ETSI TS 103 596-3 V1.1.1 (2021-05)

TECHNICAL SPECIFICATION

Methods for Testing and Specification (MTS); Test Specification for CoAP; Part 3: Performance Tests

Reference
DTS/MTS-TSTCoAP-3

Keywords
performance, 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 - APE 7112B
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° w061004871

Important notice

The present document can be downloaded from:
http://www.etsi.org/standards-search

The present document may be made available in electronic versions and/or in print. The content of any electronic and/or print versions of the present document shall not be modified without the prior written authorization of ETSI. In case of any existing or perceived difference in contents between such versions and/or in print, the prevailing version of an ETSI deliverable is the one made publicly available in PDF format at www.etsi.org/deliver.

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 https://portal.etsi.org/TB/ETSIDeliverableStatus.aspx

If you find errors in the present document, please send your comment to one of the following services:
https://portal.etsi.org/People/CommiteeSupportStaff.aspx

Notice of disclaimer & limitation of liability

The information provided in the present deliverable is directed solely to professionals who have the appropriate degree of experience to understand and interpret its content in accordance with generally accepted engineering or other professional standard and applicable regulations.
No recommendation as to products and services or vendors is made or should be implied.
No representation or warranty is made that this deliverable is technically accurate or sufficient or conforms to any law and/or governmental rule and/or regulation and further, no representation or warranty is made of merchantability or fitness for any particular purpose or against infringement of intellectual property rights.
In no event shall ETSI be held liable for loss of profits or any other incidental or consequential damages.

Any software contained in this deliverable is provided "AS IS" with no warranties, express or implied, including but not limited to, the warranties of merchantability, fitness for a particular purpose and non-infringement of intellectual property rights and ETSI shall not be held liable in any event for any damages whatsoever (including, without limitation, damages for loss of profits, business interruption, loss of information, or any other pecuniary loss) arising out of or related to the use of or inability to use the software.

Copyright Notification

Contents

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to normative deliverables may have been declared to ETSI. The declarations pertaining to these essential IPRs, if any, are 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 (https://ipr.etsi.org/).

Pursuant to the ETSI Directives including the ETSI IPR Policy, no investigation regarding the essentiality of IPRs, 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.

Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

DECT™, **PLUGTESTS™**, **UMTS™** and the ETSI logo are trademarks of ETSI registered for the benefit of its Members. **3GPP™** and **LTE™** are trademarks of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners. **oneM2M™** logo is a trademark of ETSI registered for the benefit of its Members and of the oneM2M Partners. **GSM**® and the GSM logo are trademarks registered and owned by the GSM Association.

Foreword

This Technical Specification (TS) has been produced by ETSI Technical Committee Methods for Testing and Specification (MTS).

The present document is part 3 of a multi-part deliverable covering the Constrained Application Protocol (CoAP), as identified below:

Part 1: "Conformance Tests";

Part 2: "Security Tests";

Part 3: "Performance Tests".

Modal verbs terminology

In the present document "**shall**", "**shall not**", "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the ETSI Drafting Rules (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

Introduction

The present document provides an introduction and possible test specification, i.e. an overall test suite structure and catalogue of performance test purposes for the Constrained Application Protocol (CoAP) protocol. It will be a reference base for both client side test campaigns and server side test campaigns addressing the performance issues.

1 Scope

The present document provides a test specification, i.e. an overall test suite structure and catalogue of test purposes for the Constrained Application Protocol (CoAP) protocol. It will be a reference base for both client side test campaigns and server side test campaigns addressing the performance issues.

2 References

2.1 Normative references

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

Referenced documents which are not found to be publicly available in the expected location might be found at https://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.

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

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

[2] ETSI TS 103 596-1: "Methods for Testing and Specification (MTS); Test Specification for CoAP; Part 1: Conformance Tests".

[3] IETF RFC 8323: "CoAP (Constrained Application Protocol) over TCP, TLS, and WebSockets".

[4] ETSI ES 203 119-4: "Methods for Testing and Specification (MTS); The Test Description Language (TDL); Part 4: Structured Test Objective Specification (Extension)".

2.2 Informative references

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

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

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

[i.1] IETF RFC 2544: "Benchmarking Methodology for Network Interconnect Devices".

[i.2] ETSI TR 101 577: "Methods for Testing and Specifications (MTS); Performance Testing of Distributed Systems; Concepts and Terminology".

3 Definition of terms, symbols and abbreviations

3.1 Terms

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

acknowledgement message: message which acknowledges that a specific Confirmable message arrived as defined in IETF RFC 7252 [1]

NOTE: By itself, an Acknowledgement message does not indicate success or failure of any request encapsulated in the Confirmable message, but the Acknowledgement message may also carry a Piggybacked Response.

benchmark test: procedure by which a test system interacts with a System Under Test to measure its behaviour and produce a benchmark report

benchmark test report: document generated at the conclusion of a test procedure containing the metrics measured during the test

client: originating endpoint of a request; the destination endpoint of a response as defined in IETF RFC 7252 [1]

CoAP-to-CoAP proxy: proxy that maps from a CoAP request to a CoAP request, i.e. uses the CoAP protocol both on the server and the client side

NOTE: Contrast to cross-proxy.

confirmable message: some messages that require an acknowledgement as defined in IETF RFC 7252 [1]

NOTE: These messages are called "Confirmable". When no packets are lost, each Confirmable message elicits exactly one return message of type Acknowledgement or type Reset.

conformance: extent to which an implementation of a standard satisfies the requirements expressed in that standard

conformance testing: process to verify to what extent the IUT conforms to the standard

content-format: combination of an Internet media type, potentially with specific parameters given, and a content-coding (which is often the identity content-coding), identified by a numeric identifier defined by the "CoAP Content-Formats" registry as defined in IETF RFC 7252 [1]

NOTE: When the focus is less on the numeric identifier than on the combination of these characteristics of a resource representation, this is also called "representation format".

critical option: option that would need to be understood by the endpoint ultimately receiving the message in order to properly process the message as defined in IETF RFC 7252 [1]

NOTE: The implementation of critical options is, as the name "Option" implies, generally optional: unsupported critical options lead to an error response or summary rejection of the message.

cross-proxy: cross-protocol proxy, or "cross-proxy" for short, proxy that translates between different protocols, such as a CoAP-to-HTTP proxy or an HTTP-to-CoAP proxy

NOTE: While the present document makes very specific demands of CoAP-to-CoAP proxies, there is more variation possible in cross-proxies.

Design Objective Capacity (DOC): largest load an SUT can sustain while not exceeding design objectives defined for a use-case

elective option: option that is intended to be ignored by an endpoint that does not understand it as defined in IETF RFC 7252 [1]

NOTE: Processing the message even without understanding the option is acceptable.

empty message: message with a Code of 0.00; neither a request nor a response as defined in IETF RFC 7252 [1]

NOTE: An Empty message only contains the 4-byte header.

endpoint: entity participating in the CoAP protocol as defined in IETF RFC 7252 [1]

NOTE: Colloquially, an endpoint lives on a "Node", although "Host" would be more consistent with Internet standards usage, and is further identified by transport-layer multiplexing information that can include a UDP port number and a security association.

forward-proxy: endpoint selected by a client, usually via local configuration rules, to perform requests on behalf of the client, doing any necessary translations as defined in IETF RFC 7252 [1]

NOTE: Some translations are minimal, such as for proxy requests for "CoAP" URIs, whereas other requests might require translation to and from entirely different application-layer protocols.

intermediary: CoAP endpoint that acts both as a server and as a client towards an origin server (possibly via further intermediaries) as defined in IETF RFC 7252 [1]

NOTE: A common form of an intermediary is a proxy; several classes of such proxies are discussed in the present document.

non-confirmable message: As defined in IETF RFC 7252 [1], some other messages do not require an acknowledgement. This is particularly true for messages that are repeated regularly for application requirements, such as repeated readings from a sensor.

origin server: server on which a given resource resides or is to be created as defined in IETF RFC 7252 [1]

parameter: attribute of a SUT, test system, system load, or traffic set whose value is set externally and prior to a benchmark test, and whose value affects the behaviour of the benchmark test

piggybacked response: included right in a CoAP Acknowledgement (ACK) message that is sent to acknowledge receipt of the Request for this Response as defined in IETF RFC 7252 [1]

proxy: intermediary that mainly is concerned with forwarding requests and relaying back responses, possibly performing caching, namespace translation, or protocol translation in the process as defined in IETF RFC 7252 [1]

NOTE: As opposed to intermediaries in the general sense, proxies generally do not implement specific application semantics. Based on the position in the overall structure of the request forwarding, there are two common forms of proxy: forward-proxy and reverse-proxy. In some cases, a single endpoint might act as an origin server, forward-proxy, or reverse-proxy, switching behaviour based on the nature of each request.

recipient: destination endpoint of a message as defined in IETF RFC 7252 [1]

NOTE: When the aspect of identification of the specific recipient is in focus, also "destination endpoint".

reset message: a specific message (Confirmable or Non-confirmable) is received, but some context is missing to properly process it as defined in IETF RFC 7252 [1]

NOTE: This condition is usually caused when the receiving node has rebooted and has forgotten some state that would be required to interpret the message. Provoking a Reset message (e.g. by sending an Empty Confirmable message) is also useful as an inexpensive check of the liveness of an endpoint ("CoAP ping").

resource discovery: process where a CoAP client queries a server for its list of hosted resources as defined in IETF RFC 7252 [1]

reverse-proxy: endpoint that stands in for one or more other server(s) and satisfies requests on behalf of these, doing any necessary translations as defined in IETF RFC 7252 [1]

NOTE: Unlike a forward-proxy, the client may not be aware that it is communicating with a reverse-proxy; a reverse-proxy receives requests as if it were the origin server for the target resource.

safe-to-forward option: option that is intended to be safe for forwarding by a proxy that does not understand it as defined in IETF RFC 7252 [1]

NOTE: Forwarding the message even without understanding the option is acceptable.

sender: originating endpoint of a message as defined in IETF RFC 7252 [1]

NOTE: When the aspect of identification of the specific sender is in focus, also "source endpoint".

separate response: when a Confirmable message carrying a request is acknowledged with an Empty message (e.g. because the server does not have the answer right away), a Separate Response is sent in a separate message exchange as defined in IETF RFC 7252 [1]

server: destination endpoint of a request; the originating endpoint of a response as defined in IETF RFC 7252 [1]

test scenario: specific path through a use-case, whose implementation by a test system creates a system load

test suite structure: document defining (hierarchical) grouping of test cases according to some rules

traffic-time profile: evolution of the average scenario over a time interval

unsafe option: option that would need to be understood by a proxy receiving the message in order to safely forward the message as defined in IETF RFC 7252 [1]

NOTE: Not every critical option is an unsafe option.

use case: description of a goal that a user has in interacting with a system, the various actors and the SUT

3.2 Symbols

Void.

3.3 Abbreviations

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

CoAP	Constrained Application Protocol
CPU	Central Processing Unit
DOC	Design Objective Capacity
GTW	Gateway
HTTP	Hyper Text Transfer Protocol
IP	Internet Protocol
IUT	Implementation Under Test
KPI	Key Performance Indicator
LAN	Local Area Network
MTS	Methods for Testing and Specifications
NoS	Number of Subscribers
NoC	Number of Clients
OS	Operating System
PICS	Protocol Implementation Conformance Statement
PING	Packet Internet Groper

NOTE: Send a packet to a computer and wait for its return.

PONG	Ping response packet
RAM	Random Access Memory
SSD	Solid State Drive
SoC	System on a Chip
SUT	System Under Test
TCP	Tranmission Control Protol
TDL	Test Description Language
TDL-TO	Test Description Language - Test Objectives
TP	Test Purpose
TS	Test System
UDP	User Datagram Protocol
URI	Unified Resource Identifier
WLF	WorkLoad Factor

4 Performance metrics

4.0 Introduction

The performance metrics specified herein pertain to the specifics of a CoAP IUT. As such, the objective is to use these metrics in order to determine how well the CoAP component (be it client or server) is performing its functions. As CoAP is a transport protocol, the metrics will be focused on how fast, reliable and efficient the transport is handled. The metrics are designed to fit this purpose while covering multiple use-case scenarios. Following below are the specific messages of the CoAP protocol as defined in IETF RFC 8323 [3] for which the performance metrics are defined.

4.1 Concepts

4.1.1 Concepts introduction

For measuring performance of a given Test System (TS), a comprehensive description of the test environment is required. This includes but it is not limited to:

- TS hardware infrastructure: resource specification, type, capacity and distribution.
- Test environment type and resources (virtualisation technology, allocated resources).
- Measurement equipment hardware/software infrastructure, measurement probe distribution/placement, clock synchronization precision, allocated resources.
- Communication infrastructure: transport network specification, number of switches/hops between TS components, bandwidth capacity.

Additional to the specific characteristics of the SUT, the CoAP protocol [3] specifies sessions as stateful interactions between clients and servers. Because of this, additional performance session-based metrics are considered.

4.1.2 Measurement Preliminary Considerations

In order for the collected measurement data to be useful, special consideration needs to be given to the TS setup. Given that the performance evaluation is targeting one or several IUTs same TS setup characteristics are required in order for the evaluation results to allow valid comparisons between them. Some of the characteristics may refer to infrastructure, hardware, physical or virtual resources as well as network connectivity resources.

4.2 Measurement Methodology

4.2.0 Introduction

This clause presents the test methodology for CoAP performance evaluation. From the performance perspective, all measurable metrics related to the protocol should be considered. Although not exhaustive, these metrics can be categorized as follows:

Powerfulness metrics as defined in ETSI TR 101 577 [i.2] include 3 sub categories: Responsiveness, Capacity and Scalability. From the Responsiveness category the response time, roundtrip time and latency time metrics are used. From the Capacity category the arrival capacity, peak capacity, in progress capacity, streaming capacity and Throughput capacity metrics are used. From the Scalability category the scaling capacity metric is used.

Reliability metrics as defined in ETSI TR 101 577 [i.2] include 6 sub categories: Quality-of-Service, Stability, Availability, Robustness, Recovery, and Correctness. The Quality of Service sub category refers to well defined requirements which may include acceptable values or ranges for metrics from other categories. Stability refers to the capacity of the System to deliver acceptable performance over time. From the Availability sub category, the logical availability metric is used. From the Robustness sub category, the service capacity reduction and service responsiveness deterioration metrics are used. From the Recovery sub category, the service restart characteristics metric is used. Correctness metrics cover the ability of delivering correctly processed requests under high or odd load conditions.

Efficiency metrics as defined in ETSI TR 101 577 [i.2] cover resource utilization. The metrics cover the characteristics of resource usage, linearity, scalability and bottleneck. The efficiency metrics used in the present document are referring to the service level and not covering the platform level.

4.2.1 Metric Post-processing

The collection of metric values from a SUT is performed by multiple agents and/or directly by the IUT. Often a better insight into the IUT performance is gained by post-processing these values in order to get more meaningful results. To this scope, the data samples can be aggregates over time intervals in the experiment. From such common practices, the following are used for the metrics listed in this clause:

- Mean Average: $\frac{1}{n}\sum_1^n x_i$, where n is the number of samples and x is a sample value.
- Standard deviation: $\sqrt{\frac{1}{n}\sum_1^n (x_i - \bar{x}_i)^2}$, where n is the number of samples, x is a sample value and $\bar{x}$ is the mean average.
- Minimum: min(x_i), the smallest sample value, relative to the rest of the samples.
- Maximum: max(x_i), the greatest sample value, relative to the rest of the samples.

4.2.2 Message Types

Table 1 contains the set control packet messages specified by the CoAP standard [1].

Table 1: Message Types

Control Packet Name	Description	Client -> Server	Server -> Client	Payload
GET	Retrieves a representation for the information that currently corresponds to the resource identified by the request URI.	✓		Required
POST	Requests that the representation enclosed in the request be processed.	✓		None
PUT	Requests that the resource identified by the request URI be updated or created with the enclosed representation.	✓		Required
DELETE	Requests that the resource identified by the request URI be deleted.	✓		None
Success 2.xx	This class of Response Code indicates that the clients request was successfully received, understood, and accepted.		✓	None
2.01 Created	Like HTTP 201 "Created", but only used in response to POST and PUT requests.		✓	Optional
2.02 Deleted	This Response Code is like HTTP 204 "No Content" but only used in response to requests that cause the resource to cease being available, such as DELETE and, in certain circumstances, POST.		✓	Optional
2.03 Valid	This Response Code is related to HTTP 304 "Not Modified" but only used to indicate that the response identified by the entity-tag identified by the included ETag Option is valid.		✓	None
2.04 Changed	This Response Code is like HTTP 204 "No Content" but only used in response to POST and PUT requests.		✓	Optional
2.05 Content	This Response Code is like HTTP 200 "OK" but only used in response to GET requests.		✓	Required
Client Error 4.xx	This class of Response Code is intended for cases in which the client seems to have erred. These Response Codes are applicable to any request method.		✓	Optional
4.00 Bad Request	This Response Code is Like HTTP 400 "Bad Request".		✓	None

Control Packet Name	Description	Client -> Server	Server -> Client	Payload
4.01 Unauthorized	The client is not authorized to perform the requested action.		✓	None
4.02 Bad Option	The request could not be understood by the server due to one or more unrecognized or malformed options.		✓	None
4.03 Forbidden	This Response Code is like HTTP 403 "Forbidden".		✓	None
4.04 Not Found	This Response Code is like HTTP 404 "Not Found".		✓	None
4.05 Method Not Allowed	This Response Code is like HTTP 405 "Method Not Allowed" but with no parallel to the "Allow" header field.		✓	None
4.06 Not Acceptable	This Response Code is like HTTP 406 "Not Acceptable", but with no response entity.		✓	None
4.12 Precondition Failed	This Response Code is like HTTP 412 "Precondition Failed".		✓	None
4.13 Request Entity Too Large	This Response Code is like HTTP 413 "Request Entity Too Large".		✓	None
4.15 Unsupported Content-Format	This Response Code is like HTTP 415 "Unsupported Media Type".		✓	None
Server Error 5.xx	This class of Response Code indicates cases in which the server is aware that it has erred or is incapable of performing the request. These Response Codes are applicable to any request method.		✓	Optional
5.00 Internal Server Error	This Response Code is like HTTP 500 "Internal Server Error".		✓	Optional
5.01 Not Implemented	This Response Code is like HTTP 501 "Not Implemented".		✓	Optional
5.02 Bad Gateway	This Response Code is like HTTP 502 "Bad Gateway".		✓	Optional
5.03 Service Unavailable	This Response Code is like HTTP 503 "Service Unavailable" but uses the Max-Age Option in place of the "Retry-After" header field to indicate the number of seconds after which to retry.		✓	Optional
5.04 Gateway Timeout	This Response Code is like HTTP 504 "Gateway Timeout".		✓	Optional
5.05 Proxying Not Supported	The server is unable or unwilling to act as a forward-proxy for the URI specified in the Proxy-Uri Option or using Proxy-Scheme.		✓	Optional

4.2.3 Test parameters

The benchmark test parameters are used to control the behaviour of the test script. The data elements required to configure the test system are listed in table 2.

Table 2 is a non-exhaustive list of test parameters defined for the benchmark standard. The list is expected to grow over time, as additional subsystems and system configurations are developed.

Table 2: Test parameters

Parameter	Description
Duration	Amount of time that a system load is presented to a SUT
Type of call	Type of messages contained within a workload
NoC	number of clients generating or subscribing to data/control traffic
NoS	Number of servers handling data/control traffic
Transport interface	Underlying transport interface
WLF for GTW	Work load factor for gateway expressed in number messages received per second, by type of message
Payload	Size of the data in Bytes carried within a message
Monitoring Window(s)	The time interval window for which the monitored metrics are recorded. This reflects the measuring accuracy (e.g. per second, minute, hour, etc.)
Validation threshold(s)	The specific metric thresholds used for validating whether a system performs at specifications

Table 3: Test output

Metric	Description
Minimum call duration	The minimum duration of a successful message request/response interaction within a Monitoring Window
Maximum call duration	The maximum duration of a successful message request/response interaction within a Monitoring Window
Average call duration	The average duration of a successful message request/response interaction within a Monitoring Window
Total number of calls	The total number of workload specified request/response type operations executed during the test
Success rate	Percentage number of successful workload operations relative to the total workload operations
Error rate	Percentage number of failed workload operations relative to the total workload operations
Requests processed per time unit	This metric reflects the average number of successfully processed requests per preferred time unit (second/minute/etc.)

4.2.4 Operation Message Flows

The IUT will be evaluated based on the metric values obtain as a result of the service operations using the messages described in clause 4.2.2. The set of messages exchanged triggered by the initial client message are further referred as operations. For the tests, the metrics use operations rather than specific messages because it is easier to handle the measurements. If specific test system network measurements are available, by subtracting the measured network delayed from the duration, the operation processing time can be deducted.

1) GET: This section describes the CoAP operation types and message sequences required for test execution using a Client GET example.

Preconditions:

- Client, Server
- TCP/UDP connection between Client and Server established

Operation sequence:

- Client sends GET message
- Client receives SUCCESS (Content) message

Measurement: Time period expressed in milliseconds between the moment client forwards the GET Message and the moment Client receives SUCCESS message from server.