto>
		<tns:activated>true</tns:activated>
		<tns:apoc>coap://DA IP Address:Port</tns:apoc>
		<td< td=""></td<>

Table 5: Examples of M2M Resources Representations

M2M Primitive	Resource	Resource Representation
	name	
applicationCreateResponse	<app_ann></app_ann>	<tns:application< td=""></tns:application<>
		xmlns:tns="http://uri.etsi.org/m2m"
		tns:id="app_ann ">
		<tns:expirationtime>2012-10-</tns:expirationtime>
		25T13:13:04
subscriptionCreateRequest		xml version="1.0"?
		<tns:subscription< td=""></tns:subscription<>
		xmlns:tns="http://uri.etsi.org/m2m" tns:id="sub">
		<tns:contact>coap://DA_IP_Addr:Port/da_notif</tns:contact>
		:contact>
subscriptionCreateResponse		xml version="1.0"?
·····		<tns:subscription< td=""></tns:subscription<>
		xmlns:tns="http://uri.etsi.org/m2m" tns:id="sub
		">
		<tns:expirationtime>2012-10-</tns:expirationtime>
		25T13:13:04
subscriptionNotifvRequest		<pre><?xml version="1.0"?></pre>
eubeenpacin temprequeet		<tns:notify xmlns:tns="http://uri.etsi.org/m2m"></tns:notify>
		<pre><statuscode>1</statuscode></pre>
		<pre><representation></representation></pre>
		base64Binary encoded representation of
		application resource
		<pre>/representation></pre>
		<pre><pre>csubscriptionReference></pre></pre>
		coap://GW_IP_Addr:Port/gw01/applications/app/subs
		criptions/sub
subscriptionNotifyResponse		<pre><?xml version="1.0"?></pre>
Subscription totily response	(Bub)	<pre><tns:notify xmlns:tns="http://uri_etsi_org/m2m"></tns:notify></pre>
		<pre><statuscode>1</statuscode></pre>
containerCreateRequest	<container1></container1>	<pre><?xml version="1.0"?></pre>
containeroreaterrequest	· · · · · · · · · · · · · · · · · · ·	<pre>container</pre>
		xmlns:tns="http://uri_etsi_org/m2m"
		tns:id="container1"/>
containerCreateResponse	<container1></container1>	<pre><?xml version="1 0"?></pre>
containeroreater(coponise	(concarner 1)	<pre><tns:container< pre=""></tns:container<></pre>
		xmlns:tns="http://uri_etsi_org/m2m"
		<pre>tns:id="container1"/></pre>
	<test></test>	<pre><?xml version="1 0"?></pre>
contentinistanceoreaterrequest		<pre><tns:contentinstance< pre=""></tns:contentinstance<></pre>
		xmlns:tns="http://uri_etsi_org/m2m">
		<pre>ctns:content></pre>
		<pre><tns:textcontent>content</tns:textcontent></pre>
contontinatanac Croata Dagager	<tost></tost>	<pre>></pre>
contentinstanceCreateResponse		<pre></pre>
		vmlng:tng="http://uri_etgi_org/m2m"
		thatid="togt"/>
		$lis \cdot ia = les l' / 2$

Resource name	Description	Used in
/test	Default test resource	TD_COAP_CORE_01
		TD_COAP_CORE_02
		TD_COAP_CORE_03
		TD COAP CORE 04
		TD COAP CORE 05
		TD_COAP_CORE_08
		TD_COAP_CORE_10
		TD_COAP_CORE_11
		ID_COAP_CORE_14
		TD_COAP_CORE_18
		TD_COAP_CORE_22
		TD_COAP_LINK_08
		TD_COAP_LINK_10
/validate	Resource which varies	TD COAP CORE 21
		TD_COAP_CORE_27
		TD COAP CORE 29
/greate1	Pasaurea which does not exist yet (to perform atomic	
/ CIEALEI		ID_OOMF_OOKE_23
/create2	Resource which does not exist vet	TD COAP CORF 24
/create3	Resource which does not exist yet	
/seg1/seg2/seg2	Long path recourses	
/ Seg1/ Seg2/ Seg3	Long path resource	TD_COAP_COKE_12
/location1/location2/lo	Location path resource	TD_COAP_CORE_18
cation3		TD_COAP_CORE_24
/location-query	Resource accepting location query parameters	TD_COAP_CORE_19
		TD_COAP_CORE_25
/query	Resource accepting query parameters	TD COAP CORE 13
/separate	Resource which cannot be served immediately and	TD COAP CORE 09
- -	which cannot be acknowledged in a piggy-backed	TD COAP CORE 15
	way	TD COAP CORE 16
/large		
/ laige	Large resource	
		TD_COAP_BLOCK_02
/large-update	Large resource that can be updated using PUT method	TD_COAP_BLOCK_03
/large-create	Large resource that can be created using POST method	TD_COAP_BLOCK_04
/obs	Observable resource which changes every 5 seconds	TD_COAP_OBS_01
	and for which the server is configured to send	TD COAP OBS 03
	confirmable (CON) notifications	TD COAP OBS 04
		TD COAP OBS 05
		TD_COAP_OBS_06
		TD_COAP_OBS_09
/obs-non	Observable resource which changes every 5 seconds	TD_COAP_OBS_02
	and for which the server is configured to send non-	
	confirmable (NON) notifications	
/.well-known/core	Core Link Format	TD_COAP_LINK_01
		TD_COAP_LINK_02
		TD COAP LINK 03
		TD_COAP_LINK_04
		TD COAP LINK 05
		TD COAP LINK 06
	_	TD_COAP_LINK_10
/multi-format	Resource that exists in different content formats	TD_COAP_CORE_20
	(text/plain utf8 and application/xml)	TD_COAP_CORE_26
/link1	Link test resource	TD_COAP_LINK_07
		TD COAP LINK 08
/link2	Link test resource	TD COAP LINK 07

Table 6: Resources offered by CoAP Servers

Note on resource sizes:

- Ressources used in TD_COAP_CORE tests should not exceed 64 bytes
- Large resources used in TD_COAP_BLOCK tests shall not exceed 2 048 bytes
- TD_COAP_LINK tests may require usage of Block options with some implementations

6.2 M2M Access Control

M2M Access control is not being used. Hence there is no primitive attribute 'requestingEntity' being mapped to any CoAP query parameter.

6.3 aPoc Re-Targeting Procedure

When M2M DA registers to its GSCL it can:

- either use the aPoc Re-Targeting mechanism;
- or create and update contentInstance resource on the GSCL.

As a consequence, when the GSCL receives a resource retrieve request, it will:

- either forward the request to DA;
- or reply directly to the request itself.

6.4 CoAP settings

Unless stated otherwise, the following settings shall be applied:

- Each equipment under test shall be configured with a unicast address
- Client cache shall be cleaned up after each test
- Use of ETag option shall be avoided, but implementation should be prepared to handle it
- Use of Token shall be avoided, but implementation should be prepared to handle it
- Use of Piggybacked responses shall be preferred

7 Test Configurations

This clause defines the different test configurations.





7.2 M2M CoAP Multihop (M2M_CFG_02)





7.3Basic CoAP 1 (CoAP_CFG_01)



Figure 3: Basic One-2-One CoAP client/server Configuration

7.4 CoAP in lossy context (CoAP_CFG_02)



15

Figure 4: Basic One-2-One CoAP client/server Configuration in lossy context

The Gateway emulates a lossy medium between the client and the server. It does not implement the CoAP protocol itself (in other terms it is not a CoAP proxy), but works at the transport layer. It provides two features:

- It performs NAT-style UDP port redirections towards the server (thus the client contacts the gateway and is transparently redirected towards the server)
- It randomly drops packets that are forwarded between the client and the server

7.5 Test Configuration 3 (CoAP_CFG_03)



Figure 5: Basic One-2-One CoAP proxy/server Configuration

The reverse proxy shown in the Figure 5 is assumed as CoAP/CoAP proxy. Test operator includes an interface (it can be a CoAP client) that creates the stimulus to initiate the tests for reverse proxy.

More clearly, there exist two methods to create the stimulus for reverse proxy:

- 1) Reverse proxy can provide a direct interface to create and launch the stimulus
- 2) A CoAP client can be connected to reverse proxy to create and launch the stimulus for the tests

In the both cases, reverse proxy and client equally act as point of observation.

8 Test Descriptions

8.1 CoAP Binding for M2M REST Resources

8.1.1 ApplicationCreateRequest

TD_M2M	COAD 01			
TD_M2M_COAP_01				
M2M DA registers to its local SCL via an applicationCreateRequest (CoAP POST)				
M2M_CF0	G_01			
[5], clause	es 10.8.2, Annex D			
	0.0.2.0			
Pre-test void conditions:				
Step	Туре	Description		
1	Stimulus	M2M DA is requested to send a applicationCreateRequest (CoAP POST)		
2	Check (dla)	Sent POST request contains: • Code = 2(POST) • Uri-Path: <sclbase> • Uri-Path: applications • Payload: application resource <app> to be created • Content Type option</app></sclbase>		
3	Check (dla)	 SCL sends response containing: Code = 65(2.01 Created) Location-Path: <sclbase></sclbase> Location-Path: applications Location-Path: <app></app> The same Message ID as that of the previous request 		
	M2M DA 1 M2M_CF([5], clause [6], clause void 1 2 3 3	M2M DA registers to its loca M2M_CFG_01 [5], clauses 10.8.2, Annex D [6], clause 9.3.2.8 void 2 Check (dla) 3 Check (dla) 4 Verify (dla)		

8.1.2 ApplicationRetrieveRequest

Interoperability Test Description			
Identifier:	TD_M2M_COAP_02		
Objective:	M2M DA retrieves application resource via an applicationRetrieveRequest (CoAP GET)		
Configuration:	M2M CFG 01		
References:	[5]. clauses 10.8.3. Annex D		
	[6], claus	e 9.3.2.8	
Pre-test	• DA	has created an A	Application resource <app> on SCL</app>
conditions:			
Test Sequence:	Step	Туре	Description
	1	Stimulus	M2M DA is requested to send a applicationRetrieveRequest (CoAP GET)
	2	Check (dla)	Sent GET request contains
			 Code = 1(GET)
			 Uri-Path: <sclbase></sclbase>
			Uri-Path: applications
			 Uri-Path: <app></app>
			Content Format option
	3	Check (dla)	SCL sends response containing:
			 Code = 69(2.05 Content)
			The same Message ID as that of the previous request
			 Payload: application resource for <app></app>
	4	Verify (dla)	M2M DA indicates successful operation

8.1.3 ApplicationUpdateRequest

		Interoperab	bility Test Description					
Identifier:	TD_M2M	TD M2M COAP 03						
Objective:	M2M DA ((CoAP PL	M2M DA updates attribute in application resource via an applicationUpdateRequest (CoAP PUT)						
Configuration:	M2M_CF0	G_01						
References:	[5], clause [6], clause	[5], clauses 10.8.4, Annex D [6], clause 9.3.2.8						
Pre-test conditions:	• DA I	nas created an App	olication resource <app> on SCL</app>					
Test Sequence:	Step	Type	Description					
	1	Stimulus	M2M DA is requested to send a applicationUpdateRequest (CoAP PUT)					
	2	Check (dla)	Sent PUT request contains Code = 3 (PUT) Uri-Path: <sclbase> Uri-Path: applications Uri-Path: <app> Payload: modified application resource Content Format option</app></sclbase>					
	3	Check (dla)	 SCL sends response containing: Code = 68 (2.04 Changed) The same Message ID as that of the previous request 					
1	4	verity (dla)	M2M DA Indicates successful operation					

8.1.4 SubscriptionCreateRequest

		Interoperab	ility Test Description					
Identifier:	TD_M2M_COAP_04							
Objective:	M2M DA ((CoAP PC	M2M DA creates a subscription to application resource via subscriptionCreateRequest (CoAP POST)						
Configuration:	M2M_CF0	G_01						
References:	[5], clause [6], clause	es 10.25.2, Annex [9.3.2.8.19						
-								
Pre-test conditions:	• DA I	nas created an App	olication resource <app> on SCL</app>					
Test Sequence:	Step	Туре	Description					
	1	Stimulus	M2M DA is requested to send a					
			subscriptionCreateRequest (CoAP POST)					
	2	Check (dla)	Sent POST request contains					
			• Code = 2(POST)					
			 Uri-Path: <sclbase></sclbase> 					
			Uri-Path: applications					
			Uri-Path: <app></app>					
			 Uri-Path: subscriptions 					
			Payload: subscription resource _{to be created}					
			Content Format option					
	3	Check (dla)	SCL sends response containing:					
			 Code = 65(2.01 Created) 					
			 Location-Path: <sclbase></sclbase> 					
			Location-Path: applications					
			Location-Path: <app></app>					
			Location-Path: subscriptions					
			Location-Path:					
			The same Message ID as that of the previous					
			request					
	4	Verify (dla)	M2M DA indicates successful operation					

8.1.5	SubscriptionNotifyRequest
-------	---------------------------

		Interoperab	ility Test Description				
Identifier:	TD_M2M	TD_M2M_COAP_05					
Objective:	M2M GSC	M2M GSCL sends notification(s) via subscriptionNotifyRequest (CoAP POST)					
Configuration:	M2M_CF0	G_01					
References:	[5], clause	es 10.25.7, Annex D					
	[6], clause 9.3.2.8.19						
_							
Pre-test	 DA h 	nas created an App	lication resource <app> on SCL</app>				
conditions:	• DAł	nas created subscri	ption _{to <app> on SCL</app>}				
	-	_					
Test Sequence:	Step	Туре	Description				
	1	Stimulus	M2M DA is requested to send a				
			applicationUpdateRequest (CoAP PUT)				
	2	Check (dla)	Sent PUT request contains				
			• Code = 3 (PUT)				
			Uri-Path: <sclbase></sclbase>				
			Uri-Path: applications				
			Uri-Path: <app></app>				
			 Payload: modified application resource 				
			Content Format option				
	3	Check (dla)	Server sends response containing:				
			 Code = 68 (2.04 Changed) 				
			The same Message ID as that of the previous				
			request				
	4	Verify (dla)	M2M DA indicates successful operation				
	5	Verify (dla)	SCL sends subscriptionNotifyRequest (CoAP POST)				
	6	Check (dla)	Sent POST request contains				
			• Type = 0 (CON)				
			• Code = 2(POST)				
			 Uri-Path: contact attribute of 				
			 Payload: notify structure for <app></app> 				
			Content Format option				
	7	Verify (dla)	M2M DA sends subscriptionNotifyResponse				
	8	Check (dla)	M2M DA sends response containing:				
			• Code = 65(2.01 Created)				
			The same Message ID as that of the previous				
			request				
	9	Verify (dla)	M2M DA indicates updated value for <app></app>				

		Interoperab	bility Test Description					
Identifier:	TD M2M	TD M2M COAP 06						
Objective:	M2M DA	M2M DA cancels subscription via an subscriptionDeleteRequest (CoAP DELETE)						
Configuration:	M2M_CF0	G_01						
References:	[5], clause	es 10.25.5, Annex I	D					
	[6], clause	9.3.2.8.19						
Pre-test		has created an Anr	plication resource canno on SCI					
conditions.		as created all App	intion resource capps on SCI					
	• DAT		iption _{to <app> on SOL</app>}					
Test Sequence:	Step	Type	Description					
•	1	Stimulus	M2M DA is requested to send a					
			subscriptionDeleteRequest (CoAP DELETE)					
	2	Check (dla)	Sent POST request contains					
			• Code = 4 (DELETE)					
			 Uri-Path: <sclbase></sclbase> 					
			Uri-Path: applications					
			Uri-Path: <app></app>					
			 Uri-Path: subscriptions 					
			Uri-Path:					
	3	Check (dla)	SCL sends response containing:					
			 Code = 66(2.02 Deleted) 					
			The same Message ID as that of the previous					
			request					
	4	Verify (dla)	M2M DA indicates successful operation					

8.1.6 SubscriptionDeleteRequest

8.1.7 ApplicationDeleteRequest

		Interoperab	ility Test Description			
Identifier:	TD_M2M_	TD_M2M_COAP_07				
Objective:	M2M DA d	de-registers by dele	eting application resource via an applicationDeleteRequest			
	(CoAP DE	ELETE)				
Configuration:	M2M_CF0	G_01				
References:	[5], clause	es 10.8.5, Annex D				
	[6], clause	9.3.2.8				
Pre-test conditions:	DA has created an Application resource <app> on SCL</app>					
Test Sequence:	Step	Туре	Description			
	1	Stimulus	M2M DA is requested to send a			
			applicationDeleteRequest (CoAP DELETE)			
	2	Check (dla)	Sent POST request contains			
			• Code = 4 (DELETE)			
			 Uri-Path: <sclbase></sclbase> 			
			Uri-Path: applications			
			Uri-Path: <app></app>			
	3	Check (dla)	Server sends response containing:			
			• Code = 66 (2.02 Deleted)			
			The same Message ID as that of the previous			
			request			
	4	Verify (dla)	M2M DA indicates successful operation			

8.1.8 TargetID containing several path segments

		Interoperal	bility Test Description				
Identifier:	TD_M2M_	TD_M2M_COAP_08					
Objective:	Handle co	Handle contentInstanceRetrieveRequest with targetID containing several path					
Configuration:	M2M CFC	M2M CEG 01					
References:	[5]. clause	es 10.19.3. Annex	D				
	[6], clause	9.3.2.15					
Pre-test	• DAł	nas created an Ap	plication resource <app> on SCL</app>				
conditions:	• DAł	nas created contai	ner <container1> on SCL via containerCreateRequest</container1>				
	• DAł	nas created contai	ner <container2> on SCL via containerCreateRequest</container2>				
	• DAł	has created a reso	OUICE / <container1>/<contentinstances>/<test></test></contentinstances></container1>				
	on S	CL via contentIns	tanceCreateRequest				
		_					
Test Sequence:	Step	Туре	Description				
	1	Stimulus	M2M DA is requested to send a				
			contentInstanceRetrieveRequest (CoAP GET) on				
			resource				
	2	Check (dla)	Sent GET request contains				
	2	Oneok (dia)	Code = 1(GET)				
			 Uri-Path: <sclbase></sclbase> 				
			Uri-Path: applications				
			• Uri-Path: <app></app>				
			• Uri-Path: <container1></container1>				
			Uri-Path: <container2></container2>				
			Uri-Path: <test></test>				
			Content Format option				
	3	Check (dla)	SCL sends response containing:				
			 Code = 69(2.05 Content) 				
			 The same Message ID as that of the previous 				
			request				
	4	Verify (dla)	M2M DA indicates successful operation				

8.1.9	TargetID	containing severa	l query options
-------	----------	-------------------	-----------------

		Interoperabi	lity Test Description		
Identifier:	TD_M2M_COAP_09				
Objective:	Handle co	ntentInstanceRetrie	eveRequest with targetID containing several query options		
Configuration:	M2M_CFC	G_01			
References:	[5], clause	es 10.19.3, Annex D			
	[6], clause	9.3.2.15			
Pre-test	DA ł	nas created an App	lication resource <app> on SCL</app>		
conditions:	DA ł	nas created a collect	tion of resources <collec> with filter criteria (criteria1,</collec>		
	crite	ria2) on SCL using	contentInstancesCreateRequest		
	• DAł	has created several	resources in this collection via		
	cont	entInstantCreateRe	equest		
Test Sequence:	Step	l ype	Description		
	1	Stimulus	M2M DA is requested to send a		
			contentInstanceRetrieveRequest (CoAP GET) on		
			resource <collec> with filter criteria (criteria1, criteria2)</collec>		
	2 Check (dla) Sent GET request contains		Sent GET request contains		
			 Code = 1(GET) 		
			 Uri-Path: <sclbase></sclbase> 		
			Uri-Path: applications		
			Uri-Path: <app></app>		
			Uri-Path: <collec></collec>		
			 Uri-Query: criteria1, value1 		
			 Uri-Query: criteria2, value2 		
			Content Format option		
	3	Check (dla)	SCL sends response containing:		
			 Code = 69(2.05 Content) 		
			The same Message ID as that of the previous		
			request		
	4	Verify (dla)	M2M DA indicates successful operation		

		Interoperal	pility Test Description		
Identifier:	TD_M2M_COAP_10				
Objective:	Handle co	ntentInstanceRetr	ieveRequest with targetID using partial addressing		
Configuration:	M2M_CF0	G_01			
References:	[5], clause	es 10.19.3, Annex	D		
	[6], claus	e 9.3.2.15			
Pre-test	• DA ł	nas created an Ap	plication resource <app> on SCL</app>		
conditions:	• DA ł	nas created contai	ner <container1> on SCL via containerCreateRequest</container1>		
	• DA ł	nas created a XML	resource / <container1>/<xml-partial> containing</xml-partial></container1>		
	<pa< th=""><th>rtial> attribute on</th><th>SCL via contentInstanceCreateRequest</th></pa<>	rtial> attribute on	SCL via contentInstanceCreateRequest		
Test Sequence:	Step	Туре	Description		
	1	Stimulus	M2M DA is requested to send a		
			contentInstanceRetrieveRequest (CoAP GET) on		
			attribute <partial> in resource <xml-partial></xml-partial></partial>		
	2	Check (dla)	Sent GET request contains		
			 Code = 1(GET) 		
			 Uri-Path: <sclbase></sclbase> 		
			Uri-Path: applications		
			 Uri-Path: <app></app> 		
			 Uri-Path: <container1></container1> 		
			 Uri-Path: <xml-partial></xml-partial> 		
			 Uri-Path: <partial></partial> 		
			Content Format option		
	3	Check (dla)	SCL sends response containing:		
			 Code = 69(2.05 Content) 		
			 The same Message ID as that of the previous 		
			request		
	4	Verify (dla)	M2M DA indicates successful operation		

8.1.10 TargetID using partial addressing

		Interoperat	pility Test Description
Identifier:	TD_M2M	_COAP_11	
Objective:	M2M DA	registration with Ar	nnouncement
Configuration:	M2M_CF	G_02	
References:	[5], clause	es 10.9.2, Annex D	
	[6], clause	e 9.3.2.28	
Pre-test	• GSC	CL has registered t	o NSCL as <gscl></gscl>
conditions:			
		_	
Test Sequence:	Step	Туре	Description
	1	Stimulus	M2M DA is requested to send a
			application Create Request (COAP POST) with
			Announce to option activated
	2	Check (dia)	Sent POST request contains
			• Code = 2(POST)
			On-Path. applications Device during the particular terms of ter
			 Payload. application resource <app_ann> to be erected</app_ann>
			Content Format antion
	2	Chook (dlo)	Content Format option CSCL conde response containing:
	3	Check (ula)	GOCL serius response containing.
			 Code = 05(2.01 Created) Location Path: creatBases
			Location Path: applications
			 Location Path: capp. app.
			 Location-Fath. <app_atin></app_atin> The same Message ID as that of the providus
			 The same message id as that of the previous request
	4	Verify (dla)	M2M DA indicates successful operation
	5	Verify (mld)	M2M GSCL sends applicationAppcCreateRequest (CoAP
	Ũ	verny (inita)	POST) to M2M NSCL
	6	Check (mld)	Sent POST request contains
	-		• Code = 2(POST)
			Uri-Path: scls
			• Uri-Path: <gscl></gscl>
			Uri-Path: applications
			• Uri-Path: <app ann="">Annc</app>
			Pavload: applicationAnnc resource
			<app_ann>Annc to be created</app_ann>
			Content Format option
	7	Check (mld)	NSCL sends response containing:
			 Code = 65(2.01 Created)
			Location-Path: <nsclbase></nsclbase>
			Location-Path: scls
			Location-Path: <gscl></gscl>
			Location-Path: applications
			 Location-Path: <app_ann>Annc</app_ann>
			The same Message ID as that of the previous
			request
1	8	Verify (mld)	NSCL indicates announced resource <app_ann>Annc</app_ann>

8.1.12	Multihop	retrieval	using	Proxy	y-Uri	and aPoC	2
--------	----------	-----------	-------	-------	-------	----------	---

		Interoperab	ility Test Description			
Identifier:	TD_M2M_COAP_12					
Objective:	M2M NA multi-hop resource retrieval using Proxy-URI (CoAP proxy)					
Configuration:	M2M CFG 02					
References:	[5], clause	es 10.19.3, Annex [D 1.5			
	[6], clause	[6], clause 9.3.2.15				
Pre-test	DA has created an announceable Application resource <app_ann> on GSCL</app_ann>					
conditions:	GSCL has announced <app_ann> to NSCL</app_ann>					
	• DA 0	offers the resource	/test			
	• NA ł	has discovered the	resource /test offered by DA			
Test Sequence:	Step	Туре	Description			
	1	Stimulus	M2M NA is requested to send a			
			contentInstanceRetrieveRequest (CoAP GET) to NSCL			
			for resource /test on DA			
	2	Check (mla)	Sent GET request contains			
			• Code = 1(GET)			
			Proxy-Uri:			
			coap:// <gsclbase>/applications/<app_ann>/test</app_ann></gsclbase>			
			Content Format option			
	3	Verify (mld)	NSCL proxies the request to GSCL			
	4	Check (mld)	Proxied GET request contains			
			• Code = 1(GET)			
			Uri-Path: applications			
			• Uri-Path: <app_ann></app_ann>			
			• Uri-Path: test			
			Content Format option			
	5	Check (mid)	GSCL sends response containing:			
			• $Code = 69(2.05 \text{ Content})$			
			 The same message iD as that of the previous request 			
			Poyload: content of resource /teat			
	6	Varify (mla)	Fayload. content of resource / cest			
	7	Check (mla)	Provied response contains:			
			$\bullet Code = 69(2.05 \text{ Content})$			
			The same Message ID as that of the previous			
			request			
			Payload: content of resource /test			
	0	Verify (mla)	M2M NA indicates successful operation			

	Interoperability Test Description						
Identifier:	TD_M2M_COAP_12						
Objective:	M2M NA multi-hop resource retrieval using m2mPocs (M2M proxy)						
Configuration:	M2M_CFG_02						
References:	[5], clause	[5], clauses 10.19.3, Annex D					
	[6], clause	[6], clause 7.3, 9.2.1.9, 9.2.3.4, 9.2.3.24, 9.2.3.25, 9.3.2.21					
Pre-test	• DAł	nas created a App	lication resource <app> on GSCL</app>				
conditions:	DA offers the resource <test></test>						
	• GSC	L has created an	m2mPoc <test_poc> on NSCL for resource <test></test></test_poc>				
Test Sequence:	Step	Туре	Description				
	1	Stimulus	M2M NA is requested to send a				
			contentInstanceRetrieveRequest (CoAP GET) to NSCL				
			for resource <test></test>				
	2	Check (mla)	Sent GET request contains				
			• Code = 1(GET)				
			• Uri-Path: <gscl></gscl>				
			Uri-Path: applications				
			• Uri-Path: <app></app>				
			• Uri-Path: <container1></container1>				
			Uri-Path: <test></test>				
			Content Format ontion				
	3	Verify (mld)	NSCL provies the request to GSCL using m2mpoc				
	5	veniy (inid)	information				
	4	Check (mld)	Provied GET request contains				
			• Code $= 1(GET)$				
			 Uri-Path: <asci></asci> 				
			Uri-Path: applications				
			Uri-Path: <ann></ann>				
			 Uri-Path: <container1></container1> 				
			Uri Path: <tost> Contant Format option</tost>				
	5	Chock (mld)	On-Fain. <lest>Content Format option</lest>				
	5	Check (mu)	Code = Code = CO(2.05 Content)				
			 Code = 09(2.05 Content) The same Message ID as that of the provious 				
			The same message iD as that of the previous request				
			Payload: content of recourse steats				
	6	Varify (mla)	I ayload: content of resource <cest></cest>				
	0		Provid response containe:				
		спеск (тпа)	Code 60(2.05 Content)				
			• $CODE = 69(2.05 \text{ CONTENT})$				
			Ine same Message ID as that of the previous				
			request				
			Payload: content of resource <test></test>				
	8	Verify (mla)	M2M NA indicates successful operation				

8.1.13 Multihop retrieval using m2mPocs

8.2 Additional CoAP

8.2.1 CoAP protocol

		Interoperal	bility Test Description		
Identifier:	TD_COAP	TD_COAP_CORE_01			
Objective:	Perform G	ET transaction (C	CON mode)		
Configuration:	CoAP_CF	G_01			
References:	[1], clauses	s 5.8.1,1.2,2.1,2.	2,3.1		
Pre-test	 Serve 	Server offers the resource /test with resource content is not empty that handles			
conditions:	GET	with an arbitrary	payload		
Test Sequence:	Step	Туре	Description		
	1	Stimulus	Client is requested to send a GET request with:		
			 Type = 0(CON) 		
			 Code = 1(GET) 		
	2	Check	The request sent by the client contains:		
			 Type=0 and Code=1 		
	3	Check	Server sends response containing:		
			 Code = 69(2.05 Content) 		
			 The same Message ID as that of the request sent 		
			by the client		
			Content format option		
	4	Verify	Client displays the received information		

		Interop	erability Test Description		
Identifier:	TD_COAF	TD_COAP_CORE_02			
Objective:	Perform D	DELETE trans	action (CON mode)		
Configuration:	CoAP_CF	G_01			
References:	[1], clause	es 5.8.4, 1.2, 2	2.1, 2.2, 3.1		
Pre-test conditions:	• \$	Server offers a /test resource that handles DELETE			
Test Sequence:	Step	Туре	Description		
	1	Stimulus	Client is requested to send a DELETE request with:		
			• Type = 0(CON)		
			• Code = 4(DELETE)		
	2	Check	The request sent by the client contains:		
			 Type=0 and Code=4 		
	3	Check	Server sends response containing:		
			 Code = 66(2.02 Deleted) 		
			The same Message ID as that of the request sent by		
			the client		
	4	Verify	Client displays the received information		

		Interop	erability Test Description		
Identifier:	TD_COAF	TD_COAP_CORE_03			
Objective:	Perform P	UT transactio	on (CON mode)		
Configuration:	CoAP_CF	G_01			
References:	[1], clause	es 5.8.3, 1.2, 2	2.1, 2.2, 3.1		
Pre-test	 Serv 	er offers alrea	ady available resource / test or accepts creation of new		
conditions:	reso	urce on /test	that handles PUT		
Test Sequence:	Step	Туре	Description		
	1	Stimulus	Client is requested to send a PUT request with:		
			• Type = 0(CON)		
			• Code = 3(PUT)		
			An arbitrary payload		
			Content format option		
	2	Check	The request sent by the client contains:		
			 Type=0 and Code=3 		
	3	Verify	Server displays received information		
	4	Check	Server sends response containing:		
			 Code = 68 (2.04 Changed) or 65 (2.01 Created) 		
			The same Message ID as that of the request sent by		
			the client		
	5	Verify	Client displays the received response		

		Interop	erability Test Description		
Identifier:	TD_COAF	TD_COAP_CORE_04			
Objective:	Perform P	OST transact	ion (CON mode)		
Configuration:	CoAP_CF	G_01			
References:	[1], clause	s 5.8.2, 1.2, 2	2.1, 2.2, 3.1		
	•				
Pre-test conditions:	Server accepts creation of new resource on / test (resource does not exist yet)				
Test Sequence:	Step	Туре	Description		
	1	Stimulus	Client is requested to send a POST request with:		
			 Type = 0(CON) 		
			 Code = 2(POST) 		
			 An arbitrary payload 		
			Content format option		
	2	Check	The request sent by the client contains:		
			 Type=0 and Code=2 		
	3	Verify	Server displays received information		
	4	Check	Server sends response containing:		
			 Code = 65(2.01 Created) or 68 (2.04 changed) 		
			 The same Message ID as that of the request sent by 		
			the client		
	5	Verify	Client displays the received response		

		Interop	erability Test Description	
Identifier:	TD_COAP_CORE_05			
Objective:	Perform G	ET transactio	n (NON mode)	
Configuration:	CoAP_CF	G_01		
References:	[1], clause	s 5.8.1, 5.2.3		
Pre-test	Server offers a /test resource with resource content is not empty that handles			
conditions:	GET			
Test Sequence:	Step	Туре	Description	
	1	Stimulus	Client is requested to send a GET request with:	
			 Type = 1(NON) 	
			• Code = 1(GET)	
	2	Check	The request sent by the client contains:	
			Type=1 and Code=1	
	3	Check	Server sends response containing:	
			 Type = 1(NON) 	
			 Code= 69(2.05 Content) 	
			Content format option	
	4	Verify	Client displays the received information	

		Interop	erability Test Description		
Identifier:	TD_COAF	TD_COAP_CORE_06			
Objective:	Perform D	ELETE transa	action (NON mode)		
Configuration:	CoAP_CF	G_01			
References:	[1], clause	s 5.8.4, 5.2.3			
Pre-test conditions:	Server offers a /test resource that handles DELETE				
Test Sequence:	Step	Туре	Description		
	1	Stimulus	Client is requested to send a DELETE request with:		
			 Type = 1(NON) 		
			• Code = 4(DELETE)		
	2	Check	The request sent by the client contains:		
			Type=1 and Code=4		
	3	Check	Server sends response containing:		
			 Type = 1(NON) 		
			 Code = 66(2.02 Deleted) 		
	4	Verify	Client displays the received information		

		Interop	erability Test Description		
Identifier:	TD_COAF	TD_COAP_CORE_07			
Objective:	Perform P	UT transactio	n (NON mode)		
Configuration:	CoAP_CF	G_01			
References:	[1], 5.8.3,	5.2.3			
Pre-test	Serve	er offers a /te	st resource that handles PUT		
conditions:					
		-			
Test Sequence:	Step	Туре	Description		
	1	Stimulus	Client is requested to send a PUT request with:		
			 Type = 1(NON) 		
			• Code = 3(PUT)		
			An arbitrary payload		
			Content format option		
	2	Check	The request sent by the client contains:		
			Type=1 and Code=3		
	3	Verify	Server displays the received information		
	4	Check	Server sends response containing:		
			• Type = 1(NON)		
			 Code = 68 (2.04 Changed) or 65 (2.01 Created) 		
	5	Verify	Client displays the received response		

		Interope	erability Test Description	
Identifier:	TD_COAP_CORE_08			
Objective:	Perform P	OST transacti	on (NON mode)	
Configuration:	CoAP_CF	G_01		
References:	[1], clause	s 5.8.2, 5.2.3		
Pre-test conditions:	 Serve 	r accepts crea	ation of new resource on /test (resource does not exist yet)	
Test Sequence:	Step	Туре	Description	
	1	Stimulus	Client is requested to send a POST request with:	
			 Type = 1(NON) 	
			• Code = 2(POST)	
			An arbitrary payload	
			Content format option	
	2	Check	The request sent by the client contains:	
			 Type=1 and Code=2 	
	3	Verify	Server displays the received information	
	4	Check	Server sends response containing:	
			 Type = 1(NON) 	
			 Code = 65(2.01 Created) or 68 (2.04 changed) 	
	5	Verify	Client displays the received response	

		Interop	erability Test Description		
Identifier:	TD_COAF	TD COAP CORE 09			
Objective:	Perform G	ET transactio	on with separate response (CON mode, no piggyback)		
Configuration:	CoAP_CF	G_01			
References:	[1], clause	s 5.8.1, 5.2.2			
Pre-test	 Serv 	er offers a res	source /separate which cannot be served immediately and		
conditions:	whic	h cannot be a	cknowledged in a piggybacked way.		
Test Sequence:	Step	Туре	Description		
	1	Stimulus	Client is requested to send a confirmable GET request to		
			server's resource		
	2	Check	The request sent by the client contains:		
			• Type = 0 (CON)		
			• Code = 1 (GET)		
			Client generated Message ID		
	3	Check	Server sends response containing:		
			• Type = 2 (ACK)		
			• Code = 0		
			 Same message ID as in the request cont by the client 		

			• Code = 0
			Same message ID as in the request sent by the client
			empty Payload
	4	Check	Server sends response containing:
			• Type = 0 (CON)
			 Code = 69 (2.05 content)
			 Server generated Message ID
			Not empty Payload
			Content format option
	5	Check	Client sends response containing:
			• Type = 2 (ACK)
			• Code = 0
			Same message ID as in the response sent by the
			server in step 4
			empty Payload
	6	Verify	Client displays the response
NOTE: Steps 3	and 4 may	occur out-of-	order.

		Interop	perability Test Description		
Identifier:	TD_COA	TD_COAP_CORE_10			
Objective:	Perform 0	GET transaction	on containing Token option (CON mode)		
Configuration:	CoAP_CF	G_01			
References:	[1], clause	[1], clauses 2.2, 5.8.1, 5.10.1			
Pre-test conditions:	 Serv GET 	Server offers a /test resource with resource content is not empty that handles GET			
Test Sequence:	Step	Туре	Description		
	1	Stimulus	Client is requested to send a GET request to server's resource including Token option		
	2	Check	The request sent by the client contains:		

L	Oncork	 Type = 0 (CON) Code = 1 (GET) Option Type = Token Token value = a value generated by the client Length of the token should be between 1 to 8 B
3	Check	 Server sends response containing: Code = 69 (2.05 content) Length of the token should be between 1 to 8 B Token = the same value as in the request sent by the client Not empty Payload Content format option
4	Verify	Client displays the response

		Interop	erability Test Description		
Identifier:	TD_COAF	TD_COAP_CORE_11			
Objective:	Perform G	Perform GET transaction containing token option with a separate response (CON			
	mode)				
Configuration:	CoAP_CF	G_01			
References:	[1], clause	s 2.2, 5.2.2, 5	5.8.1		
Pre-test	• Serv	er offers a res	source /separate which cannot be served immediately.		
conditions.					
Test Sequence:	Step	Type	Description		
	1	Stimulus	Client is requested to send a GET request to server's		
	-		resource including Token option		
	2	Check	The request sent by the client contains:		
			• Type = 0 (CON)		
			• Code = 1 (GET)		
			Option Type = Token		
			 Token value = a value generated by the client 		
			 Length of the token should be between 1 to 8 B 		
	3	Check	Server sends acknowledgement containing:		
			• Type = 2 (ACK)		
			 Code = 0 (Empty) 		
			 same Message-Id as in step 2 		
			empty Payload		
	4	Check	Server sends response containing:		
			• Type = 0 (CON)		
			 Code = 69 (2.05 content) 		
			 Length of the token should be between 1 to 8 B 		
			 Token value = the same value as in the request sent 		
			by the client in step 2		
			Not empty Payload		
	5	Check	Client sends acknowledgement containing:		
			• Type = 2 (ACK)		
			• $Code = 0$ (Empty)		
			same Message-Id as in step 4		
	-		empty Payload		
	6	Verity	Client displays the response		

		Interope	erability Test Description		
Identifier:	TD_COAP	TD_COAP_CORE_12			
Objective:	Perform G	ET transaction	n not containing Token option (CON mode)		
Configuration:	CoAP_CF	G_01			
References:	[1], clause	s 2.2, .8.1, 5.1	0.1		
_	ſ				
Pre-test	 Serve 	er offers a /tes	at resource with resource content is not empty that handles		
conditions:	GET				
Test Sequence:	Step	Туре	Description		
	1	Stimulus	Client is requested to send a confirmable GET request not		
			containing Token option to server's resource		
	2	Check	The request sent by the client contains:		
			 Type = 0 (CON) 		
			 Code = 1 (GET) 		
			No Token option		
	3	Check	Server sends response containing:		
			 Code = 69 (2.05 content) 		
			No Token option		
			Not empty Payload		
			Content format option		
	4	Verify	Client displays the response		

		Interop	erability Test Description		
Identifier:	TD_COAF	TD_COAP_CORE_13			
Objective:	Perform G	ET transactio	n containing several URI-Path options (CON mode)		
Configuration:	CoAP_CF	G_01			
References:	[1], clause	s 5.4.5, 5.10.	2, 6.5		
Pre-test	 Serve 	Server offers a /seg1/seg2/seg3 resource with resource content is not empty			
conditions.					
Test Sequence:	Step	Туре	Description		
	1	Stimulus	Client is requested to send a confirmable GET request to server's resource		
	2	Check	 The request sent by the client contains: Type = 0 (CON) Code = 1 (GET) Option type = URI-Path (one for each path segment), not containing '/ symbol 		
	3	Check	Server sends response containing: • Code = 69 (2.05 content) • Not empty Payload • Content format option		
	4	verify	Ulient displays the response		

		Interop	erability Test Description			
Identifier:	TD_COAF	TD_COAP_CORE_14				
Objective:	Perform G	Perform GET transaction containing several URI-Query options (CON mode)				
Configuration:	CoAP_CF	G_01				
References:	[1], clause	s 5.4.5, 5.10.	2, 6.5			
Pre-test	 Serve 	er offers a /qı	Jery resource with resource content is not empty			
conditions:						
Test Sequence:	Step	Туре	Description			
	1	Stimulus	Client is requested to send a confirmable GET request with			
			three Query parameters (e.g. ?first=1&second=2&third=3) to			
		<u>.</u>	the server's resource			
	2	Check	The request sent by the client contains:			
			• Type = 0 (CON)			
			 Code = 1 (GET) 			
			 Option type = URI-Query (More than one query parameter) 			
	3	Check	Server sends response containing:			
			 Type = 0 (CON) or 2 (ACK) 			
			 Code = 69 (2.05 content) 			
			 Not empty Payload Content format option 			
	4	Verify	Client displays the response			

	Interoperability Test Description				
Identifier:	TD_COAF	TD_COAP_CORE_15			
Objective:	Perform G	ET transactic	on (CON mode, piggybacked response) in a lossy context		
Configuration:	CoAP_CF	G_02			
References:	[1], clause	s 4.4.1, 5.2.1	, 5.8.1		
Pre-test	Gate	way is introdu	ced and configured to produce packet losses		
conditions:	Serve GET	Server offers a <i>/test</i> resource with resource content is not empty that can handle GET			
Test Sequence:	Step	Туре	Description		
	1	Stimulus	Client is requested to send a confirmable GET request to		
			server's resource		
	2	Check	Sent request shall contain:		
			• Type = 0		
			• Code = 1		
			Client generated Message ID		
	3	Check	Server sends response containing:		
			• Type = 2 (ACK)		
			 Code = 69 (2.05 content) 		
			Not empty Payload		
			Content format option		
	4	Verify	Client displays the response		
	5	Check	Repeat steps 1-4 until at least one of the following actions has		
			been observed:		
			One dropped request		
			One dropped response		
	6	Verify	For each case mentioned in step 5:		
			Observe that retransmission is launched		

	Interoperability Test Description				
Identifier:	TD_COAP	TD_COAP_CORE_16			
Objective:	Perform G	Perform GET transaction (CON mode, delayed response) in a lossy context			
Configuration:	CoAP_CF	G_02			
References:	[1], clause	s 4.4.1, 5.2.2,	5.8.1		
_	T				
Pre-test	 Gate 	way is introdu	ced and configured to produce packet losses		
conditions:	 Serve 	er offers a /se	parate resource which cannot be served immediately and		
	which	n cannot be a	cknowledged in a piggybacked way.		
Test Sequence:	Step	Туре	Description		
	1	Stimulus	Client is requested to send a confirmable GET request to server's resource		
	2	Check	The requested sent by the client contains:		
			• Type = 0		
			• Code = 1		
			 a message ID generated by the client 		
	3	Check	Server sends response containing:		
			 Type = 2 (ACK) 		
			 message ID is the same as in the request 		
			empty Payload		
	4	Check	Server sends response containing:		
			• Type = 0 (CON)		
			 Code = 69 (2.05 content) 		
			Not empty Payload		
			Content format option		
	5	Check	Client sends response containing:		
			• Type = 2 (ACK)		
			message ID is the same as in the response of step 3		
			empty Payload		
	6	Verify	Client displays the response		
	7	Check	Repeat steps 1-6 until at least one of the following actions has		
			been observed:		
			One dropped request		
			One dropped request ACK		
			One dropped response		
			One dropped response ACK and its retransmission		
	8	Verify	For each case mentioned in step 7:		
	-	- ,	Observe that retransmission is launched		

		Interop	erability Test Description		
Identifier:	TD_COAF	TD_COAP_CORE_17			
Objective:	Perform G	Perform GET transaction with a separate response (NON mode)			
Configuration:	CoAP_CF	G_01			
References:	[1], clause	es 2.2, 5.2.2, §	5.8.1		
Pre-test conditions:	• Serv	er offers a res	source /separate which cannot be served immediately.		
Test Sequence:	Step	Туре	Description		
	1	Stimulus	Client is requested to send a non-confirmable GET request to server's resource		
	2	Check	 The request sent by the client contains: Type = 1 (NON) Code = 1 (GET) A message ID generated by the Client 		
	3	Check	 Server DOES NOT send response containing: Type = 2 (ACK) Same message ID as in the request in step 2 empty Payload 		
	4	Check	Server sends response containing: • Type = 1 (NON) • Code = 69 (2.05 content) • Not empty Payload d • Content format option		
	5	Verify	Client displays the response		

		Interop	erability Test Description		
Identifier:	TD_COAP_CORE_18				
Objective:	Perform P	Perform POST transaction with responses containing several Location-Path options			
	(CON mod	de)			
Configuration:	CoAP_CF	G_01			
References:	[1], clause	s 5.8.1, 5.10.	8, 5.9.1.1		
	T				
Pre-test	 Serve 	er accepts cre	eation of new resource on <i>/test</i> and the created resource is		
conditions:	locat	ed at /locatio	n1/location2/location3 (resource does not exist yet)		
	-				
Test Sequence:	Step	Туре	Description		
	1	Stimulus	Client is requested to send a confirmable POST request to server's resource		
	2	Check	The request sent by the client contains:		
			 Type = 0 (CON 		
			 Code = 2 (POST) 		
			 An arbitrary payload 		
			Content-format option		
	3	Check	Server sends response containing:		
			 Code = 65 (2.01 created) 		
			 Option type = Location-Path (one for each segment) 		
			 Option values shall contain "location1", "location2" & 		
			"location3" without containing any '/'		
	4	Verify	Client displays the response		

		Interop	erability Test Description		
Identifier:	TD_COAF	TD_COAP_CORE_19			
Objective:	Perform P	OST transact	ion with responses containing several Location-Query options		
-	(CON mod	de)			
Configuration:	CoAP_CF	G_01			
References:	[1], clause	s 5.8.1, 5.10.8	3, 5.9.1.1		
Pre-test	Serve	er accepts cre	ation of new resource on uri <i>/location-query</i> , the location of		
conditions:	the c	reated resour	ce contains two query parameters ?first=1&second=2		
Test Sequence:	Step	Туре	Description		
	1	Stimulus	Client is requested to send a confirmable POST request to		
			server's resource		
	2	Check	The request sent by the client contains:		
			• Type = 0 (CON)		
			 Code = 2 (POST) 		
			An arbitrary payload		
			Content-format option		
	3	Check	Server sends response containing:		
			 Code = 65 (2.01 created) 		
			 Two options whose type is Location-Query 		
			 The first option contains first=1 		
			 The second option contains second=2 		
	4	Verify	Client displays the response		

		Interop	erability Test Description		
Identifier:	TD_COAF	TD_COAP_CORE_20			
Objective:	Perform G	Perform GET transaction containing the Accept option (CON mode)			
Configuration:	CoAP_CF	G_01			
References:	[1], clause	s 5.8.1, 5.10.	5, 5.10.4		
Pre-test	Server sho	ould provide a	a resource /multi-format which exists in two formats:		
conditions:	 text/p 	lain;charset=	utf-8		
	 appli 	cation/xml			
Test Sequence:	Step	Type	Description		
•	Pa	rt A: client r	equests a resource in text format		
	1	Stimulus	Client is requested to send a confirmable GET request to server's resource		
	2	Check	 The request sent request by the client contains: Type = 0 (CON) Code = 1 (GET) Option: type = Accept, value = 0 (text/plain; charset=utf-8) 		
	3	Check	 Server sends response containing: Code = 69 (2.05 content) Option type = Content-Format, value = 0 (text/plain;charset=utf-8) Payload = Content of the requested resource in text/plain;charset=utf-8 format 		
	4	Verify	Client displays the response		
	Pa	rt B: client r	equests a resource in xml format		
	5	Stimulus	Client is requested to send a confirmable GET request to server's resource		
	6	Check	 The request sent by the client contains: Type = 0 (CON) Code = 1 (GET) Option: type = Accept, value = 41 (application/xml) 		
	7	Check	 Server sends response containing: Code = 69 (2.05 content) Option: type = Content-Format, value = 41 (application/xml) Payload = Content of the requested resource in application/xml format 		
	1 0	veniv			

Interoperability Test Description				
Identifier:	TD_COAF	P_CORE_21		
Objective:	Perform GET transaction containing the ETag option (CON mode)			
Configuration:	CoAP_CFG_01			
References:	[1], clause	s 5.8.1, 5.10.	7, 5.10.10, 12.1.12	
Pre-test	Server sh	ould offer a /v	alidate resource which vary in time	
conditions:	Client & se	erver supports	s ETag option	
	The Client	's cache has	been purged	
Test Sequence:	Step	Туре	Description	
		Part A: Verif	ying that client cache is empty	
	1	Stimulus	Client is requested to send a confirmable GET request to server's resource	
	2	Check	The request sent request by the client contains:	
			• Type = 0 (CON)	
			• Code = 1 (GET)	
			No ETag option	
	3	Check	Server sends response containing:	
			• Code = 69 (2.05 content)	
			• Option type = E l ag	
			Option value = an arbitrary E l ag value	
			Not empty Payload	
	4	verity	Client displays the response	
	<u>г</u> е	Stimulus	Client is requested to send a confirmable GET request to	
	5	Sumuus	server's resource so as to check if the resource was undated	
	6	Check	The request sent by the client contains:	
	Ũ	Chook	• Type = 0 (CON)	
			• Code = 1 (GET)	
			Option Type=FTag	
			 Option value=the ETag value received in step 3 	
	7	Check	Server sends response containing:	
			• Code = 67 (2.03 Valid)	
			Option type = ETag	
			 Option value = the ETag value sent in step 3 	
			An empty payload	
	8	Verify	Client displays the response	
	Part C:	Verifying that	at client cache entry is no longer valid	
	9	Stimulus	Update the content of the server's resource from a CoAP client	
	10	Stimulus	Client is requested to send a confirmable GET request to server's resource so as to check if the resource was updated	
	11	Check	The request sent by the client contains:	
			• Type = 0 (CON)	
			• Code = 1 (GET)	
			Option Type=ETag	
			 Option value=the ETag value received in step 3 	
	12	Check	Server sends response containing:	
			• Code = 69 (2.05 Content)	
			Option type = ETag	
			Option value = an arbitrary ETag value which differs	
			from the ETag sent in step 3	
			The payload of the requested resource, which should	
			be different from the payload in step 3	
	13	Verify	Client displays the response	

		Interop	erability Test Description	
Identifier:	TD_COAF	CORE_22		
Objective:	Perform G	ET transactio	n with responses containing the ETag option and requests	
	containing	the If-Match	option (CON mode)	
Configuration:	CoAP_CFG_01			
References:	[1], clauses 5.8.1, 5.10.7, 5.10.9, 12.1.12			
Des test				
Pre-test	 Serve 	er should offe	r a /validate resource	
conditions:	 Client & server supports ETag and If-Match option 			
	I he Client 's cache has been purged			
		-		
Test Sequence:	Step	Туре	Description	
		Preambl	e: client gets the resource	
	1	Stimulus	Client is requested to send a confirmable GET request to	
			server's resource	
	2	Chook	The request cent by the client containe:	
	2	Check	The request sent by the client contains. $T_{\rm res} = 0$ (CON)	
			- 1ype = 0 (00N)	
	2	Chook	Goue = 1 (GET) Server sends response containing:	
	3	CHECK	c Codo = 60 (2.05 content)	
			• Code = $09 (2.00 \text{ coment})$	
			Option type = E Tag	
			Option value = an arbitrary Etag value	
		D	Not empty Payload	
	4	Ctimuluo	Client is requested to cond a confirmable DUT request to	
	4	Sumulus	Client is requested to send a confirmable POT request to	
	5	Chook	The request sent by the glight contains:	
	5	Check	The request sent by the client contains. $T_{\rm res} = 0$ (CON)	
			• Type = 0 (CON) • Code = 2 (PUT)	
			• Coue = $5(POT)$	
			Option Type=II-Match Option value_ETag value received in step 2	
			 Option value=E ray value received in step 5 An orbitrory poyload (which differe from the poyload 	
			 An arbitrary payloau (which differs from the payloau received in step 3) 	
	6	Check	Server sends response containing:	
	0	Oneck	 Code – 68 (2.04 Changed) 	
	7	Verify	Client displays the response and the server changed its	
	,	Veniy	resource	
	1	Part	B: concurrent updates	
	8	Stimulus	Client is requested to send a confirmable GET request to	
			server's resource	
	9	Check	The request sent by the client contains:	
	-		• Type = 0 (CON)	
			• Code = 1 (GET)	
	10	Check	Server sends response containing:	
			• Code = 69 (2.05 content)	
			 Option type = ETag 	
			 Option value = an arbitrary Etag value which differs 	
			from the ETag sent in step 3	
			The Payload sent in step 5	
	11	Verify	Client displays the response	
	12	Stimulus	Update the content of the server's resource from a CoAP	
			client	
	13	Stimulus	Client is requested to send a confirmable PUT request to	
			server's resource so as to perform an atomic update	

14	Check	 The request sent by the client contains: Type = 0 (CON) Code = 3 (PUT) Option Type=If-Match Option value=ETag value received in step 10 An arbitrary payload (which differs from the previous payloads)
15	Check	 Server sends response containing: Code = 140 (4.12 Precondition Failed)
16	Verify	Client displays the response and the server did not update the content of the resource

	Interoperability Test Description				
Identifier:	TD_COAP_CORE_23				
Objective:	Perform PUT transaction containing the If-None-Match option (CON mode)				
Configuration:	CoAP_CF	G_01			
References:	[1], clause	s 5.8.1, 5.10.	7, 5.10.10, 12.1.12		
Due (est					
Pre-test	Server sho	Server should offer a /create1 resource, which does not exist and which can be			
conditions.	Client 8 cr	rue client	If Non Match		
	Client & St				
Test Sequence:	Step	Type	Description		
•		Pa	rt A: single creation		
	1	Stimulus	Client is requested to send a confirmable PUT request to		
			server's resource so as to atomically create the resource.		
	2	Check	The request sent by the client contains:		
			• Type = 0 (CON)		
			• Code = 3 (PUT)		
			 Option Type=If-None-Match 		
			An arbitrary payload		
	3	Check	Server sends response containing:		
			• Code = 65 (2.01 Created)		
	4	Verify	Client displays the response and the server created a new		
			resource		
		Part E	3: concurrent creations		
	5	Stimulus	Client is requested to send a confirmable PUT request to		
			server's resource so as to atomically create the resource.		
	6	Check	The request sent by the client contains:		
			• Type = 0 (CON)		
			• Code = 3 (PUT)		
			Option Type=If-None-Match		
			An arbitrary payload		
	7	Check	Server sends response containing:		
			140 (4.12 Precondition Failed)		
	8	Verify	Client displays the response		

	Interoperability Test Description			
Identifier:	TD_COAP_CORE_24			
Objective:	Perform POST transaction with responses containing several Location-Path options (Reverse Proxy in CON mode)			
Configuration:	CoAP_CFG_03			
References:	[1], clause	s 5.8.1, 5.10.	8, 5.9.1.1, 8.2.2, 8.2.1, 10.2.2, 11.2	
Pre-test	Proxy is configured as a reverse-proxy for the server			
conditions:	 Proxy 	/'s cache is cl	eared	
	Serve	er accepts cre	eation of new resource on /create2 and the created resource is	
	IOCal		Invocation2/locations (resource does not exist yet)	
Test Sequence:	Sten	Туре	Description	
rest bequence.	1	Stimulus	Client is requested to send a confirmable POST request to	
	1	Otimulus	orient is requested to send a commable r OOT request to	
	2	Check	The POST sent by the client contains:	
	_	Chicon	• Type = 0 (CON)	
			• Code = 2 (POST)	
			An arbitrary payload	
			Content-format option	
	3	Check	The Proxy forwards the POST request to server's resource	
	_		and that it contains:	
			 Type = 0 (CON) 	
			• Code = 2 (POST)	
			An arbitrary payload	
			Content-format option	
	4	Check	Server sends a response to the proxy containing:	
			 Code = 65 (2.01 created) 	
			 Option type = Location-Path (one for each segment) 	
			 Option values shall contain "location1", "location2" & 	
			"location3" without contain any '/'	
	5	Check/	Observe that the Proxy forwards the response (in step 4) to	
			client and check that the forwarded response contains:	
			• Code = 65 (2.01 created)	
			 Option type = Location-Path (one for each segment) 	
			 Option values shall contain "location1", "location2" & "location 0" with put contain any 10" 	
	<u> </u>) / a rife /	"iocation3" without contain any /	
	6	Verify	Client displays the response	
		verity	Client interface returns the response	
			2.01 Greated	
Pre-test conditions: Test Sequence:	 Proxy Proxy Server locate Step 1 2 3 4 5 6 7 	/ is configured /'s cache is cl er accepts cre ed at /locatio Type Stimulus Check Check Check Check/ Verify Verify	d as a reverse-proxy for the server leared sation of new resource on /create2 and the created resource is n1/location2/location3 (resource does not exist yet) Description Client is requested to send a confirmable POST request to proxy The POST sent by the client contains: • Type = 0 (CON) • Code = 2 (POST) • An arbitrary payload • Content-format option The Proxy forwards the POST request to server's resource and that it contains: • Type = 0 (CON) • Code = 2 (POST) • An arbitrary payload • Content-format option Server sends a response to the proxy containing: • Code = 65 (2.01 created) • Option type = Location-Path (one for each segment) • Option values shall contain "location1", "location2" & "location3" without contain any '/' Client displays the response Client interface returns the response 2.01 created Location: coap://proxy/location1/location2/location3	

		Interop	erability Test Description	
Identifier:	TD_COAP_CORE_25			
Objective:	Perform POST transaction with responses containing several Location- Query option			
	(Reverse	(Reverse proxy)		
Configuration:	CoAP_CF	G_03		
References:	[1], clause	es 5.8.1, 5.10.	8, 5.9.1.1, 8.2.2, 8.2.1, 10.2.2, 11.2	
Pre-test	 Prox 	y is configure	d as a reverse-proxy for the server	
conditions:	 Prox 	y's cache is c	leared	
	 Serv 	er accepts cr	eation of new resource on uri /location-query, the location of	
	the o	created resour	rce contains two query parameters ?first=1&second=2	
		I —		
Test Sequence:	Step	Туре	Description	
	1	Stimulus	Client is requested to send a confirmable POST request to	
			proxy	
	2	Check	Proxy receives the request from client & forwards it to server's	
			resource	
	3	Check	Forwarded request shall contain:	
			• Type = 0 (CON)	
			 Code = 2 (POST) 	
			An arbitrary payload	
			Content-format option	
	4	Check	Server sends response to proxy containing:	
			 Code = 65 (2.01 created) 	
			 Two options whose type is Location-Query 	
			 The first option contains first=1 	
			 The second option contains second=2 	
	5	Check	Proxy forwards the response to client	
	6	Check	Client displays the message	
	7	Verify	Client interface returns the response:	
			2.01 created	
			Location: coap://proxy/?first=1&second=2	

		Interope	erability Test Description
Identifier:	TD_COAF	CORE_26	
Objective:	Perform G	ET transaction	n containing the Accept option (CON mode
Configuration:	CoAP_CF	G_03	
References:	[1], clause	s 5.8.1, 5.10.5	5, 5.10.4, 8.2.2, 8.2.1, 10.2.2, 11.2
Pre-test conditions:	 Prox Prox Servent - - 	y is configured y's cache is cle er should prov text/plain;cha application/xn	as a reverse-proxy for the server eared ide a resource /multi-format which exists in two formats: rset=utf-8 nl
Test Sequence:	Step	Type	Description
		Part A: c	lient requests text format
	1	Stimulus	Client is requested to send a confirmable GET request to proxy
	2	Check	Proxy receives the request from client & forwards it to server's resource
	3	Check	 Forwarded request shall contain: Type = 0 (CON) Code = 1 (GET) Option: type = Accept, value = 0 (text/plain;charset=utf- 8)
	4	Check	 Server sends response containing: Code = 69 (2.05 content) Option: type = Content-Format, value = 0 (text/plain;charset=utf-8) Payload = Content of the requested resource in text/plain;charset=utf-8 format
	5	Check	Proxy forwards the response to client
	6	Verify	Client receives & displays the response
	7	Check	 Response contains: Code = 69 (2.05 content) Option: type = Content-Format, value = 0 (text/plain;charset=utf-8) Payload = Content of the requested resource in text/plain;charset=utf-8 format
		Part B: c	lient requests xml format
	8	Stimulus	Client is requested to send a confirmable GET request to Proxy
	9	Check	Proxy forwards the request to server
	10	Check	 Sent request shall contain: Type = 0 (CON) Code = 1 (GET) Option: type = Accept, value = 41 (application/xml)
	11	Check	 Server sends response containing: Code = 69 (2.05 content) Option: type = Content-Format, value = 41 (application/xml) Payload = Content of the requested resource in application/xml format
	12	Check	Proxy forwards the response to client
	<u> 13</u> 14	Verify Check	Client receives & displays the response Client displays the response received: • Code = 69 (2.05 content) • Option: type = Content-Format, value = 41 (application/xml) • Payload = Content of the requested resource in application/xml format

		Interop	erability Test Description
Identifier:	TD_COAF	P_CORE_27	
Objective:	Perform G	ET transactio	on with responses containing the ETag option and requests
0 11 11	Icontaining the IT-Match option (CON mode)		
Configuration:	COAP_CF	<u>G_03</u>	7 5 40 0 40 4 40 0 0 0 4 40 0 0 44 0
References:	[[1], clause	s 5.8.1, 5.10.	7, 5.10.9, 12.1.12, 8.2.2, 8.2.1, 10.2.2, 11.2
Pre-test	Prox	v is configure	d as a reverse-proxy for the server
conditions:	Prox	y is conliguie v's cache is c	leared
	 Sorver should offer a /validate resource with resource content is not a 		
	Clien	t & server su	nports ETag ontion and If-Match ontion
	e olici		
Test Sequence:	Step	Type	Description
		Preamb	le: client gets the resource
	1	Stimulus	Client is requested to send a confirmable GET request to
			proxy
	2	Check	Proxy forwards the request to server
	3	Check	Forwarded request shall contain:
			• Type = 0 (CON)
			• Code = 1 (GET)
	4	Check	Server sends response containing:
			Code = 69 (2.05 content)
			Option type = E l ag
			Option value = an arbitrary E i ag value
	F	Chook	Prove forwards the response to alight
	IJ		priory forwards the response to cheft
	6	Stimulus	Client is requested to send a confirmable DLIT request to
	0	Sumuus	Proxy
	7	Check	Sent request shall contain:
			• Type = 0 (CON)
			• Code = 3 (PUT)
			Option Type=If-Match
			 Option value=ETag value received in step 4
			An arbitrary payload (which differs from the payload
			received in step 3)
	8	Verify	Proxy forwards the request to servers resource & server
			updates the resource
	9	Check	Server sends response containing:
			 Code = 68 (2.04 Changed)
			Option type = ETag
			Option value = an arbitrary ETag value which differs
	40		trom the E I ag received in step 4
	10	Check	Proxy forwards the response to client
	11	Check	Forwarded response contains:
			• Code = $b\delta$ (2.04 Unanged)
			 Option type = ETag Option value = come ETag value found in stan 8
	10	Vorify	 Option value = same = rag value round in step 8 Client displays the response
	12	Dart	B: concurrent undates
	12	Stimulus	Undate the content of the server's resource from a CoAP
	13	Sumulus	client
	14	Stimulus	Client is requested to send s confirmable PUT request to
			proxy so as to perform an atomic update
	15	Check	Sent request shall contain:
			• Type = 0 (CON)
			• Code = 3 (PUT)
			Option Type=If-Match
			 Option value=ETag value received in step 8
			An arbitrary payload (which differs from the previous
			payloads)
	16	Check	Proxy forwards the request to server's resource

17	Check	 Sent request shall contain: Type = 0 (CON) Code = 3 (PUT) Option Type=If-Match Option value=same ETag value found in step 14 An arbitrary payload (which differs from the previous payloads)
18	Check	 Server sends response containing: Code = 140 (4.12 Precondition Failed)
19	Verify	Proxy forwards the response to client
20	Check	Response contains:
		 Code = 140 (4.12 Precondition Failed)
21	Verify	Client displays the response

Interoperability Test Description					
Identifier:	TD_COAP_CORE_28				
Objective:	Perform GET transaction with responses containing the ETag option and requests				
	containing the If-None-Match option (CON mode) (Reverse proxy)				
Configuration:	CoAP_CF	G_03			
References:	[1], clause	s 5.8.1, 5.10.7	7, 5.10.10, 12.1.12, 8.2.2, 8.2.1, 10.2.2, 11.2		
	-		· · · · · · · · · · · · · · · · · · ·		
Pre-test	Proxy	y is configured	l as a reverse-proxy for the server		
conditions.	Proxy's cache is cleared				
	• Serve	er should offer	a /create3 resource, which does not exist and which can be		
	creat	ed by the clier	II norte la Nortek		
	• Clien	t & server sup	ports II-None-Match		
Teet Convences	Ston	Time	Description		
Test Sequence.	Step	l Type Pa	Description		
	1	Stimuluc	Client is requested to cond a confirmable PLIT request to		
	1	Sumulus	provy to atomically create resource in server		
	2	Check	Proxy forwards the request to server		
	3	Check	Forwarded t request shall contain:		
	Ŭ	Oncork	• Type = 0 (CON)		
			• Code = 3 (PUT)		
			Ontion Type=If-None-Match		
			An arbitrary payload		
	4	Check	Server sends response containing:		
	•	Chicola	• Code = $65(2.01 \text{ Created})$		
	5	Check	Proxy forwards the response to client		
	6	Verify	Client displays the response & and server created new		
	-	,	resource		
	Part B: concurrent creations				
	5	Stimulus	Client is requested to send s confirmable PUT request to		
			proxy to atomically create resource in server		
	6	Check	Sent request shall contain:		
			• Type = 0 (CON)		
			• Code = 3 (PUT)		
			 Option Type=If-Non-Match 		
			Option value=Received ETag value		
	7	Check	Server sends response containing:		
			140 (4.12 Precondition Failed)		
	8	Verify	Proxy forwards the response to client		
	9	Check	Response contains:		
			140 (4.12 Precondition Failed)		
ļ	10	Verify	Client displays the response		

		Interop	erability Test Description			
Identifier:	TD_COAF	TD_COAP_CORE_29				
Objective:	Perform G	ET transactio	n with responses containing the Max-Age option (Reverse			
	proxy)					
Configuration:	CoAP_CF	G_03				
References:	[1], clause	s 5.8.1, 5.10.0	6, 5.9.1.3, 5.9.1.5, 8.2.2, 8.2.1, 10.2.2, 11.2			
Pre-test	 Proxy 	Proxy offers a cache				
conditions:	Prox	, y is configured	d as a reverse-proxy for the server			
	 Serve 	ers resource v	ary in time and supports Max-Age option			
	Proxy	y's cache is cl	eared			
	 Serve 	er offers a res	ource /validate that varies in time, with a Max-Age set to 30 s			
	•					
Test Sequence:	Step	Туре	Description			
	1	Stimulus	A confirmable GET request is sent to Proxy from Client			
	2	Check	Proxy Sends request containing:			
			 Type = 0 (CON) 			
			• Code = 1 (GET)			
	3	Check	Server sends response containing:			
			 Code = 69 (2.05 Content) 			
			 Option type = ETag 			
			 Option value = ETag value 			
			 Option type = Max-age 			
			Option value			
			Not empty Payload			
	4	Verify	Proxy forwards response to client			
	5	Stimulus	A confirmable GET request is sent to proxy from Client before			
			Max-Age expires			
	6	Check	Proxy dos not forward any request to the server			
	7	Check	Proxy sends response to client			
	8	Verify	Response contains:			
			 Option type = Max-age 			
			 Option Value = new Max-age 			
			Payload cached			

8.2.2 CoRE Link Format

Interoperability Test Description					
Identifier:	TD_COAF	TD_COAP_LINK_01			
Objective:	Access to	well-known ir	terface for resource discovery		
Configuration:	CoAP_CF	G_01			
References:	[2]				
Pre-test	Client and	server suppo	rts CoRE Link Format		
conditions:	Server su	oports /.well-k	known/core resource and the CoRE Link Format		
Test Sequence:	Step	Туре	Description		
	1	Stimulus	Client is requested to retrieve Server's list of resource		
	2	Check	Client sends a GET request to Server for /.well-known/core		
			resource		
	3	Check	Server sends response containing:		
			 Content-format option indicating 40 (application/link- 		
			format)		
			 Code indicating 69 (2.05 content) 		
			 Payload indicating all the links available on Server 		
	4	Verify	Client displays the list of resources available on Server		

Interoperability Test Description					
Identifier:	TD_COAP_LINK_02				
Objective:	Use filtere	d requests for	limiting discovery results		
Configuration:	CoAP_CF	G_01			
References:	[2], clause	4.1			
Pre-test	Client	Client supports CoRE Link Format			
conditions.	 Serve 	r supports Co			
	 Serve 	r offers differe	ent types of resources (<i>Type1</i> , <i>Type2</i> ,; see note)		
		-	N 1.4		
Test Sequence:	Step	Туре	Description		
	1	Stimulus	Client is requested to retrieve Server's list of resource of a specific type <i>Type1</i>		
	2	Check	Client sends a GET request to Server for /.well-known/core resource containing URI-Query indicating "rt= <i>Type1</i> "		
	3 Check Server sends response containing:				
	Content- format option indicating 40 (application/link				
	format) Payload indicating only the links of type Type1				
	available on Server				
	4	Verify	Client displays the list of resources of type Type1 available on		
	Server				
NOTE: Type1, Type2, refer to real resource types available on Server and shall be extracted from					
Server's /.well-known/core resource.					

Interoperability Test Description				
Identifier:	TD_COAP_LINK_03			
Objective:	Handle err	pty prefix val	ue strings	
Configuration:	CoAP_CF	G_01		
References:	[2], clause	4.1 §2		
Pre-test	 Client 	supports Cor	e Link Format	
conditions:	 Serve 	r supports Co	ore Link Format	
	 Serve 	r offers differe	ent types of resources (Type1, Type2,; see note)	
	 Serve 	r offers resou	rces with no type (i.e. no rt attribute)	
Test Sequence:	Step	Туре	Description	
	1	Stimulus	Client is requested to retrieve Server's list of resources	
			matching an rt empty value	
	2	Check	Client sends a GET request to Server for /.well-known/core	
		resource containing URI-Query indicating rt="*"		
3 Check Server sends response containing:				
			 Content-format option indicating 40 (application/link- 	
			format)	
			 Payload indicating only the links having an rt attribute 	
	4	Verify	Client displays the list of resources with rt attribute available	
	on Server			
NOTE: Type1, Type2, refer to real resource types available on Server and shall be extracted from Server's / well-known/core resource				

	Interoperability Test Description				
Identifier:	TD_COAF	TD_COAP_LINK_04			
Objective:	Filter disco	Filter discovery results in presence of multiple rt attributes			
Configuration:	CoAP_CF	G_01			
References:	[2], clause	s 3.1, 4.1 §2			
Pre-test	 Client 	Client supports Core Link Format			
conditions:	 Serve 	Server supports Core Link Format			
	 Serve 	r offers 4 gro	ups of resources:		
	1.	1. Resources with rt="Type1 Type2"			
	2.	Resources with rt="Type2 Type3"			
	3.	Resources with rt="Type1 Type3"			
	4. Resources with rt=""				
Test Sequence:	Step	Туре	Description		
	1	Stimulus	Client is requested to retrieve Server's list of resources of a specific type <i>Type2</i>		
	2	Check	Client sends a GET request to Server for /.well-known/core resource containing URI-Query indicating rt="Type2"		
	3	Check	Server sends response containing:		
			 Content-format option indicating 40 (application/link- 		
			format)		
			 Payload indicating only the links of groups 1 and 2 		
	4	Verify	Client displays the list of resources of type Type2 available on		
			Server		

	Interoperability Test Description				
Identifier:	TD_COAF	TD_COAP_LINK_05			
Objective:	Filter disco	overy results u	using if attribute and prefix value strings		
Configuration:	CoAP_CF	G_01			
References:	[2], clause	s 3.2, 4.1 §5			
Pre-test	Client	Client supports Core Link Format			
conditions:	 Serve 	Server supports Core Link Format			
	 Serve 	er offers 4 grou	ups of resources:		
	1.	Resources wi	th if="lf1"		
	2.	Resources wi	th if="If2"		
	3.	Resources wi	th if="foo"		
	4. Resources with no if attribute				
Test Sequence:	Step	Туре	Description		
	1	Stimulus	Client is requested to retrieve Server's list of resources		
			matching the interface description pattern "If*"		
	2	Check	Client sends a GET request to Server for /.well-known/core		
			resource containing URI-Query indicating if="If*"		
	3	Check	Server sends response containing:		
			 Content-format option indicating 40 (application/link- 		
			format)		
			 Payload indicating only the links of groups 1 and 2 		
	4	Verify	Client displays the retrieved list of resources		

Interoperability Test Description					
Identifier:	TD_COAF	TD_COAP_LINK_06			
Objective:	Filter disco	overy results	using sz attribute and prefix value strings		
Configuration:	CoAP_CF	G_01			
References:	[2], clause	s 3.3, 4.1 §5			
Pre-test	 Client 	Client supports Core Link Format			
conditions:	Server supports Core Link Format				
	 Serve 	r offers resou	Irce with sz attribute		
	 Serve 	 Server offers resources with no sz attribute 			
Test Sequence:	Step	Туре	Description		
	1	Stimulus	Client is requested to retrieve Server's list of resources having a sz attribute		
	2	Check	Client sends a GET request to Server for /.well-known/core resource containing URI-Query indicating sz="*"		
	3	Check	Server sends response containing:		
			 Content-format option indicating 40 (application/link- format) 		
			 Payload indicating only the links having a sz attribute 		
	4	Verify	Client displays the retrieved list of resources		

		Interop	erability Test Description		
Identifier:	TD_COAF	TD_COAP_LINK_07			
Objective:	Filter disc	overy results	using href attribute and complete value strings		
Configuration:	CoAP_CF	G_01			
References:	[2], clause	94.1			
Pre-test	Client	t supports Co	re Link Format		
conditions:	Server supports Core Link Format				
	 Serve 	 Server offers resources /link1 /link2 and /link3 			
Test Sequence:	Step	Туре	Description		
	1	Stimulus	Client is requested to retrieve the link-value anchored at /link1		
	2	Check	Client sends a GET request to Server for /.well-known/core		
			resource containing URI-Query indicating href="/link1"		
	3	Check	Server sends response containing:		
			 Content-format option indicating 40 (application/link- 		
			format)		
			 Payload indicating only the link for /link1 		
	4	Verify	Client displays the retrieved list of resources		

		Interop	erability Test Description	
Identifier:	TD_COAP_LINK_08			
Objective:	Filter disco	overy results	using href attribute and prefix value strings	
Configuration:	CoAP_CF	G_01		
References:	[2], clause	4.1		
Pre-test	Client supports Core Link Format			
conditions:	Server supports Core Link Format			
	Server offers resources /link1 /link2 and /link3			
	Server offers resource /test			
Test Sequence:	Step	Туре	Description	
	1	Stimulus	Client is requested to retrieve the link-value anchored at /link*	
	2	Check	Client sends a GET request to Server for /.well-known/core	
			resource containing URI-Query indicating href="/link*"	
	3	Check	Server sends response containing:	
			 Content-format option indicating 40 (application/link- 	
			format)	
			 Payload indicating only the link matching /link* 	
	4	Verify	Client displays the retrieved list of resources	

	Interoperability Test Description				
Identifier:	TD_COAP_LINK_09				
Objective:	Arrange link descriptions hierarchically				
Configuration:	CoAP_CF	G_01			
References:	[2], clause	5 §4			
Pre-test	Client supports Core Link Format				
conditions:	 Serve 	r supports Co	ore Link Format		
	 Serve 	r offers an er	try located at /path with ct=40		
	 Serve 	r offers sub-r	esources /path/sub1, /path/sub2, (see note)		
Test Sequence:	Step	Туре	Description		
	1	Stimulus	Client is requested to retrieve one of the sub-resources		
	2	Check	Client sends a GET request to Server for /.well-known/core		
			resource		
	3	Check	Server sends response containing:		
			 Content-format option indicating 40 (application/link- 		
			format)		
			 Payload indicating the link description for /path 		
	4	Check	Client sends a GET request for /path to Server		
	5	Check	Server sends response containing:		
			Content-format option indicating 40 (application/link-format)		
			Payload indicating the link description for /path/sub1,		
			/path/sub2,		
	6	Check	Client sends a GET request for /path/sub1		
	7	Check	Server sends 2.05 (Content) response.		
			Payload contains /path/sub1		
	8	Verify	Client displays the retrieved sub-resource.		
NOTE: /path/sub1, /path/sub2, refer to real resources available on Server and shall be extracted from					
Server's /.well-known/core resource.					

	Interoperability Test Description				
Identifier:	TD_COAP_LINK_10				
Objective:	Handle an	alternate link			
Configuration:	CoAP_CF	G_01			
References:	[2], clause	5 §6			
Pre-test	Client supports Core Link Format				
conditions:	Server supports Core Link Format				
	 Serve 	Server offers resources /test and /alternate (see note)			
	 Resource /alternate is anchored at /test (i.e. anchor="/test") with rel="alternate" 				
Test Sequence:	Step	Туре	Description		
	1	Stimulus	Client is requested to retrieve the resource /test using the		
	alternate /alternate				
	2	Check	Client sends a GET request to Server for /alternate		
	3	Check	Server sends response containing the resource /test		
	4	Verify	Client displays the response		
NOTE: /test and /	NOTE: /test and /alternate refer to a real resource and its alternate available on Server and shall be				
extracted from Server's <i>I.well-known/core</i> resource.					

Identifier:	TD COAP BLOCK 01				
Objective:	Handle GET blockwise transfer for large resource (early negotiation)				
Configuration:	CoAP CFG 01				
References:	[4], clause 2.2				
Pre-test	Client supports Block transfers				
conditions:	Server supports Block transfers				
	Server	offers a larg	e resource /large		
	 Client k 	nows /large	requires block transfer		
-	1	0			
Test Sequence:	Step	Туре	Description		
-	1	Stimulus	Client is requested to retrieve resource /large		
	2	Check	Client sends a GET request. The request optionally contains		
			a Block2 option indicating:		
			• NUM = 0;		
			• M = 0:		
			• SZX = the desired block size.		
	3	Check	Server sends 2.05 (Content) response with a Block2 option		
	_		indicating:		
			• NUM = 0:		
			• M = 1:		
			 SZX is less or equal to the desired block size indicated 		
			by the GET request		
			Pavload size is 2 ^{SZX+4} bytes.		
	4	Check	Client send GET requests for further blocks indicating:		
	(see note)	0	 NUM = i where "i" is the block number of the current 		
	(,		block:		
			• M = 0:		
			• SZX is the SZX at step 3		
	5	Check	Server sends 2.05 (Content) response containing Block2		
	(see note)	Check	option indicating:		
	(000 11010)		 NUM = i where "i" is the block number used at step 4. 		
			• $M = 1$		
			 S7X is the S7X at step 3 		
			Pavload size SHALL be 2 ^{SZX+4} bytes		
	6	Check	Client send GET request for the last block indicating:		
	Ũ	Oneon	 NI IM – n where "n" is the last block number: 		
			• $M = 0$:		
			• $M = 0$, • S7X is the S7X at step 3		
	7	check	Server sends 2.05 (Content) response with a Block2 ontion		
	'	CHECK	indicating:		
			 NILIM – n where "n" is the block number used at stop 6: 		
			• $M = 0$		
			• $W = 0$, • CTV is the CTV at step 2		
			• SLA IS the SLA at step 3. Devled size is leaser or equal to 2^{SZX+4} bytes		
		Marita	Client displays the received information		
	U B	verity	Unent displays the received information		

8.2.3 Blockwise transfers

NOTE: Steps 4 and 5 are in a loop.

Identifier:	TD_COAP_BLOCK_02				
Objective:	Handle GET	blockwise t	ransfer for large resource (late negotiation)		
Configuration:	CoAP_CFG_01				
References:	[4], clause 2	.2			
Pre-test	Client supports Block transfers				
conditions:	Server supports Block transfers				
	 Server of 	offers a large	e resource /large		
	Client d	nes not kno	w /large requires block transfer		
	e onorit d		W / large lequilee block traileler		
Test Sequence:	Step	Type	Description		
root ooquomoor	1	Stimulus	Client is requested to retrieve resource /large		
	2	Check	Client sends a GET request not containing Block2 option		
	3	Check	Server sends 2.05 (Content) response with a Block2 option		
	0	Oneok	indicating.		
			• $NIIM = 0$		
			• M – 1		
			 S7X – the proposed block size 		
			Pavload size is 2 ^{SZX+4} bytes		
	1	Check	Client switches to blockwise transfer mode and sends a GET		
	7	Oneck	request with a Block2 ontion indicating:		
			MIM is the next block number (should be equal to		
			$^{\circ}$ SZX_in_step_4 - SZX_in_step_3.		
			M = 0		
			 NI = 0, SZX is loss or equals to SZX at stop 2 		
	Б	Chock	Sorver conde 2.05 (Content) response with a Block2 option		
	5	CHECK	indicating:		
			 NI IM – k whore "k" is the block number used at step 4: 		
			• NOW = K where K is the block humber used at step 4, • $M = 1$:		
			• $ V = 1$, • CZV is the CZV at step 4		
			• SZX is the SZX at step 4.		
	6	Chaol	Payload Size is z Dyles.		
	o (acc poto)	Check	Chefit sends GET request for further blocks indicating.		
	(see note)		 NUM = I where I is the block number of the current block; 		
			DIOCK,		
			• $ \mathbf{V} = \mathbf{U}$, • $ \mathbf{C}\mathbf{T} $ is the $ \mathbf{C}\mathbf{T} $ at star 4		
	7	Ohaala	• SZX is the SZX at step 4.		
	(Спеск	Server sends 2.05 (Content) response with a Block2 option		
	(see note)				
			• NUM = I where "I" is the block number used at step 6;		
			• M = 1;		
			• SZX is the SZX at step 4.		
	0	Ohaala	Payload size is 2° bytes.		
	8	Спеск	Client send GET request for the last block indicating:		
			• NUM = n where "n" is the last block number;		
			• $M = 0;$		
			SZX is the SZX at step 4.		
	9	Check	Server senas 2.05 (Content) response with a BIOCK2 option		
			Indicating:		
			• INUM = n where "n" is the block number used at step 8;		
			• IVI = 0;		
			• S $\angle X$ is the S $\angle X$ at step 4.		
	4.6	N/ 11	Payload size is lesser or equal to 2 ⁻²²⁴ .		
	10	Verity	Ulient displays the received information		
NOTE: Steps 6 a	nd 7 are in a	loop.			

Identifier:	TD_COAP_BLOCK_03			
Objective:	Handle PUT	blockwise t	ransfer for large resource	
Configuration:	CoAP_CFG_01			
References:	[4], clause 2	.2		
Pre-test	 Client s 	upports Bloo	ck transfers	
conditions:	Server a	supports Blo	ock transfers	
	Server	offers a larg	e updatable resource /large-update	
-		0		
Test Sequence:	Step	Type	Description	
-	1	Stimulus	Client is requested to update resource /large-update on Server	
	2	check	Client sends a PUT request containing Block1 option indicating: • NUM = 0;	
			 M = 1; SZX = the desired block size. Payload size is 2^{SZX+4} bytes. 	
	3	Check	Server sends 2.04 (Changed) response with a Block1 option indicating: • NUM = 0;	
			 M = 0 (stateless) or 1 (atomic); SZX is less or equal to the SZX at step 2. 	
	4	Check	Client sends further requests containing Block1 option	
	(see note)		indicating:	
			 NUM = i where "i" is the block number of the current block. If the server decreased the SZX parameter in step 3, then the client should adapt the block size accordingly and may resume the transfer from block id 2^{size_in_step_2-size_in_step_3} instead of block 1) 	
			• M = 1;	
			• SZX is the SZX at step 3.	
			Payload size is 2 ^{32,44} bytes.	
	5	Check	Server sends 2.04 (Changed) response containing Block1	
	(see note)		option indicating:	
			• NUM = 1 where "1" is the block number used at step 4;	
			• IVI = 0 (stateless) or 1 (atomic);	
			SZX is the SZX at step 3.	
	б	Спеск	Linent send PUT request containing the last block and	
			NIIM - n whore "n" is the last block number:	
			\sim 101 - 0, \sim S7X is the S7X at step 3	
			Pavload size is lesser or equal to 2^{SZX+4}	
	7	Check	Server sends 2.04 (Changed) response with a Block1 ontion	
		CHOOK	indicating:	
			• NUM = n where "n" is the block number used at step 6:	
			• M = 0;	
			SZX is the SZX at step 3.	
	8	Verify	Server indicates presence of the complete updated resource	
			/large-update	
NOTE: Steps 4 a	nd 5 are in a	loop.		

Identifier:	TD_COAP_BLOCK_04			
Objective:	Handle POST blockwise transfer for large resource			
Configuration:	CoAP CFG 01			
References:	[4], clause 2	2		
	<u>[[·]</u> , ········			
Pre-test	Client s	upports Blog	ck transfers	
conditions:	Server supports Block transfers			
oonaniono.	Server	acconte cro	ation of now resources on large-create	
	• Server		allon of new resources on narge-create	
Toot Seguenee	Stop	Turno	Description	
Test Sequence:	Step			
	1	Stimulus	Client is requested to create a new resource /large-create on	
		Chaoli	Server	
	2	Check	circline sends a POST request containing block r option	
			indicating.	
			• NUM = 0;	
			• M = 1;	
			• $SZX =$ the desired block size.	
			Payload size is 2 ^{52/14} bytes.	
	3	Check	Server sends 2.01 (Created) response containing	
			Block1 option indicating:	
			• NUM = 0;	
			 M = 0 (stateless) or 1 (atomic); 	
			 SZX is less or equal to the SZX at step 2. 	
	4	Check	Client sends further requests containing	
	(see note)		Block1 option indicating:	
	· · · ·		• NUM = i where "i" is the block number of the current	
			block. If the server decreased the SZX parameter in	
			step 3, then the client should adapt the block size	
			accordingly and may resume the transfer from block id	
			2 ^{size_in_step_2-size_in_step_3} instead of block 1)	
			• M = 1:	
			• SZX is the SZX at step 3	
			Payload size is 2 ^{SZX+4} bytes	
	5	Check	Server sends 2.01 (Created) response containing	
	(see note)	Oneen	Block1 ontion indicating:	
	(000 11010)		 NI IM – i where "i" is the block number used at step 4: 	
			• Now = 1 where 1 is the block humber used at step 4, • $M = 1$:	
			• $M = 1$, • S7Y is the S7Y at step 2	
	6	Chaoly	SZA IS LIFE SZA at Step 5	
	0	Check	cilent send POT request containing the last block and	
			Indicating.	
			• M = 0;	
			• SZX is the SZX at step 3.	
			Payload size is lesser or equal to 2	
	(Check	Server sends 2.01 (Created) response containing Block1	
			option indicating:	
			 NUM = n where "n" is the block number used at step 6; 	
			• M = 0;	
			SZX is the SZX at step 3.	
	8	Verify	Server indicates presence of the complete new resource	
			/large-create	
NOTE: Steps 4 a	nd 5 in a loor	D.		

		Interop	erability Test Description		
Identifier:	TD_COAP_	TD_COAP_OBS_01			
Objective:	Handle resource observation with CON messages				
Configuration:	CoAP_CFG	_01			
References:	[4], clause 1	.2,			
Pre-test	Client supports Observe option				
conditions:	 Server 	supports Ob	oserve option		
	 Server which p 	offers an ob roduces cor	servable resource /obs changing periodically (e.g. every 5 s) nfirmable notifications		
Test Sequence:	Step	Туре	Description		
	1	Stimulus	Client is requested to send to the server a confirmable GET		
			request with observe option for resource /obs		
	2	Check	The request sent by client contains:		
			• Type = 0 (CON)		
			• Code = 1 (GET)		
			 Token value = a value generated by the client 		
			 Observe option = empty 		
	3	Check	Server sends the response containing:		
			• Type = 2 (ACK)		
			 Content-format of the resource /obs 		
			 Token value = same as one found in the step 2 		
			 Observe option with a sequence number 		
	4	Check	Server sends a notification containing:		
	(see note)		 Type = 0 (CON) 		
			 Content-format = same as one found in the step 3 		
			• Token value = same as one found in the step 3		
			Observe option indicating increasing values		
	5	Verify	Client displays the received information		
	6	Check	Client sends an ACK		

8.2.4 Observing Resources

NOTE: Steps 4-6 are in a loop.

Interoperability Test Description				
Identifier [.]				
Objective:	Handle reso	urce observ	ation with NON messages	
Configuration:	COAR CEG	01	alloff with NON messages	
Poforoncos		2		
Nelelences.		.∠		
Pre-test conditions:	 Client supports Observe option Server supports Observe option Server offers an observable resource <i>lobs-non</i> changing periodically (e.g. every 5 s) which produces non-confirmable notifications 			
Test Convenses	Description			
Test Sequence:	Step	Туре	Description	
	1	Stimulus	Client is requested to send to the server a non-confirmable	
			GET request with observe option for resource /obs	
	2	Check	The request sent by client contains:	
			• Type = 1 (NON)	
			 Code = 1 (GET) 	
			 Token value = a value generated by the client 	
			 Observe option = empty 	
	3	Check	Server sends a notification containing:	
	(see note)		• Type = 1 (NON)	
	()		 Content-format = the same for all notifications 	
			 Token value – same as one found in the step 2 	
			 Observe option indicating increasing values 	
	1	Verify	Client displays the received information	
NOTE: Steps 3-4 are in a loop				

	Interoperability Test Description				
Identifier:	TD_COAP_	TD_COAP_OBS_03			
Objective:	Stop resource observation				
Configuration:	CoAP_CFG_01				
References:	[4], clause 4	.1 §3			
Pre-test	Client s	upports Obs	serve option		
conditions:	Server	supports Ob	oserve option		
	Server	offers an ob	servable resource /obs changing periodically (e.g. every 5 s)		
	which p	roduces cor	nfirmable notifications		
Test Sequence:	Step	Туре	Description		
	1	Stimulus	Client is requested to send to the server a confirmable GET request with observe option for resource /obs		
	2	Check	The request sent by client contains:		
			 Type = 0 (CON) 		
			• Code = 1 (GET)		
			• Token value = a value generated by the client		
			 Observe option = empty 		
	3	Check	Server sends the response containing:		
			• Type = 2 (ACK)		
			Content-format of the resource /obs		
			 Token value = same as one found in the step 2 		
			Observe option with a sequence number		
	4	Check	Server sends a notification containing:		
	(see		• Type = 0 (CON)		
	note 1)		 Content-format = same as one found in the step 3 		
			 Token value = same as one found in the step 2 		
			Observe option indicating increasing values		
	5	Check	Client displays the received information		
	6	Check	Client sends an ACK		
	7	Stimulus	Client is requested to stop observing the resource /obs on the		
	(see		server		
	note 2)				
	8	Check	Client sends a request containing:		
			• Type = 0 (CON)		
			• Code = 1 (GET)		
			 Token value = a value generated by the client 		
			DOES NOT contain observe option		
	9	Check	Server sends response not containing Observe option		
	10	Verify	Client displays the received information		
	11	Check	Server does not send further response		
	12	Verify	Client does not display updated information		
NOTE 1: Steps 4-	6 are in a loop).			
NOTE 2: Step 7-1	2: Step 7-12 are asynchronous to the loop.				

	Interoperability Test Description			
Identifier:	UL_COAF	2_OBS_04	violantian (Marchan)	
Objective:	Client detection of deregistration (Max-Age)			
Configuration:				
References:	[[4], clause	3.3 94		
Pre-test conditions:	 Client supports Observe option Server supports Observe option Server offers an observable resource <i>lobs</i> changing periodically (e.g. every 5 s) which produces confirmable notifications 			
Test Sequence:	Sten	Туре	Description	
rest Sequence.	1	Stimulus	Client is requested to send to the server a confirmable GET	
	ľ	Sumuus	request with observe option for resource /obs	
	2	Check	The request sent by client contains: • Type = 0 (CON) • Code = 1 (GET) • Token value = a value generated by the client • Observe option = empty	
	3	Check	 Server sends the response containing: Type = 2 (ACK) Content-format of the resource /obs Token value = same as one found in the step 2 Observe option with a sequence number 	
	4 (note 1)	Check	 Server sends a notification containing: Type = 0 (CON) Content-format = same as one found in the step 3 Token value = same as one found in the step 2 Observe option indicating increasing values 	
	5	Verify	Client displays the received information	
	6	Check	Client sends an ACK	
	7 (note 2)	Stimulus	Server is rebooted	
	8	Check	Server does not send notifications	
	9	Verify	Client does not display updated information	
	10	Verify	After Max-Age expiration ⁴ the client internally decides to send another GET request to the server with observe option for resource /obs	
	11	Verify	 Client sends a GET request to the server for resource /obs: Type = 0 (CON) Code = 1 (GET) Token value = a value generated by the client different from the token at step 2 Observe option = empty 	
	12	Check	 Server sends the response containing: Type = 2 (ACK) Content-format of the resource /obs Token value = same as one found in the step 11 Observe option with a sequence number 	
	13 (note 3)	Check	 Server sends a notification containing: Type = 0 (CON) Content-format = same as one found in the step 12 Token value = same as one found in the step 11 Observe option indicating increasing values 	
	14	Verify	Client displays the received information	
	15	Check	Client sends an ACK	
NOTE 1: Steps 4-6 NOTE 2: Steps 7-9 NOTE 3: Steps 13	6 are in a lo 9 are asyncl -15 are in a	op. hronous to the loop.	e loop 4-6.	

IOTE 4: A new registration should be attempted after Max-Age + MAX_LATENCY as recommended by [3]. MAX_LATENCY is defined by [1] and set to 100 seconds.

	Interoperability Test Description					
Identifier:	TD_COAP_	TD_COAP_OBS_05				
Objective:	Server dete	Server detection of deregistration (client OFF)				
Configuration:	CoAP_CFG	CoAP_CFG_01				
References:	[4], clause 4	4.5 §2				
Pre-test	Client s	Client supports Observe option				
conditions:	Server supports Observe option					
	 Server 	offers an obs	ervable resource /obs changing periodically (e.g. every 5 s)			
	which p	produces con	firmable notifications			
Test Sequence:	Step	Туре	Description			
	1	Stimulus	Client is requested to send to the server a confirmable GET			
			request with observe option for resource /obs			
	2	Check	The request sent by client contains:			
			• Type = 0 (CON)			
			• Code = 1 (GET)			
			 Token value = a value generated by the client 			
			 Observe option = empty 			
	3	Check	Server sends the response containing:			
			 Type = 2 (ACK) 			
			 Content-format of the resource /obs 			
			 Token value = same as one found in the step 2 			
			 Observe option with a sequence number 			
	4	Check	Server sends a notification containing:			
	(note 1)		 Type = 0 (CON) 			
			 Content-format = same as one found in the step 3 			
			 Token value = same as one found in the step 2 			
			 Observe option indicating increasing values 			
	5	Check	Client displays the received information			
	6	Check	Client sends an ACK			
	7	Stimulus	Client is switched off			
	(note 2)					
	8	Check	Server's confirmable responses are not acknowledged			
			Server's retransmissions have an updated Observe option			
			value			
	9	Check	Server should keep retransmitting the responses until at least			
			Max-Age seconds after the first un-acknowledged response.			
NOTE 1: Steps 4-	6 are in a loo	op.				
VOTE 2: Steps 7-12 are asynchronous to the loop.						

NOTE 2: Steps 7-12 are asynchronous to the loop.

	Interoperability Test Description					
Identifier:	TD_COAP	TD_COAP_OBS_06				
Objective:	Server det	Server detection of deregistration (explicit RST)				
Configuration:	CoAP_CF	CoAP_CFG_01				
References:	[4], clause	4.2 §5				
Pre-test conditions:	 Client supports Observe option Server supports Observe option Server offers an observable resource <i>lobs</i> changing periodically (e.g. every 5 s) which produces confirmable notifications 					
Test Sequence:	Step	Type	Description			
	1	Stimulus	Client is requested to send to the server a confirmable GET request with observe option for resource /obs			
	2	Check	 The request sent by client contains: Type = 0 (CON) Code = 1 (GET) Token value = a value generated by the client Observe option = empty 			
	3	Check	 Server sends the response containing: Type = 2 (ACK) Content-format of the resource /obs Token value = same as one found in the step 2 Observe option with a sequence number 			
	4 (note 1)	Check	 Server sends a notification containing: Type = 0 (CON) Content-format = same as one found in the step 3 Token value = same as one found in the step 2 Observe option indicating increasing values 			
	5	Check	Client displays the received information			
	6	Check	Client sends an ACK			
	7 (note 2)	Stimulus	Client is rebooted			
	8	Check	 Server is still sending notifications for the request in step 2. Notification contains: Type = 0 (CON) Content-format = same as one found in the step 3 Token value = same as one found in the step 2 Observe option indicating increasing values 			
	9	Verify	Client discards response and does not display information			
	10	Check	Client sends RST to Server			
	11	Verify	Server does not send further response			
	12	Verify	Client does not display further received information			
NOTE 1: Steps 4-	-6 are in a loo					
NOTE 2: Step 7-1	12 are asyncl	nronous to the	e loop.			

		Interop	erability Test Description		
Identifier:	TD_COAP_OBS_07				
Objective:	Server clea	Server cleans the observers list on DELETE			
Configuration:	CoAP_CFG_01				
References:	[4], clause 3	3.2 §4			
Pre-test	Client s	supports Obs	erve option		
conditions:	 Server 	Server supports Observe option			
	 Server 	offers an obs	servable resource /obs changing periodically (e.g. every 5 s)		
	which	produces con	firmable notifications		
Test Sequence:	Step	Туре	Description		
	1	Stimulus	Client is requested to send to the server a confirmable GET		
			request with observe option for resource /obs		
	2	Check	The request sent by client contains:		
			• Type = 0 (CON)		
			• Code = 1 (GET)		
			 Token value = a value generated by the client 		
			Observe option = empty		
	3	Check	Server sends the response containing:		
			• Type = 2 (ACK)		
			 Content-format of the resource /obs 		
			 Token value = same as one found in the step 2 		
			Observe option with a sequence number		
	4	Check	Server sends a notification containing:		
	(note 1)		• Type = 0 (CON)		
			 Content-format = same as one found in the step 3 		
			• Token value = same as one found in the step 2		
			Observe option indicating increasing values		
	5	Check	Client displays the received information		
	6	Check	Client sends an ACK		
	7	Stimulus	Delete the /obs resource of the server (either locally or by		
	(note 2)		having another CoAP client perform a DELETE request)		
	8	Check	Server sends a notification containing:		
			• Type = 0 (CON)		
			• Code = 132 (4.04 NOT FOUND)		
			• I oken value = same as one found in the step 2		
			Observe option indicating increasing values		
	9	Verity	Server does not send further responses		
	10	Verity	Client does not display further received information		
NOTE 1: Steps 4	-6 are in a lo	op.			
NOTE 2: Step 7-10 are asynchronous to the loop.					

	Interoperability Test Description				
Identifier:	TD_COAP_OBS_08				
Objective:	Server cleans the observers list when observed resource content-format changes				
Configuration:	CoAP_CFG_01				
References:	[4], clause	4.2 §3			
Pre-test conditions:	 Client supports Observe option Server supports Observe option Server offers an observable resource <i>lobs</i> changing periodically (e.g. every 5 s) which produces confirmable notifications 				
Test Sequence:	Step	Type	Description		
•	1	Stimulus	Client is requested to send to the server a confirmable GET request with observe option for resource /obs		
	2	Check	 The request sent by client contains: Type = 0 (CON) Code = 1 (GET) Token value = a value generated by the client Observe option = empty 		
	3	Check	Server sends the response containing: • Type = 2 (ACK)		
			 Content-format of the resource /obs Token value = same as one found in the step 2 Observe option with a sequence number 		
	4 (note 1)	Check	 Server sends a notification containing: Type = 0 (CON) Content-format = same as one found in the step 3 Token value = same as one found in the step 2 Observe option indicating increasing values 		
	5	Check	Client displays the received information		
	6	Check	Client sends an ACK		
	7 (note 2)	Stimulus	Update the /obs resource of the server's resource with a new payload having a different Content-Format (either locally or by having another CoAP client perform a DELETE request)		
	8	Check	 Server sends notification containing: Type = 0 (CON) Code = 160 (5.00 INTERNAL SERVER ERROR) Token value = same as one found in the step 2 Observe option indicating increasing values 		
	9	Verify	Server does not send further notifications		
	10	Verify	Client does not display further received information		
NOTE 1: Steps 4-6 are in a loop. NOTE 2: Step 7-10 are asynchronous to the loop.					

		Interop	erability Test Description		
Identifier:	TD_COAP	_OBS_09			
Objective:	Update of	Update of the observed resource			
Configuration:	CoAP_CF	G_01			
References:	[4], clause	4.2 §3			
Pre-test	Client	supports Obs	serve option		
conditions:	 Serve 	r supports Ob	serve option		
	 Serve 	r offers an ob	servable resource /obs changing periodically (e.g. every 5 s)		
	which	produces cor	nfirmable notifications		
	-	-			
Test Sequence:	Step	Туре	Description		
	1	Stimulus	Client is requested to send to the server a confirmable GET		
			request with observe option for resource /obs		
	2	Check	The request sent by client contains:		
			 Type = 0 (CON) 		
			• Code = 1 (GET)		
			 Token value = a value generated by the client 		
			 Observe option = empty 		
	3	Check	Server sends the response containing:		
			 Type = 2 (ACK) 		
			 Content-format of the resource /obs 		
			 Token value = same as one found in the step 2 		
			Observe option with a sequence number		
	4	Check	Server sends a notification containing:		
	(note 1)		 Type = 0 (CON) 		
			 Content-format = same as one found in the step 3 		
			 Token value = same as one found in the step 2 		
			 Observe option indicating increasing values 		
	5	Check	Client displays the received information		
	6	Check	Client sends an ACK		
	7	Stimulus	Update the /obs resource of the server's resource with a new		
	(note 2)		payload having the same Content-Format (either locally or by		
			having another CoAP client perform a DELETE request)		
	8	Check	Server notifications contains:		
	(note 3)		 Type = 0 (CON) 		
			 Content-format = same as one found in the step 3 		
			 Token value = same as one found in the step 2 		
			 Observe option indicating increasing values 		
			 Payload = the new value sent at step 8 		
	9	Verify	Client displays the new value of /obs sent in step 8		
	10	Check	Client sends an ACK		
NOTE 1: Steps 4-6	6 are in a lo	op.			

NOTE 2: Steps 7-9 are asynchronous to the loop 4-6. NOTE 3: Steps 8-10 are in a loop (the same loop at steps 4-6 but /obs is updated).

- ETSI EG 202 337: "Methods for Testing and Specification (MTS); Internet Protocol Testing (IPT); Generic Approach to Interoperability Testing".
- ETSI EG 201 015: "Methods for Testing and Specification (MTS); Standards Engineering Process; A handbook of validation methods".
- CoAP#2 Plugtests technical report :
 <u>http://docbox.etsi.org/M2M/Open/Information/CoAP%20Plugtest%20events/CoAP2%20Plugtests%20report.z</u>
 <u>ip</u>

History

Document history		
V1.1.1	April 2013	Publication