CoAP#4 Plugtests;
London, UK;
7- 9 March 2014

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

Contents

1 Executive Summary

The CoAP#4 Plugtests event was held from 7 to 9 March 2014 in London UK, co-located with the IETF#89 meeting. This event was organized by ETSI with the support of the IPSO Alliance.

Following the 3 previous CoAP Plugtests, ETSI was asked by several participants to hold another interoperability event on CoAP in 2014. The latest changes in the Block and Observe specifications caused some interoperability problems at the CoAP#3 Plugtest. Due to misunderstanding or lack of time, some participants had not implemented the changes.

So, the CoAP community asked ETSI to run another event to enable companies to test the stable drafts, especially Observe and Block. It was also a good opportunity to go beyond CoAP, by proposing that companies test the CoAP security using DTLS and OMA Lightweight M2M, which is based on CoAP and also 6LoWPAN.

This is in line with the standardization work of the oneM2M Partnership Project, where CoAP, DTLS, OMA LWM2M and 6LoWPAN are considered as a key component of the future global standardized M2M architecture.

This event had an excellent participation of 11 companies/organizations providing various CoAP clients and servers, DTLS, 6LoWPAN, OMA LWM2M implementations.

The conclusions are that

- All implementations have been compatible on a basic level
- CoAP standards are clearly mature
- The last changes in the Block and Observe specifications have been tested with success
- The level of Interoperability of OMA LWM2M is excellent, especially for a first event. As the scenarios were basic, it shows a good maturity on basic level. The testing needs now to be extended with more deep test scenarios

Following the conclusion of a technical discussion at the IETF#89 on Friday 7 March during the CoRE WG, the need arose to test the CoAP “Cancel resource observation” method. A new Test Description was written for this purpose. This enabled Plugtests to provide an immediate and consistent input to the IETF CoRE WG (one benefit of holding the event in co-location with IETF#89). This highlights once again the synergy between Plugtest events and other standardization bodies.

2 Introduction

This Plugtests event tested the interoperability of CoAP client and server implementations, DTLS, 6LoWPAN as well as OMA LWM2M client and server implementations

The implementations were connected via both IPv6 and IPV4 test networks.

3 Test documents have been used for the testing:

- A Plugtests guide produced by ETSI containing 56 interoperability tests on CoAP (CoRE, Block, Observe and Link) and DTLS.
- A 6LoWPAN Plugtests Guide, containing 29 Interoperability tests, updated from the ones used at the 6LoWPAN Plugtests, in Berlin in July 2013.
- An Enabler Test Specification of LWM2M produced by OMA, containing 16 interoperability tests

ETSI provided the interoperability tool suite including the wiki, scheduling, test reporting tool and the network infrastructure.

Each day test sessions for IOP assessment were conducted. Each day, a wrap-up meeting was held to discuss the main interoperability points of the day.

3 Base Specifications

The following documents were used as basis for the tests:

[1]	Constrained Application Protocol (CoAP); draft-ietf-core-coap-18
[2]	Core Link Format; RFC 6690
[3]	Observing Resources in CoAP; draft-ietf-core-observe-12
[4]	Blockwise transfers in CoAP; draft-ietf-core-block-14
[5]	Lightweight Machine to Machine Technical Specification: Draft Version 19 February 2014

4 Abbreviations

CoAP	Constrained Application Protocol
NO	Test is recorded as NOT successfully passed.
NA	Test is not applicable.
OK	Test is recorded as successfully passed.
OT	Test is recorded as not being executed due to lack of time.
Test Session	A paring of vendors that test together during a given time slot.
TSR	Test Session Report. Report created during a test session.

5 Participants

The companies which attended the Plugtests event are listed in the table below.

Table 1: organizations/companies participating

#	Company
1	ARM
2	ERICSSON
3	ETH Zurich
4	ETRI
5	HUAWEI
6	RIOT/INRIA
7	TZI / Uni Bremen
8	HITACHI
9	iMinds
10	Open Mobile Alliance
11	Carnegie Mellon University

6 Technical and Project Management

All the information presented in this chapter is an extract of the ETSI event wiki
https://services.plugtests.net/wiki/CoAP-2014/ (access for registered participants only).

6.1 Test Plan

The CoAP test plan containing 56 interoperability tests was developed by ETSI CTI. The coverage of the specifications has been improved regarding the former CoAP Plugtests.

The 6LoWPAN test plan contains 29 interoperability tests covering Frame Format, Neighbor Discovery and Header Compression.

OMA has provided an Enabler Test Specification of LWM2M containing 16 interoperability tests.

During the event preparation, companies had the possibility to review the test plan and to propose additional tests.

The tests were grouped in 3 categories split in 9 groups:

Category	Group	
COAP		Constrained Application Protocol
	CORE	Core protocol
	LINK	Core Link Format
	BLOCK	Blockwise transfers
	OBS	Observing Resources
	DTLS	DTLS
LWM2M		OMA Lightweight M2M
6LoWPAN		6LoWPAN
	FORMAT	6 LoWPAN frame format
	HC	6LoWPAN Header Compression
	ND	6 LoWPAN Neighbor Discovery
	ND_HC	Combined 6LoWPAN HC-ND

The features covered by all tests are listed below:

- CoAP Testing based on updated base specifications (updated since the first CoAP Plugtests event)
  - Constrained Application Protocol (CoAP); draft-ietf-core-coap-18
  - Core Link Format; RFC 6690
  - Observing Resources in CoAP; draft-ietf-core-observe-12
  - Blockwise transfers in CoAP; draft-ietf-core-block-14
- DTLS as profiled in draft-ietf-core-coap-18:
  - Datagram Transport Layer Security Version 1.2, RFC 6347, January 2012
  - AES-CCM Cipher Suites for Transport Layer Security (TLS), RFC 6655
  - AES-CCM ECC Cipher Suites for TLS, draft-mcgrew-tls-aes-ccm-ecc-06
  - Out-of-Band Public Key Validation for Transport Layer Security (TLS), draft-ietf-tls-oob-pubkey-07
- OMA LWM2M
  - OMA-TS-LightweightM2M-V1_0-20131105-D: Lightweight Machine to Machine: Technical Specification

- OMA-ETS-LightweightM2M-V1_0-20140214-D: Enabler Test Specification for Lightweight M2M

Table 2: CoAP Tests

Test Id	Test Summary
TD_COAP_CORE_01	Perform GET transaction (CON mode)
TD_COAP_CORE_02	Perform DELETE transaction (CON mode)
TD_COAP_CORE_03	Perform PUT transaction (CON mode)
TD_COAP_CORE_04	Perform POST transaction (CON mode)
TD_COAP_CORE_05	Perform GET transaction (NON mode)
TD_COAP_CORE_06	Perform DELETE transaction (NON mode)
TD_COAP_CORE_07	Perform PUT transaction (NON mode)
TD_COAP_CORE_08	Perform POST transaction (NON mode))
TD_COAP_CORE_09	Perform GET transaction with separate response (CON mode, no piggyback)
TD_COAP_CORE_10	Perform GET transaction containing non-empty Token option (CON mode)
TD_COAP_CORE_11	Perform GET transaction containing non-empty Token with a separate response (CON mode)
TD_COAP_CORE_12	Perform GET transaction using empty Token (CON mode)
TD_COAP_CORE_13	Perform GET transaction containing several URI-Path options (CON mode)
TD_COAP_CORE_14	Perform GET transaction containing several URI-Query options (CON mode)
TD_COAP_CORE_15	Perform GET transaction (CON mode, piggybacked response) in a lossy context
TD_COAP_CORE_16	Perform GET transaction (CON mode, delayed response) in a lossy context
TD_COAP_CORE_17	Perform GET transaction with a separate response (NON mode)
TD_COAP_CORE_18	Perform POST transaction with responses containing several Location-Path options (CON mode)
TD_COAP_CORE_19	Perform POST transaction with responses containing several Location-Query options (CON mode)
TD_COAP_CORE_20	Perform GET transaction containing the Accept option (CON mode)
TD_COAP_CORE_21	Perform GET transaction containing the ETag option (CON mode)
TD_COAP_CORE_22	Perform GET transaction with responses containing the ETag option and requests containing the If-Match option (CON mode)
TD_COAP_CORE_23	Perform PUT transaction containing the If-None-Match option (CON mode)
TD_COAP_CORE_31	Perform CoAP Ping (CON mode)

Table 3: Link Tests

Test Id	Test Summary
TD_COAP_LINK_01	Access to well-known interface for resource discovery
TD_COAP_LINK_02	Use filtered requests for limiting discovery results
TD_COAP_LINK_03	Handle empty prefix value strings
TD_COAP_LINK_04	Filter discovery results in presence of multiple rt attributes
TD_COAP_LINK_05	Filter discovery results using if attribute and prefix value strings
TD_COAP_LINK_06	Filter discovery results using sz attribute and prefix value strings
TD_COAP_LINK_07	Filter discovery results using href attribute and complete value strings
TD_COAP_LINK_08	Filter discovery results using href attribute and prefix value strings
TD_COAP_LINK_09	Arrange link descriptions hierarchically

Table 5: Block Tests

Test Id	Test Summary
TD_COAP_BLOCK_01	Handle GET blockwise transfer for large resource (early negotiation)
TD_COAP_BLOCK_02	Handle GET blockwise transfer for large resource (late negotiation)
TD_COAP_BLOCK_03	Handle PUT blockwise transfer for large resource
TD_COAP_BLOCK_04	Handle POST blockwise transfer for creating large resource
TD_COAP_BLOCK_05	Handle POST with two-way blockwise transfer
TD_COAP_BLOCK_06	Handle GET blockwise transfer for large resource (early negotiation, 16 byte block size)

Table 6: Observ Tests

Test Id	Test Summary
TD_COAP_OBS_01	Handle resource observation with CON messages
TD_COAP_OBS_02	Handle resource observation with NON messages
TD_COAP_OBS_04	Client detection of deregistration (Max-Age)
TD_COAP_OBS_05	Server detection of deregistration (client OFF)
TD_COAP_OBS_06	Server detection of deregistration (explicit RST)
TD_COAP_OBS_07	Server cleans the observers list on DELETE
TD_COAP_OBS_08	Server cleans the observers list when observed resource content-format changes
TD_COAP_OBS_09	Update of the observed resource
TD_COAP_OBS_10	GET does not cancel resource observation
TD_COAP_OBS_11	Handle resource observation with CON messages (lossy case)
TD_COAP_OBS_12	GET with Observe=1 does cancel resource observation
TD_COAP_OBS_13	Handle observation of large resources (with Block2)
TD_COAP_OBS_14	Handle observation of variable size large resources (with Block2)

Table 7: DTLS

Test Id	Test Summary
TD_COAP_DTLS_01	Basic DTLS PSK (success case)
TD_COAP_DTLS_02	Basic DTLS PSK (failure case — wrong PSK)
TD_COAP_DTLS_03	Lossy DTLS PSK (success case)
TD_COAP_DTLS_04	Basic DTLS RPK (success case)
TD_COAP_DTLS_05	Basic DTLS RPK (client failure case)
TD_COAP_DTLS_06	Basic DTLS RPK (server failure case)
TD_COAP_DTLS_07	Lossy DTLS RPK (success case)

Table 8: LWM2M Tests

Test Id	Test Summary

TD-LWM2M-01	Initial Registration
TD-LWM2M-02	Registration Update
TD-LWM2M-03	Deregistration
TD-LWM2M-04	Registration Update Trigger
TD-LWM2M-05	Querying basic information from the client in Plain Text format
TD-LWM2M-06	Querying basic information from the client in TLV format
TD-LWM2M-07	Querying basic information from the client in JSON format
TD-LWM2M-08	Querying the firmware version from the client
TD-LWM2M-09	Rebooting the device
TD-LWM2M-10	Querying power status of the terminal
TD-LWM2M-11	Firmware update (via COAP)
TD-LWM2M-12	Firmware update (via alternative mechanism)
TD-LWM2M-13	Querying of connectivity parameters
TD-LWM2M-14	Observation and notification of parameter values
TD-LWM2M-15	Cancel observations using “Cancel Observation” operation
TD-LWM2M-16	UDP Channel Security – Pre-shared Key Mode

Table 9: 6LoWPAN Tests

Test Id	Test Summary	Test Group
TD_6LoWPAN_FORMAT_01	Check that EUTs correctly handle uncompressed 6LoWPAN packets (EUI-64 link-local)	FORMAT
TD_6LoWPAN_FORMAT_02	Check that EUTs correctly handle uncompressed 6LoWPAN packets (16-bit link-local)	FORMAT
TD_6LoWPAN_FORMAT_03	Check that EUTs correctly handle uncompressed 6LoWPAN fragmented packets	FORMAT
TD_6LoWPAN_FORMAT_04	Check that EUTs correctly handle maximum size uncompressed 6LoWPAN fragmented packets	FORMAT
TD_6LoWPAN_FORMAT_05	Check that EUTs correctly handle uncompressed 6LoWPAN multicast to all-nodes (16-bit link-local)	FORMAT
TD_6LoWPAN_FORMAT_06	Check that EUTs correctly handle uncompressed 6LoWPAN multicast to all-nodes (EUI-64 link-local)	FORMAT
TD_6LoWPAN_FORMAT_07	Check that EUTs correctly handle uncompressed 6LoWPAN packets (EUI-64 to 16-bit link-local)	FORMAT
TD_6LoWPAN_FORMAT_08	Check that EUTs correctly handle uncompressed 6LoWPAN packets (16-bit to EUI-64 link-local)	FORMAT
TD_6LoWPAN_HC_01	Check that EUTs correctly handle compressed 6LoWPAN packets (EUI-64 link-local, hop limit=64)	HEADER COMPRESSION
TD_6LoWPAN_HC_02	Check that EUTs correctly handle compressed 6LoWPAN packets (16-bit link-local, hop limit=64)	HEADER COMPRESSION
TD_6LoWPAN_HC_03	Check that EUTs correctly handle compressed 6LoWPAN packets (EUI-64 link-local, hop limit=63)	HEADER COMPRESSION
TD_6LoWPAN_HC_04	Check that EUTs correctly handle compressed 6LoWPAN packets (16-bit link-local, hop limit=63)	HEADER COMPRESSION
TD_6LoWPAN_HC_05	Check that EUTs correctly handle compressed UDP packets (EUI-64, server port 5683)	HEADER COMPRESSION
TD_6LoWPAN_HC_06	Check that EUTs correctly handle compressed UDP packets (16-bit, server port 5683)	HEADER COMPRESSION

TD_6LoWPAN_HC_07	Check that EUTs correctly handle compressed UDP packets (EUI-64, server port 61616)	HEADER COMPRESSION
TD_6LoWPAN_HC_08	Check that EUTs correctly handle compressed UDP packets (16-bit, server port 61616)	HEADER COMPRESSION
TD_6LoWPAN_HC_09	Check that EUTs correctly handle compressed 6LoWPAN packets (EUI-64 to 16-bit link-local, hop limit=64)	HEADER COMPRESSION
TD_6LoWPAN_HC_10	Check that EUTs correctly handle compressed 6LoWPAN packets (16-bit to EUI-64 link-local, hop limit=64)	HEADER COMPRESSION
TD_6LoWPAN_ND_01	Check that a host is able to register its global IPv6 address (EUI-64)	NEIGHBOR DISCOVERY
TD_6LoWPAN_ND_02	Check that a host is able to register its global IPv6 address (16-bit)	NEIGHBOR DISCOVERY
TD_6LoWPAN_ND_03	Check Host NUD behavior	NEIGHBOR DISCOVERY
TD_6LoWPAN_ND_04	Check 6LR NUD behavior (ICMP version)	NEIGHBOR DISCOVERY
TD_6LoWPAN_ND_05	Check 6LR NUD behavior (UDP version)	NEIGHBOR DISCOVERY
TD_6LoWPAN_ND_06	Check host behavior under multiple prefixes (EUI-64)	NEIGHBOR DISCOVERY
TD_6LoWPAN_ND_07	Check host behavior under multiple prefixes (16-bit)	NEIGHBOR DISCOVERY
TD_6LoWPAN_ND_HC_01	Check that EUTs make use of context 0 (EUI-64)	COMBINED HC-ND
TD_6LoWPAN_ND_HC_02	Check that EUTs make use of context 0 (16-bit)	COMBINED HC-ND
TD_6LoWPAN_ND_HC_03	Check that EUTs make use of context ≠ 0 (EUI-64)	COMBINED HC-ND
TD_6LoWPAN_ND_HC_04	Check that EUTs make use of context ≠ 0 (16-bit)	COMBINED HC-ND

6.2 Test Scheduling

The preliminary test schedule was developed before the Plugtest and was circulated to all the participants in advance for comments. The initial test schedule allowed for each company to test against a fair number of other companies. Two companies were assigned one test slot which had duration of 3 hours. In this test slot the companies could run tests for the configurations: CompA-Client-CompB- Server and CompA-Server-CompB-Client for CoAP, DTLS and LWM2M and 6LoWPAN. Up to 5 parallel test sessions were planned.

During the test event the test schedule was updated according to the progress of the test sessions. This was done during the daily wrap-up meetings at the end of each day and during face-to-face meetings with the participants.

The figure below shows the final version of the test schedule.

Figure 1: Test Schedule

		Area 1	Area 2	Area 3	Area 4	Area 5
Fri 07	9:30-12:30	Free Testing Free Testing	Free Testing Free Testing	Free Testing Free Testing	Free Testing Free Testing	
	13:30-16:30	Ericsson Ericsson LWM2M	ARM ARM Device	Eth Zurich ETH Zurich device	TZI TZI device	
		Huawei Huawei LWM2M	ETRI ETRI device	RIOT RIOT device	Hitachi Hitachi device	
	16:30-19:30	ARM ARM Device	RIOT RIOT device	iMinds iMinds device	Ericsson Ericsson LWM2M	
		Huawei Huawei device	Hitachi Hitachi device	ETRI ETRI device	TZI TZI LWM2M	
Sat 08	8:00-9:30	Free Testing Free Testing	Free Testing Free Testing	Free Testing Free Testing	Free Testing Free Testing	
	9:30-12:30	Eth Zurich ETH Zurich device	ARM ARM Device	ETRI ETRI device		Ericsson Ericsson LWM2M
		iMinds iMinds device	Hitachi Hitachi device	Huawei Huawei device		ARM ARM LWM2M
	13:30-16:30	ARM ARM LWM2M	Huawei Huawei device	Eth Zurich ETH Zurich device	TZI TZI device	iMinds iMinds device
		Huawei Huawei LWM2M	Hitachi Hitachi device	ARM ARM Device	ETRI ETRI device	RIOT RIOT device
	16:30-19:30	iMinds iMinds device	TZI TZI device	Eth Zurich ETH Zurich device	ETRI ETRI device	ARM ARM LWM2M
		ARM ARM Device	Huawei Huawei device	Hitachi Hitachi device	RIOT RIOT device	TZI TZI LWM2M
Sun 09	8:00-9:30	TZI TZI device	Free Testing Free Testing	Free Testing Free Testing	Free Testing Free Testing	Free Testing Free Testing
		RIOT RIOT device				
	9:30-12:30	iMinds iMinds device	ARM ARM Device	Eth Zurich ETH Zurich device	ETRI ETRI device	
		TZI TZI device	RIOT RIOT device	Huawei Huawei device	Hitachi Hitachi device	
	13:30-16:30	ARM ARM Device	Eth Zurich ETH Zurich device	Huawei Huawei device	iMinds iMinds device	
		TZI TZI device	ETRI ETRI device	RIOT RIOT device	Hitachi Hitachi device	
	16:30-19:30	iMinds iMinds device	Eth Zurich ETH Zurich device	Huawei Huawei LWM2M	Free Testing Free Testing	
		Huawei Huawei device	TZI TZI device	TZI TZI LWM2M		

6.3 Interoperability Test Procedure

Each test was executed in the same manner as listed below:

1) Connect client and server over test network

2) Check connectivity between devices

3) Perform tests according to the Plugtests guide

   a. Check if test runs to completion

   b. Check results from an interoperability point of view:
      Is the intended result visible at the application layer?

4) Result determination and reporting

   a. Result OK: run next test

   b. Result not OK: check monitor tools to identify source of error

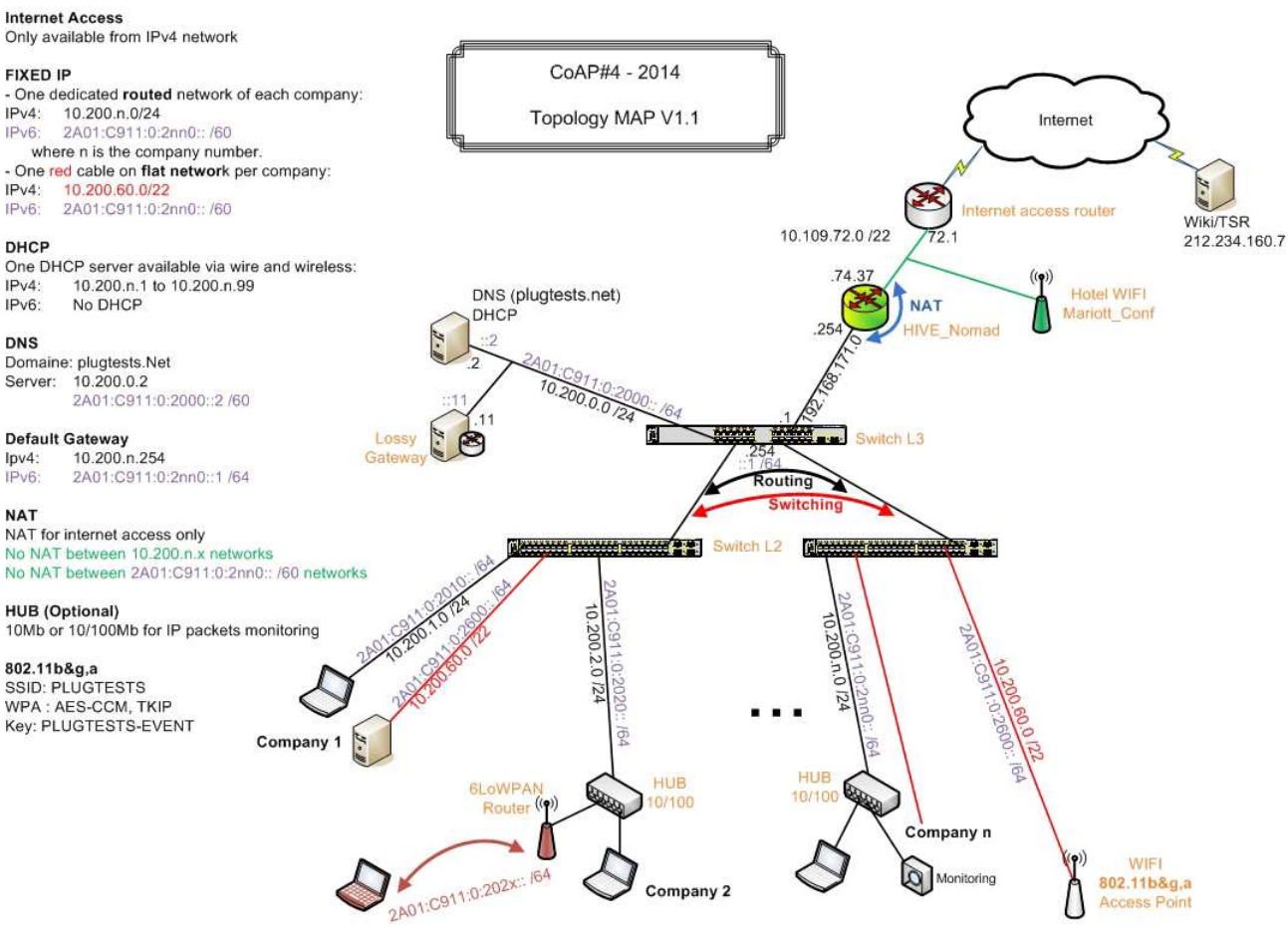
   c. Report results in ETSI Test Reporting Tool

5) Once all tests executed swap client / server roles and run all tests again

6.4 Test Infrastructure

The test infrastructure provided for the Plugtests event is shown below.

Figure 2: Test Network



6.5 Tooling

6.5.1 ETSI Test Reporting Tool

The purpose of the ETSI Test Reporting Tool (TRT) is to provide a means to report the test sessions. It provides statistical overviews of the test results. The graphical information in the latter section on results was created with the ETSI TRT. It also provides a means to create a test schedule (see section 6.2).

6.5.2 Lossy Gateway

The purpose of the UDP lossy gateway is to perform packet loss in CoAP conversations according to the lossy context test descriptions defined in the Plugtests guide.

The configuration of the setup is shown below:

CoAP Client ----- UDP Lossy Gateway ----- CoAP Server

Figure 3: UDP Lossy Gateway Configuration

The UDP lossy gateway assigns one listening port for each CoAP server. Thus the UDP lossy gateway provides for each CoAP server a unique lossy address.

A CoAP client that does lossy context test sends the CoAP message to the lossy address of the specified CoAP server. Then the UDP lossy gateway decides the right destination address according to the UDP socket on which the message was received.

Then the UDP lossy gateway starts a new UDP socket to communicate with the appropriate CoAP server. This UDP socket is also used for forwarding back the CoAP server's responses to the right CoAP client. The server-side communication expires after idling 5mn.

Packet loss is performed at 2 places:

- forwarding CoAP client's message to the CoAP server
- forwarding back CoAP server's message to the CoAP client

The program generates random numbers to decide whether to perform packet loss or not. A 50% packet loss rate was used for the Plugtests.

6.5.3 UDP V4-V6 Gateway

Some participants needed to perform testing between only-V4 devices and only-V6 devices. For enabling such pairings, the UDP lossy gateway has been used with a loss = 0 in the setting.

6.5.4 Pre-Testing

Prior to the event, 3 companies had posted on the wiki the addresses of CoAP servers, in order to enable the participants to perform pre-testing. The feedback we received is that it has been appreciated and helpful for preparing the event.

7 Achieved Results

The achieved results show that all implementations have been compatible on a basic level, i.e. sent data could be decoded and interpreted properly by receivers and a vast majority of equipment performed well.

7.1 Overall CoAP Results

Due to NDA constraints, it is not possible to provide detailed results.

The figure below shows the overall result of CoAP tests, gathering the CoRE, Block, Link and Observ tests. In total more than 1367 tests were executed.

The execution rate of 71.6% (number of tests indeed performed compared to number of tests proposed) is an excellent result, especially considering the high number of test scenarios provided in such a short event. Each test session lasted 3 hours which is quite short as most of the companies had several devices (both client and servers) which of course increased considerably the number of possible pairing combinations. Globally the feedback that the participants gave is that the testing was very dense.

In addition to the test schedule defined, ETSI gave participants the possibility to add further test sessions early in the morning in addition to the scheduled ones, to allow participants to re-run the tests or complete their testing.

98.5% of the test verdicts were OK which shows definitely a very high level of maturity of the implementations.

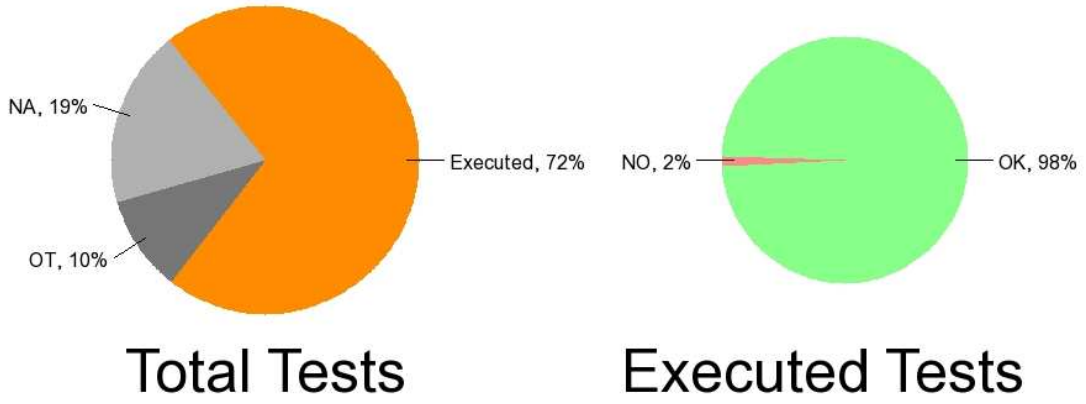


Figure 4: Overall CoAP Results

	Interoperability		Not Executed		Totals	
Group	**OK**	**NO**	**NA**	**OT**	**Run**	**Results**
CoRE	844 (99.3%)	6 (0.7%)	16 (1.8%)	46 (5.0%)	850 (93.2%)	912
BLOCK	117 (95.1%)	6 (4.9%)	84 (36.8%)	21 (9.2%)	123 (53.9%)	228
LINK	219 (96.5%)	8 (3.5%)	89 (26.0%)	26 (7.6%)	227 (66.4%)	342
OBS	166 (99.4%)	1 (0.6%)	165 (38.6%)	96 (22.4%)	167 (39.0%)	428

7.2 Results of CoRE tests

There were 24 test scenarios defined in the test plan. In total 850 tests were executed with a success rate of **99.3%.**

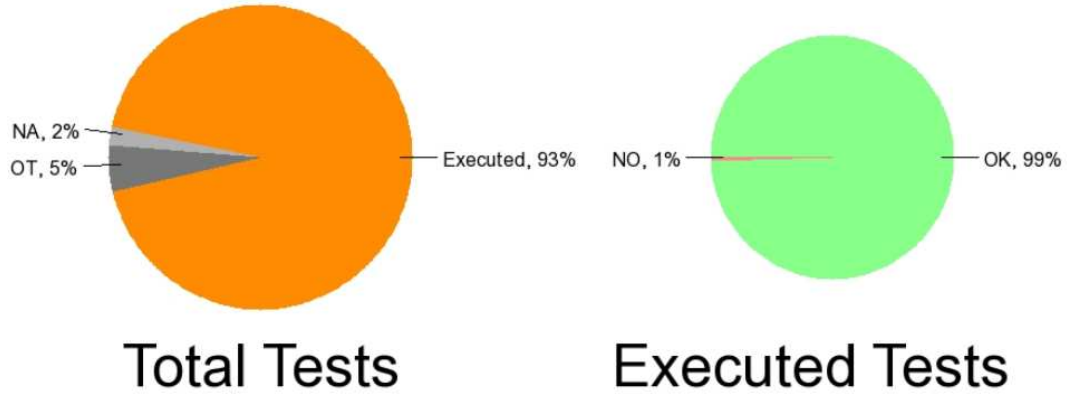


Figure 5: Results of CoRE tests

7.3 Results of Block tests

There were 6 Block test scenarios defined which were to be executed per session. In total 123 tests were executed with a success rate of **95.1%.**

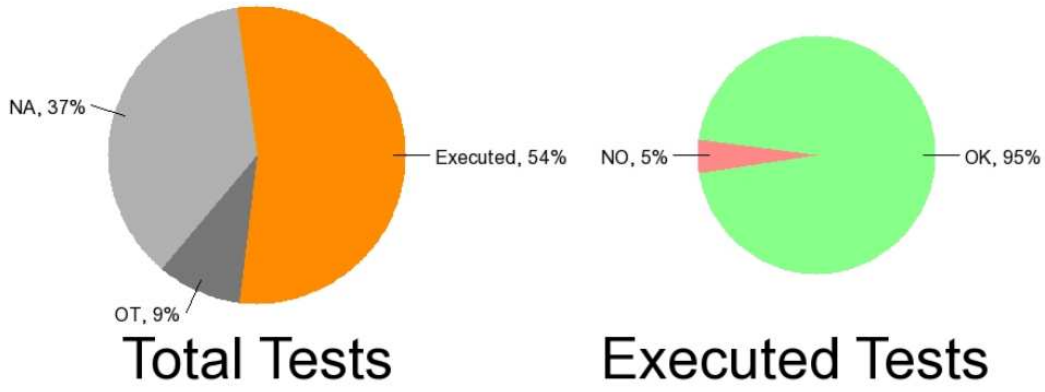


Figure 6: Results of Block tests

7.4 Results of Link tests

There were 9 Link test scenarios defined which were to be executed per session. In total 227 tests were executed with a success rate of **96.5%.**

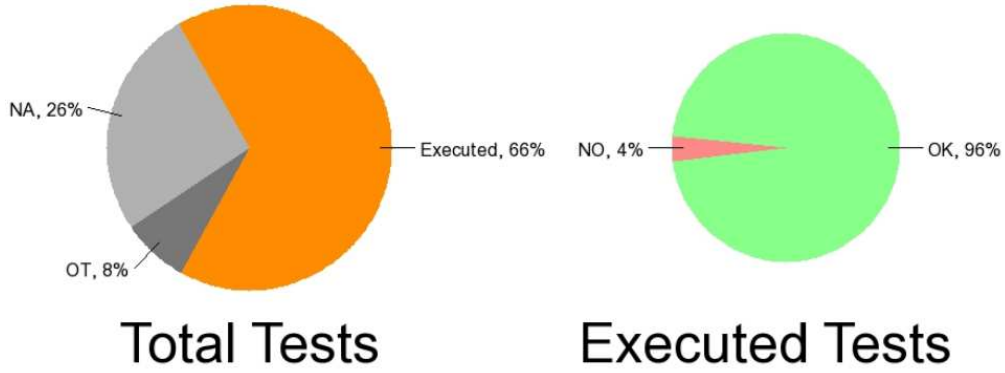


Figure 7: Results of Link tests

7.5 Results of Observe tests

There were 13 Observe test scenarios defined which were to be executed per session. In total 167 tests were executed with a success rate of **99.4%**

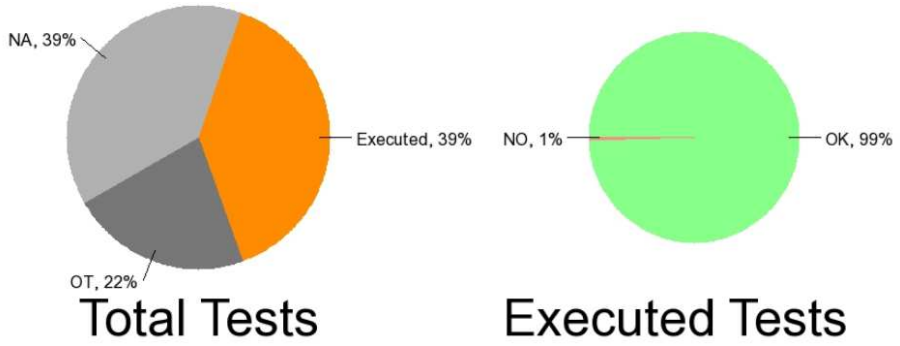


Figure 8: Results of Observ tests

7.6 Results of DTLS tests

There were 7 DTLS test scenarios defined which were to be executed per session. Some tests have been informally performed but no result were reported in the online tool.

The absence of DTLS testing could be explained mainly by the lack of implementation supporting it and also by the priority given by participants to CoAP testing which was the main scope of the event, and also considering the test session lasted only 3 hours each.

7.7 Results of OMA LWM2M tests

There were 16 LWM2M test scenarios defined which were to be executed per session. In total 42 were executed with a success rate of 85.7%.

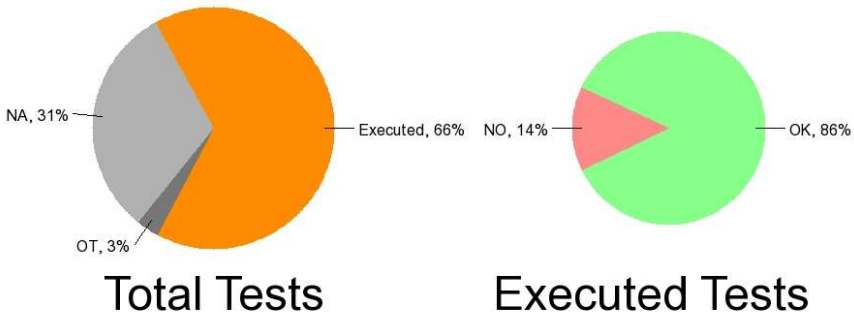


Figure 7: Results of OMA LWM2M tests

7.8 Results of 6LoWPAN tests

There were 29 6LoWPAN test scenarios defined which were to be executed per session. In total 3 were recorded with a success rate of 33.3%. With so few tests performed, the success rate is of course irrelevant.

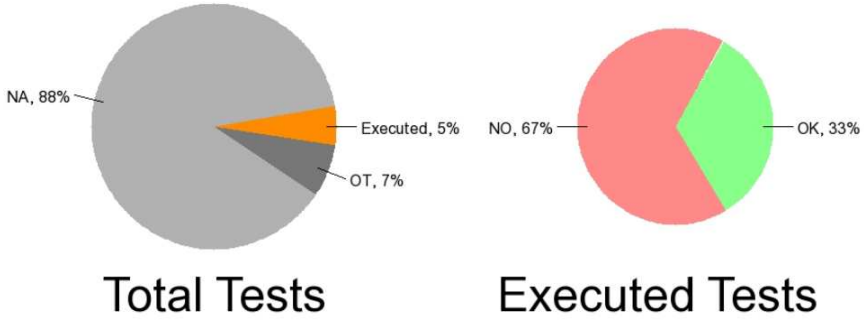


Figure 7: Results of 6LoWPAN tests

	Interoperability		Not Executed		Totals	
Group	**OK**	**NO**	**NA**	**OT**	**Run**	**Results**
Overall results(4 Groups)	1 (33.3%)	2 (66.7%)	51 (87.9%)	4 (6.9%)	3 (5.2%)	58
COMBINED HC-ND	0 (0.0%)	0 (0.0%)	8 (100.0%)	0 (0.0%)	0 (0.0%)	8
FORMAT	1 (33.3%)	2 (66.7%)	12 (75.0%)	1 (6.3%)	3 (18.8%)	16
HEADER COMPRESSION	0 (0.0%)	0 (0.0%)	17 (85.0%)	3 (15.0%)	0 (0.0%)	20
NEIGHBOR DISCOVERY	0 (0.0%)	0 (0.0%)	14 (100.0%)	0 (0.0%)	0 (0.0%)	14

8 Summary of Wrap Up Sessions

8.1 IOP

Following completion of a long-standing technical discussion during the CoRE Working Group meeting at the IETF#89 on Friday 7 March, the urgent need for further testing was raised.

This led to a new test description being written immediately (TD_COAP_OBS_12) and it was all hands on deck to ensure that the tests could be carried out just 2 days later during this event. Between the participants that had the means to react so quickly, 100 % interoperability was achieved on this test. The results were provided to the IETF CoRE WG at the end of the last day of testing.

This highlights once again the synergy between Plugtests event and standardization bodies and shows the advantage of having the event so closely linked to standardization meetings.

8.2 Test Spec Issues

More clarifications were added to the CORE tests about requiring a Content-Format option for PUT and POST requests with non-empty payload; this has been amended during the event. TD_COAP_OBS_08 was amended to use 4.06 Not Acceptable when content-format changes; TD_COAP_OBS_09 incorrectly showed a DELETE where a PUT was required for updating a resource. A total number of six further clarifications, some of which clarify more than one test, was developed based on participant feedback during the event.

Apart from the newly developed TD_COAP_OBS_12 (see Section 8.1), new test descriptions were also developed for the combination of Observe and Block2 options: TD_COAP_OBS_13 tests the basic combination, TD_COAP_OBS_14 also tests the case where the size of the resource representation varies enough to fade in and out the Block2 option during the course of an observation relationship.

8.3 Base Specification Issues

As of March 2014, still only the PSK tests for DTLS have all protocol numbers defined by IANA. For the RPK tests, while recently numbers were assigned for the TLS extensions needed, the test specification had to invent a protocol number for the Cipher Suite to enable interoperability. (The last-minute changes in this space may have contributed to a lack of recorded tests.) This is expected to be remedied only once the remaining draft Cipher Suite specification has been processed by IANA.

As discussed in Section 8.1, the testing the “observation cancellation method” discussed at the IETF in London”, supports the conclusion reached during the meeting.

With respect to the BLOCK testing, the generally high interoperability results of this Plugtest could not be achieved for all the test cases involving the Block1 option in a POST request. While no technical issues were identified with the option as defined, this may point out the need for editorial improvements in the Block specification, which is nearing second working group last call in the IETF.

Annex A CoAP Interoperability Test Specification

The CoAP Interoperability Test Specification, which forms parts of the present technical report, is contained in the file IoT_CoAP4_TestSpecification_005.pdf.

History

Document history		
V0.0.1	March 2014	Initial version
V1.0.0	March	Final version