



**HORIZON 2020**  
**Information and Communication Technologies**  
**Integrating experiments and facilities in FIRE+**

**Complementary report to  
deliverable D2.6  
Online interop test core enablers  
final iteration**

**Grant Agreement number: 687884**

**Project acronym: F-Interop**

**Project title: FIRE+ online interoperability and performance test tools to support emerging technologies from research to standardization and market launch  
The standards and innovations accelerating tool**

**Type of action: Research and Innovation Action (RIA)**

**Project website address: [www.finterop.eu/](http://www.finterop.eu/)**

**Due date of deliverable: M33**

**Dissemination level: PU**

*This deliverable has been written in the context of the Horizon 2020 European research project F-Interop, which is supported by the European Commission and the Swiss State Secretariat for Education, Research and Innovation. The opinions expressed and arguments employed do not engage the supporting parties.*



## Document properties

<b>Responsible partner</b>	INRIA
<b>Author(s)/editor(s)</b>	Federico Sismondi, César Viho
<b>Version</b>	1.0
<b>Keywords</b>	Interoperability Testing, Remote Testing, Online, Platform, Testing components, Test enablers

## Abstract

This short report corresponds to the deliverable D2.6 – Online Interoperability test core enablers final iteration. It provides complementary information to the deliverable D.2.7 which was delivered in the shape of a source code package along with a video demonstration of the interoperability tool and enablers. It describes the advancement up to the final version of the F-Interop online remote interoperability framework and the key enablers needed.

# Table of Contents

---

<b>Table of Contents</b> .....	<b>3</b>
<b>List of Acronyms</b> .....	<b>4</b>
<b>1 Introduction</b> .....	<b>5</b>
<b>1.1 About F-Interop</b> .....	<b>5</b>
<b>1.2 Deliverable Objectives</b> .....	<b>5</b>
1.2.1 Work package Objectives .....	5
1.2.2 Task Objectives .....	5
<b>2 Deliverables</b> .....	<b>6</b>
<b>3 Final iteration of the CoAP Testing Tool</b> .....	<b>7</b>
<b>3.1 The standard</b> .....	<b>7</b>
<b>3.2 The Test Description</b> .....	<b>7</b>
<b>3.3 Implementation</b> .....	<b>7</b>
<b>3.4 Integration</b> .....	<b>9</b>
<b>3.5 Last fine-tuning actions</b> .....	<b>9</b>
<b>4 Demo</b> .....	<b>10</b>
<b>4.1 Goal of the demo</b> .....	<b>10</b>
<b>4.2 Set-up of the demo</b> .....	<b>10</b>
<b>4.3 Outcomes of the demo</b> .....	<b>10</b>
<b>5 Annex</b> .....	<b>11</b>
<b>5.1 F-Interop session</b> .....	<b>11</b>
<b>5.2 An example of CoAP interoperability test description from ETSI plugtest CoAP#4, IETF89 (London)</b> .....	<b>12</b>
<b>5.3 An example of Test Extended Description (YAML file)</b> .....	<b>13</b>
<b>5.4 An example of CoAP Interoperability Test Script</b> .....	<b>14</b>

# List of Acronyms

---

6TiSCH	IPv6 over the TSCH mode of IEEE 802.15.4e
CoAP	Constrained Application Protocol
EC	European Commission
ETSI	European Telecommunications Standards Institute
EU	European Union
GPS	Global Positioning System
GUI	Graphical User Interphase
HTTPS	Hypertext Transfer Protocol Secure
ICT	Information and Communication Technologies
ID	Identifier
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IoT	Internet of Things
IP	Internet Protocol
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6
ISO	International Standards Organization
IT	Information Technology
MAC	Media Access Control
OS	Operating System
R&D	Research & Development
RR	Resources Repository
SME	Small Medium Enterprise
SO	Session Orchestrator
TAT	Test Analysis Tool
TED	Test Extended Description
TL	Task Leader
TT	Testing Tool
URL	Uniform Resource Locator
WP	Work Package
W3C	World Wide Web Consortium

# 1 Introduction

---

## 1.1 About F-Interop

---

F-Interop is a Horizon 2020 European Research project, which proposes to extend the European research infrastructure (FIRE+) with online and remote interoperability and performance test tools supporting emerging technologies from research to standardization and to market launch. The outcome will be a set of tools enabling:

- Standardization communities to save time and resources, to be more inclusive with partners who cannot afford travelling, and to accelerate standardization processes;
- SMEs and companies to develop standards-based interoperable products with a shorter time-to-market and significantly lowered engineering and financial overhead.

F-Interop intends to position FIRE+ as an accelerator for new standards and innovations.

## 1.2 Deliverable Objectives

---

### 1.2.1 Work package Objectives

- Research and develop the online remote interoperability test key enablers
- Develop the conformance test enablers
- Implement and fine tune the requested tools with a modular architecture for extensibility

### 1.2.2 Task Objectives

#### 1.2.2.1 T2.1: Online interop test core enablers M1-M33 (Task Leader: Inria)

**Work.** The main objective of this task is to define and implement the components of the F-Interop online remote interoperability-testing framework. This includes: the cloud-based interoperability test script repository (as well as its management), the test servers and test suites automation, as well as the libraries, adapters, API and hardware interfaces, and its reporting capability. This task includes developing new methods simplifying online remote interoperability testing. This task also considers associated security and authentication issues.

**Roles:** Inria will lead the task, and will integrate contributions from ETSI, UL and EANTC.

**Outcome:** All components and key enablers needed for online remote interoperability testing.

#### 1.2.2.2 T2.2: Complementary conformance test enablers M1-M33 (Task Leader: Inria)

**Work.** Conformance is a pre-requisite for interoperability. This task will provide the enablers for online remote conformance testing which complement the key enabled developed in task T2.1. We will develop new methods and/or adaptations of existing conformance testing tools to take into account the specific case of interacting online and remotely with the implementation under test (IUT). Examples include sniffers platforms, tools for measuring end-to-end latency (for example based on GPS synchronization), a protocol dissector.

**Roles:** Inria will lead the task, and will integrate contributions from ETSI, UL and EANTC.

**Outcome:** Additional components needed for online remote conformance testing. These components will complement the key enablers developed in task T2.1.

## 2 Deliverables

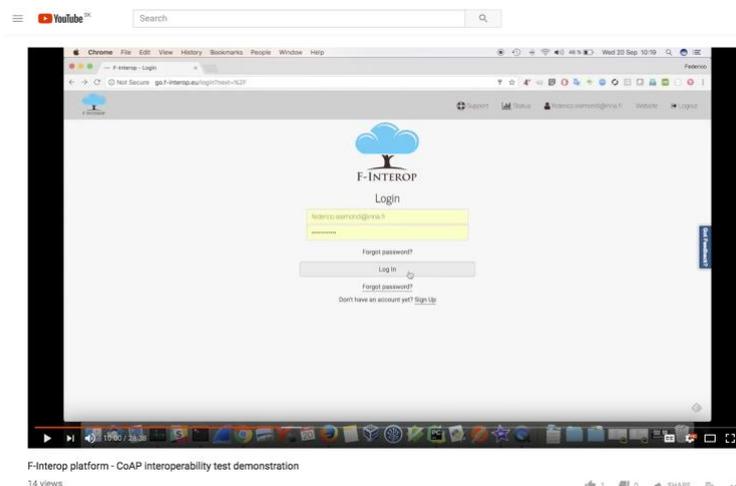
---

The deliverable corresponds to “Online interop test core enablers final iteration”. The deliverable itself is in the shape of source code and two videos demonstrating a CoAP interoperability test session in action using these enablers.

The videos present the setup and execution of a test which correspond to steps 0 to step 6 of the F-Interop session actions (see F-Interop session annex). The current document acts as complementary information of the deliverable D2.7, and focuses on stating what has been achieved and demonstrated until up to the current status.

The deliverable is composed by the following resources:

- (1) videos demonstrating a CoAP interoperability test session in action



the videos can be found at:

- single-user session (using reference implementation): <https://youtu.be/8rd4xCTc9zI>
- user-to-user session: <https://youtu.be/k0Q19zFaRp8>

- (2) the source code of the deliverable which has been tagged in F-Interop gitlab as 'D2.6':

- Testing Tool:  
source code: [https://gitlab.f-interop.eu/f-interop-contributors/coap\\_testing\\_tool/tags/D2.6](https://gitlab.f-interop.eu/f-interop-contributors/coap_testing_tool/tags/D2.6)
- Agent:  
source code : <https://gitlab.f-interop.eu/f-interop-contributors/agent/tags/D2.6>
- Session orchestrator:  
source code : [https://gitlab.f-interop.eu/f-interop/f-interop\\_ietf/tags/D2.6](https://gitlab.f-interop.eu/f-interop/f-interop_ietf/tags/D2.6)

## 3 Final iteration of the CoAP Testing Tool

---

This section presents the CoAP Testing tool and the current state of implementation of its features.

### 3.1 The standard

---

CoAP, the Constrained Application Protocol, standardized by the IETF CoRE working group, is the de-facto web transfer protocol for constrained devices. It is based on a request-response interaction model, and its basic architecture includes a CoAP client and a CoAP server.

CoAP runs on the connection-less unreliable transport protocol UDP. Therefore, it implements at application layer some lightweight reliable mechanisms for detecting duplication, re-ordering and acknowledging delivery. Orthogonal to these protocol features, CoAP defines also the GET, POST, PUT and DELETE methods in RFC7252. Then RFC7641 extends the feature set with the OBSERVE mechanism: CoAP clients can “observe” states of resources from CoAP servers (in other words it can retrieve the value of a resource, and get its updates over a given period of time).

### 3.2 The Test Description

---

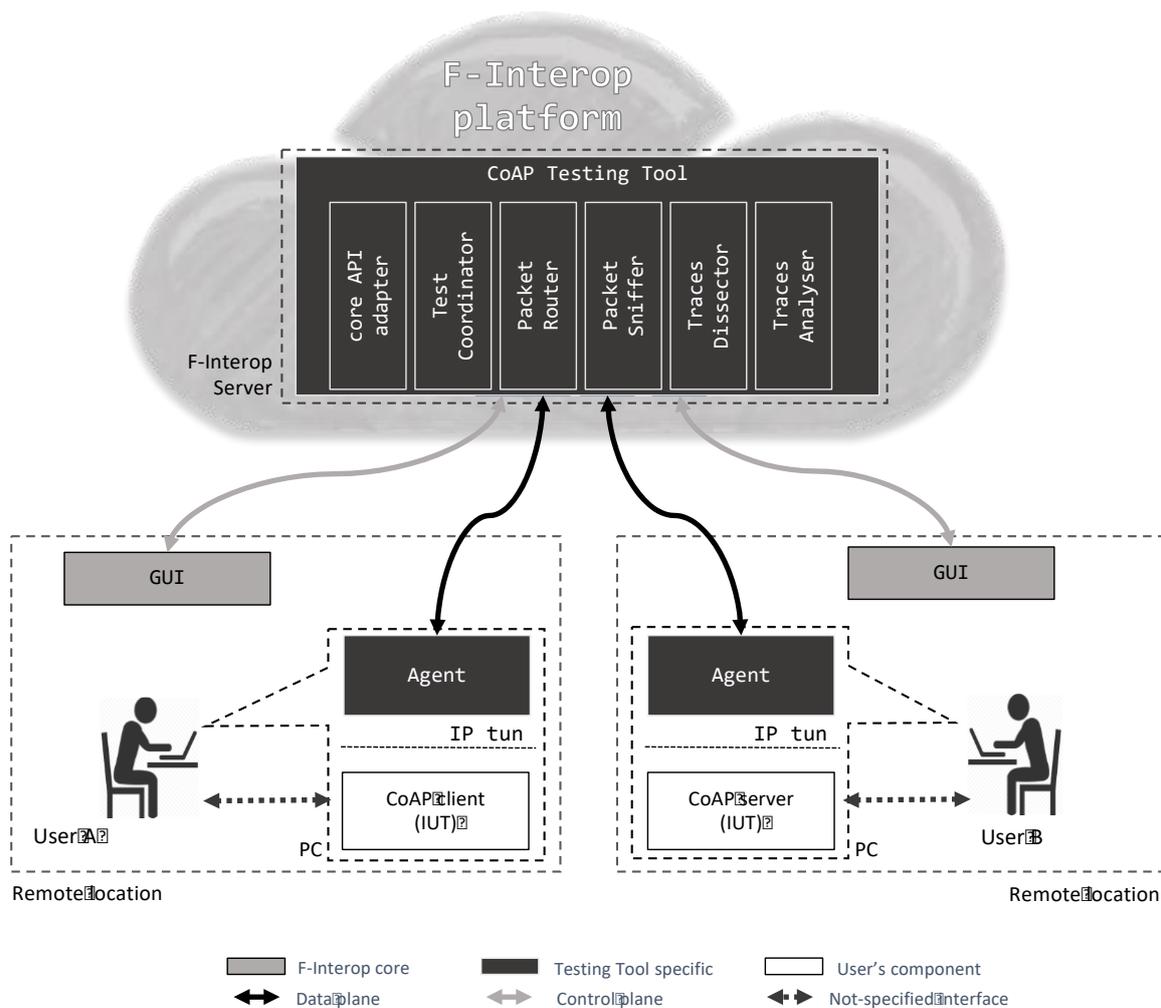
All the aforementioned protocol features and mechanisms can be tested with the F-Interop testing tool. The CoAP reference interoperability Test Description, TD has been integrated into F-Interop.

### 3.3 Implementation

---

The CoAP Testing Tool offers to end users a controlled and feature-rich environment that eases the execution of the online and remote standard-based interoperability test procedure.

The following figure shows the setup for running a CoAP remote interoperability test between two IUTs. Two protocol-specific components (the Agent and the Testing Tool) have been implemented for supporting CoAP tests.



The main components of the CoAP Testing Tool are:

- **Packet Router and Agents:** establishes a VPN-like setup between IUTs. The controlled environment helps users bypass UDP-blocking firewalls and other middle boxes installed in their facilities. The setup creates IPv6 network interfaces bound to the VPN, therefore users can test their implementations over IPv6 regardless the IP version supported by their internet provider. The Packet Router is the middle-box between IUT1's interface and IUT2's interface, it can act as a lossy gateway for test scenarios which require simulating a lossy context.
- **Test Coordinator:** coordinates the entire interoperability test. It iterates over the test steps described in the test description. It dispatches commands to users through the GUI, based on the TD (e.g. "user1: CoAP Client is requested to send a GET request"). The user is guided to perform the test remotely.
- **Packet Sniffer:** sniffs the traffic exchanged between IUTs and generates PCAP files records. The component enables the export feature for network traces so users can analyze the exchanged frames using tooling outside of F-Interop e.g. wireshark or some analysis scripts code.
- **Traces Dissector:** dissects the exchanged messages between IUTs and provides a human readable representation of the packets. It provides a wireshark-like view to helps users find problems encountered during the interoperability test execution.
- **Traces Analyzer:** analyzes the traffic exchanged between IUTs during a test case. The tool

automatically issues PASS, FAIL, INCONCLUSIVE verdicts after each test case. The analysis is based on the CHECK steps of the test cases description.

## 3.4 Integration

---

The core API adapter module handles the integration with F-Interop's core services such as GUI and Results Repository.

The integration enables two interoperability test use cases: (i) remote user-to-user interoperability session, as described in previous figure, and (ii) single-user sessions to test interoperability against reference implementations.

Both use cases are demonstrated on the demonstration videos referenced in previous section.

## 3.5 Last fine-tuning actions

---

In future developments of the CoAP testing tool needs to integrate the results repository, which is the last pending action regarding the integration with the F-Interop platform. Moreover, we will focus on fine-tuning the test cases implementation for the analysis for avoiding any false positive issued verdict.

## 4 Demo

---

### 4.1 Goal of the demo

---

The goal of the demonstration is to demonstrate how the interoperability core enablers have been used for implementing two key use cases for interoperability testing, those are:

- User-to-user session interoperability testing: each user driving his/her own implementation under test
- Single-user session interoperability testing: user plays a set of tests between his/her implementation against a reference implementation hosted in the platform.

### 4.2 Set-up of the demo

---

We have written an online document which describes what is needed to set up the testing environment here: <http://doc.f-interop.eu/interop/#guided-tour-coap-test-suite>

This is complementary information to the one provided by the GUI while executing the CoAP interop tests.

### 4.3 Outcomes of the demo

---

The videos demonstrate full remote & online interoperability test for two different use cases.

## 5 Annex

---

### 5.1 F-Interop session

---

The set of actions for executing any type of tests (interoperability, conformance, performance, etc.) in the F-Interop platform are defined as the “**F-Interop session**” and are summarized in the following table.

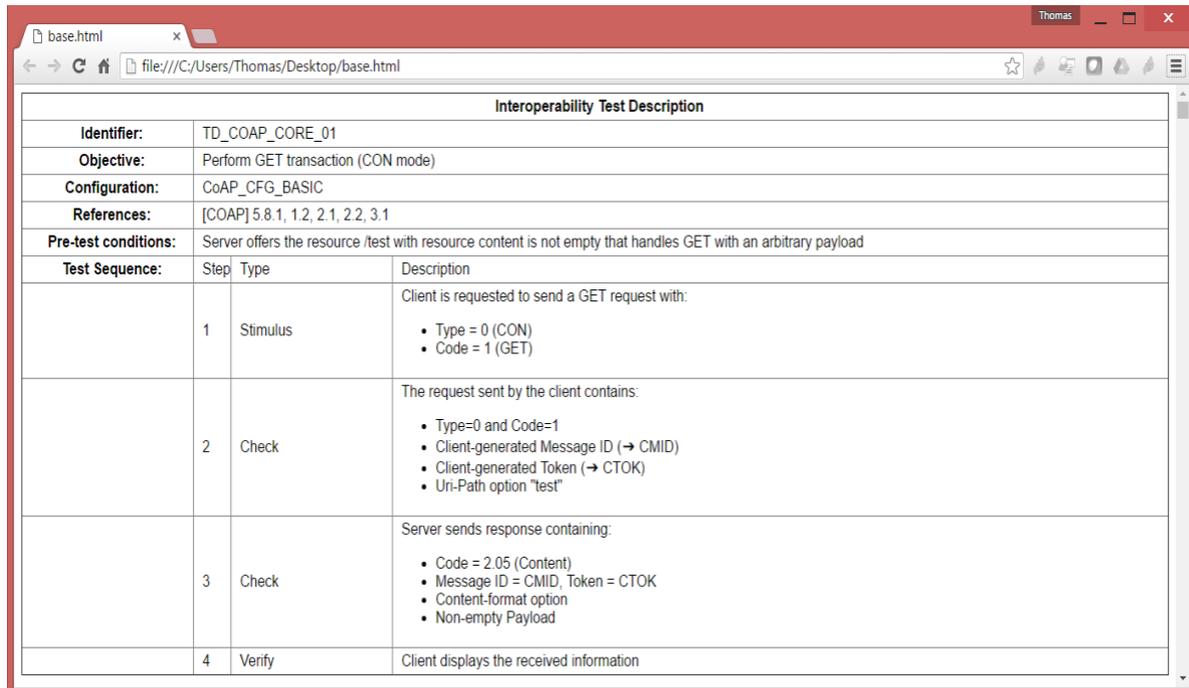
Step	Action	Description
0	FI-User authentication and authorization. IUT registration / identification	FI-User authenticates in a secure way (prior FI-User registration needed) in FI-Platform. FI-User needs to be authorized to use FI-Platform resources. FI-User identifies which IUT he/she will test (prior IUT registration needed).
1	Test suites discovery and selection	FI-User starts by discovering the available test suites and by selecting the one he/she wants to execute.
2	Resource description	FI-User specifies/selects resources in the F-Interop-Platform that are needed for his/her F-Interop session including the location models <sup>1</sup> , testing tools, libraries, etc. During this phase FI-Platform may request information from FI-User or provide information to FI-User for a coherent selection of the required resources.
3	Resource reservation	The resources selected in the previous step are actually reserved.
4	Resource provisioning, configuration and session setup	The instantiation of the F-Interop-Platform resources that fit best with the FI-User needs is done.
5	<b>Test execution</b>	The online F-Interop test campaign is launched and the selected (executable) test suites are executed against the IUTs.
6	<b>Results analysis and report</b>	Test execution information is analysed. The test results and verdicts are provided together with explanations in case of FAIL or INCONCLUSIVE verdicts or something wrong happened. A report can be provided under request in case for example the FI-User wants to apply for a certification/labelling program.
7	Session storage	Storage of the F-Interop session information (Session-id, User-id, FI-User's IUT-id, IUTs' version, test description, test version, testing tool, test log and results, etc.). This has to remain accessible beyond the F-Interop session for the involved parties.

**Table 1 - F-Interop Session**

---

<sup>1</sup> Location models are the different configurations for the location of components of the test. These will be defined in D1.3.1. [6]

## 5.2 An example of CoAP interoperability test description from ETSI plugtest CoAP#4, IETF89 (London)



The screenshot shows a web browser window with the address bar displaying 'file:///C:/Users/Thomas/Desktop/base.html'. The main content area contains a table with the following structure:

Interoperability Test Description		
<b>Identifier:</b>	TD_COAP_CORE_01	
<b>Objective:</b>	Perform GET transaction (CON mode)	
<b>Configuration:</b>	CoAP_CFG_BASIC	
<b>References:</b>	[COAP] 5.8.1, 1.2, 2.1, 2.2, 3.1	
<b>Pre-test conditions:</b>	Server offers the resource /test with resource content is not empty that handles GET with an arbitrary payload	
<b>Test Sequence:</b>	Step	Type
	1	Stimulus
	2	Check
	3	Check
	4	Verify

The 'Description' column for each step contains the following details:

- Step 1 (Stimulus):** Client is requested to send a GET request with:
  - Type = 0 (CON)
  - Code = 1 (GET)
- Step 2 (Check):** The request sent by the client contains:
  - Type=0 and Code=1
  - Client-generated Message ID (→ CMID)
  - Client-generated Token (→ CTOK)
  - Uri-Path option "test"
- Step 3 (Check):** Server sends response containing:
  - Code = 2.05 (Content)
  - Message ID = CMID, Token = CTOK
  - Content-format option
  - Non-empty Payload
- Step 4 (Verify):** Client displays the received information

## 5.3 An example of Test Extended Description (YAML file)

---

```
testcase_id: TD_COAP_CORE_01
uri : http://f-interop.paris.inria.fr/tests/TD_COAP_CORE_01
configuration: CoAP_configuration_BASIC
objective: Perform GET transaction(CON mode)
pre_conditions: Server offers the resource /test with resource content is not empty that handles GET with an
arbitrary payload
references: '[COAP] 5.8.1, 1.2, 2.1, 2.2, 3.1'
sequence:
  - step_id: 'TD_COAP_CORE_01_v01_step_01'
    type: stimuli
    iut : coap_client
    description:
      - Client is requested to send a GET request with
      - Type = 0(CON)
      - Code = 1(GET)

  - step_id: TD_COAP_CORE_01_v01_step_02
    type: check
    description:
      - The request sent by the client contains
      - Type=0 and Code=1
      - Client-generated Message ID(\u2794 CMID)
      - Client-generated Token(\u2794 CTOK)
      - Uri-Path option "test"

  - step_id: TD_COAP_CORE_01_v01_step_03
    type: check
    description:
      - Server sends response containing
      - Code = 2.05(Content)
      - Message ID = CMID, Token = CTOK
      - Content-format option
      - Non-empty Payload

  - step_id: TD_COAP_CORE_01_v01_step_04
    type: verify
    iut: coap_client
```

description:

- Client displays the received information

## 5.4 An example of CoAP Interoperability Test Script

---

### Under the Hood: What's a test?



```
#!/usr/bin/env python3
from ttproto.ts_coap.common import CoAPTestcase
from ttproto.ts_coap.templates import *

class TD_COAP_CORE_01 (CoAPTestcase):

    def run (self):

        # match stimuli
        self.match_coap ("client", CoAP (type="con", code="get",
                                         opt = self.uri ("/test")))
        CMID = self.frame.coap["mid"]
        CTOK = self.frame.coap["tok"]

        # match step 2
        self.next()
        if self.match_coap ("server", CoAP (
            code = 2.05,
            mid = CMID,
            tok = CTOK,
            pl = Not(b""),
        )):

            # match step 3
            self.match_coap ("server", CoAP (
                opt = Opt (CoAPOptionContentFormat()),
            ), "fail")
```

