



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

Deliverable D2.5
Complementary conformance test enablers
2nd 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/

Due date of deliverable: M23

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.



Co-funded by the
European Union



Co-funded by the
Swiss Confederation

Document properties

Responsible partner	INRIA
Author(s)/editor(s)	Rémy Léone, Thomas Watteyne
Version	1.0
Keywords	Interoperability Testing, Remote Testing, Online, Platform, Testing components, Test enablers

Abstract

This short report corresponds to the deliverable **D2.5 – Complementary conformance test enablers 2nd iteration**. It provides complementary information to the deliverable D.2.5 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 between the 1st iteration and the current status of the F-Interop online remote interoperability framework and the key enablers needed.

Table of Contents

Table of Contents	3
List of Acronyms	4
1 Introduction	5
1.1 About F-Interop	5
1.2 Deliverable Objectives	5
1.2.1 Work package Objectives	5
1.2.2 Task Objectives	5
2 Deliverables	6
3 6TiSCH Testing Tool	7
3.1 Testing tool features	7
3.2 Interactions between components	8
3.2.1 GUI & TT integration:.....	8
3.2.2 CLI & TT integration.....	8
3.2.3 Agent & TT integration.....	8
4 Demo	9
4.1 Goal of the demo	9
4.2 Set-up of the demo	9
4.3 Outcomes of the demo	9
5 Conclusion and future work	10
6 Annex	11
6.1 An example of 6TiSCH Test Description (YAML file)	11

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.

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 “**Complementary conformance test enablers 2nd iteration**”. The deliverable itself is in the shape of source code and a video demonstrating a 6TiSCH conformance test session in action using these enablers. The video presents the execution of a 6TiSCH test.

The following resources compose the deliverable:

1. Video introducing the setup of the 6TiSCH demo: <https://youtu.be/ArFmPyQngSk>
2. Video showing up the demo: <https://youtu.be/9qO8YI9Mji0>
3. Slides used during the video (attached)
4. The source code of the deliverable which has been tagged in F-Interop gitlab as 'video_demo': https://gitlab.distantaccess.com/f-interop/6tisch_testing_tool/tree/video_demo

3 6TiSCH Testing Tool

This section presents the 6TiSCH Testing tool and the current state of implementation of its features.

3.1 Testing tool features

The achievements regarding the 6TiSCH testing tool were:

- All core dissection requirements have been submitted and accepted into the Wireshark official repository:
 - o <https://code.wireshark.org/review/#/c/16671/>
 - o <https://code.wireshark.org/review/#/c/20547/>
 - o <https://code.wireshark.org/review/#/c/20878/>
 - o <https://code.wireshark.org/review/#/c/20928/>
 - o <https://code.wireshark.org/review/#/c/21187/>
 - o <https://code.wireshark.org/review/#/c/21336/>
 - o <https://code.wireshark.org/review/#/c/22484/>
 - o <https://code.wireshark.org/review/#/c/22500/>

This ensures a wide spreading of the core features of the 6TiSCH testing tools towards a large community of users.
- Complete AMQP integration between the 6TiSCH agent and the testing tool. This ensure that other components can see the messages exchanged between the agent and the testing tool and act upon them.
- Packaging of the testing tool in a Docker image format (which the orchestrator launch when the session starts). This ensures that we have a reliable way to build and deploy the testing tool on many servers.
- The agent of the 6TiSCH testing tool support flashing on TelosB and OpenMote board, which are the most widespread hardware boards, used to develop the 6TiSCH protocol.
- Following tests implemented (Test Reference: <https://bitbucket.org/6tisch/td-6tisch/src/>):
 - o TD_6TiSCH_6P_01
 - o TD_6TiSCH_6P_02
 - o TD_6TiSCH_6P_03
 - o TD_6TiSCH_SECJOIN_05
 - o TD_6TiSCH_SECJOIN_06
 - o TD_6TiSCH_MINIMAL_01
 - o TD_6TiSCH_MINIMAL_04
- The 6TiSCH Testing Tool already received contributions to support the SECJOIN test cases. The ARMOUR project sponsors those contributions (main developer [Mališa Vučinić](#)). Those tests passed on real hardware (OpenMote with OpenWSN). The testing tool now fully supports them. This demonstrates that new contributors can contribute to the 6TiSCH Testing Tool and have their contributions merged quickly (<1 month of development for the discovery of the 6TiSCH testing tool, test implementation, integration and deployment).

3.2 Interactions between components

During this first phase of the **Error! Reference source not found.**, we have achieved integration between several components of the F-Interop architecture.

3.2.1 GUI & TT integration:

The GUI is under refactoring effort to integrate diverse testing tools with a generic API. The 6TiSCH testing tool aims to support this upcoming version of the GUI API rather than the previous one, which have CoAP testing tool specifics abstractions.

3.2.2 CLI & TT integration

As an optional way of running the execution phase, the User can opt to use the CLI, which integrates all functionalities provided by the TT.

CLI handles features like chat between a user participating in an interoperability session for enhancing the coordination of the interoperability tests, commands for verifying the correct deployment and behaviour of the TT, and more.

3.2.3 Agent & TT integration

Testing tool drives the agent from beginning to end of the session. This includes:

- Flashing of a hardware board
- Sniffing of the emitted frames
- Aggregation of the collected packets into a PCAP-ng capture file.

4 Demo

4.1 Goal of the demo

The goal of the demo is to show that the 6TiSCH Testing Tool can run a conformity test. For this demo, we will target the TD_6TiSCH_6P_01 test. We want to demonstrate in this demo that the core features of the testing tool are all working reliably. Those features are:

- Packet sniffing
- Dissection of the captured frame
- Assertion upon those captured information
- Verdict emission

Now, the 6TiSCH testing tool and the GUI are not integrated. The GUI's API supports the requirements of the CoAP testing tool but is not suited for generic testing™. An ongoing effort aims to fix this. In the meantime, the 6TiSCH testing tool provides basic CLI tools that mock what the GUI would do in a regular setup.

During the demo, we use this CLI and show the main interactions that the testing tool got with the user:

- Asking questions (Completion of a stimuli, ...)
- Display of a verdict

4.2 Set-up of the demo

The IUT used in this demo are OpenMote boards with the latest version of OpenWSN. OpenWSN is the reference implementation of 6TiSCH and it fully supports the 6P protocol that we intend to test.

We use a CLI interface to ask the user question. The API used for this feature is the generic API that the GUI will implement in its upcoming version.

Once the test is finished, the testing tool emits a verdict message. The CLI captures this verdict message and print it to the user.

4.3 Outcomes of the demo

When the demo is over, the user has witness a verdict (PASS/FAIL). The logs and traces (PCAP files) of its test session are also available for the user.

5 Conclusion and future work

This document presented the first achievements and demo of the 6TiSCH testing tool.

The future work for the 6TiSCH Testing Tool should include:

- Integration with the generic API of the GUI
- Improve the reliability of hardware support (detection, cross-platform support)
- Support more 6TiSCH tests

6 Annex

6.1 An example of 6TiSCH Test Description (YAML file)

```
TD_6TiSCH_6P_01:
  obj: check a 6P COUNT command is successful
  cfg:
    - topology => a DAGroot and a 6N within range (single hop)
    - all devices configured to communicate on a single frequency (frequency to be
allocated during the event)
    - link-layer security disabled
  ref:
    - IEEE802.15.4-2015
    - RFC8180
    - draft-ietf-6tisch-6top-protocol-05
  pre:
    - DAGroot and 6N are turned off
  seq:
    - s:
      - turn on the DAGroot
      - turn on the 6N
    - c:
      - the 6N synchronizes to the DAGroot

    - s:
      - have the 6N send a 6P ADD request to the DAGroot
        - asking for a single TX cells from 6N to DAGroot
        - with the candidate celllist set to [(4,5)]
      - check that the format of the request comply with draft-ietf-6tisch-6top-protocol-05
      - how this is done is implementation-specific
    - c:
      - the 6P transaction is succesful and both the DAGroot and 6N have added a cell from
6N to DAGroot at slotoffset=4 and channeloffset=5
      - check that the format of the reply comply with draft-ietf-6tisch-6top-protocol-05

    - s:
      - have the 6N send a 6P COUNT request to the DAGroot
      - check that the format of the request comply with draft-ietf-6tisch-6top-protocol-05
      - how this is done is implementation-specific
    - c:
      - check that, in the 6P response, the returned code SUCCESS
      - check that, in the 6P response, the count value is 1
      - check that the format of the reply comply with draft-ietf-6tisch-6top-protocol-05
```