SIXTH FRAMEWORK PROGRAMME
PRIORITY 2
Information Society Technologies

SPECIFIC TARGETED RESEARCH OR INNOVATION PROJECT

Deliverable D2.6 - "Validation Plan"

Project acronym: **OneLab**
Project full title: **An open networking laboratory supporting communication network research across heterogeneous environments**
Proposal/Contract no.: **034819**

Date of preparation of Deliverable: **Sep, 27th, 2007**

Document properties

Document Number:	IST–4-034819-OneLab-D2.6
Document Title:	Validation Plan
Document responsible:	T. BOURGEAU (UPMC)
Author(s)/editor(s):	T. BOURGEAU (UPMC) B. DONNET (UCL) M. A. CHAOUI (INRIA) L. RIZZO (UNIPI) B. MATHIEU (FT) M. PILARSKI (TP) I. SOTO (UC3M) R. CANONICO (UNINA) G. IANNACCONE (Quantavis)
Dissemination level[1]:	PU
Security Type:[2]	Public
Status of the Document:[3]:	Released
Version & Revision History	2.1 Final Version

[1] Dissemination level as defined in the EU Contract:
PU = Public
PP = Distribution limited to 6th FP participants
RE = Distribution to a group specified by the consortium
CO = Confidential, only allowed for members of the consortium

[2] Security Type:
Confidential Internal circulation within project (and Commission project Officer if requested)
Restricted Restricted circulation defined in the document or dissemination level PP
Internal With no confidential content but intended for internal use
Public Public document

[3] Draft <#>, Draft <#> Reviewed, Final Draft, Released, Revised Draft <#>, etc

Table of contents

1. DOCUMENT DESCRIPTION **5**
1.1. ACRONYMS 6
2. INTRODUCTION **7**
3. PLANETLAB EUROPE PLATFORM VALIDATION OVERVIEW **8**
3.1. OVERVIEW OF THE PLANETLAB ARCHITECTURE 8
3.2. NAMING CONVENTIONS 9
3.3. OVERVIEW OF THE VALIDATION PROCESS 10
4. UNITARY ONELAB VALIDATION **12**
4.1. REQUIREMENTS 12
4.1.1. WiMAX *12*
4.1.2. UMTS *12*
4.1.3. Multihomed *13*
4.1.4. Wireless ad hoc *13*
4.1.5. Emulation *13*
4.2. EXPERIMENTS AND TESTS 14
4.2.1. WiMAX *14*
4.2.2. UMTS *14*
4.2.3. Multihomed *14*
4.2.4. Wireless ad hoc *15*
4.2.5. Emulation *17*
4.3. VALIDATION TOOLS 18
5. MONITORING VALIDATION **19**
5.1. PASSIVE MONITORING 19
5.1.1. Requirements *19*
5.1.2. Experiments and tests *19*
5.1.3. Validation tools *19*
5.2. TOPOLOGY MONITORING 19
5.2.1. Requirements *20*
5.2.2. Experiments and tests *20*
5.3.1. Validation tools *21*
6. ITERATIVE INTEGRATED ONELAB VALIDATION **21**
6.1. MAIN GOALS AND FEATURES 21
6.2. REQUIREMENTS 22
6.3. EXPERIMENTS AND TESTS 22
6.3.1. MyPLC installation and configuration *22*
6.3.2. MyPLC population *23*
6.3.3. Node Installation and simulation: *24*
6.3.4. Test the testbed access: *24*
7. GENERALIZED PLANETLAB EUROPE VALIDATION **25**
7.1. GLOBAL PROCESS 25
7.2. REQUIREMENTS 25
7.2.1. MyPLC installation and configuration for PlanetLab Europe *25*
7.2.2. PlanetLab Europe population *26*
7.2.3. Federation: *26*
7.2.4. PlanetLab Europe Access: *27*
7.2.5. Monitoring services *27*
7.3. EXPERIMENTS AND TESTS 27
7.3.1. MyPLC installation and configuration for PlanetLab Europe *27*
7.3.2. PlanetLab Europe population *29*
7.3.3. Federation: *30*
7.3.4. PlanetLab Europe access: *30*
7.3.5. Monitoring services *31*
7.4. VALIDATION TOOLS 31
8. CONCLUSION **32**

9. RELATION TO OTHER DELIVERABLES 32

10. REFERENCES 33

1. Document description

This deliverable is part of OneLab's Integration work package (WP2) and it defines the validation process that will begin in Month 18.

The European OneLab project will produce core code to run a testbed called PlanetLab Europe and will introduce several new components, to be integrated into PlanetLab Europe. These new components are:

- Passive monitoring component.
- Topology monitoring component.
- WiMAX component.
- UMTS component.
- Multihomed component.
- Wireless ad hoc component.
- Emulation component.

The aim of the validation process is to provide a robust and stable publicly operational PlanetLab Europe testbed at the end of the OneLab project. The plan draws on the participation of each of the project partners that is producing components for this platform, and includes the following features:

- **Unitary validation**: Specification of requirements and experiments to test each new component in a standalone manner.
- **Monitoring validation**: Definition of requirements and experiments for the passive monitoring and the topology monitoring components in a realistic environment.
- **Iterative integrated validation:** Several of the components created in OneLab add functionality and drivers to the kernel that drives the testbed. All new contributions provided during the OneLab project will be integrated into this OneLab software. We propose an iterative validation process to continuously maintain the stability of the testbed during the integration of these modifications.
- **Generalized validation:** The goal of the validation process is to provide a stable and robust platform at the end of the project. Generalized validation will be the last step, in which the entire platform will be tested in a realistic environment.

The remainder of this deliverable is organized as follows: Sec. 2 introduces the objectives for OneLab's validation plan; Sec. 3 gives an overview of the validation process; Sec. 4 defines the unitary validation; Sec. 5 describes the monitoring validation; Sec. 6 presents the iterative integrated validation; Sec. 7 describes the generalized validation; Sec. 8 summarizes this deliverable and, finally, Sec. 10 explains how this document is related to others validation deliverables.

1.1. Acronyms

UMTS	Universal Mobile Telecommunications System
WiMAX	Worldwide Interoperability for Microwave Access (IEEE 802.16 wireless networking)
Wi-Fi	Wireless Fidelity (IEEE 802.11 wireless networking)
PLC	PlanetLab Central
PLE	PlanetLab Europe
CMMI	Capability Maturity Model Integration
ISO	International Standards Organization
BGP	Border Gateway Protocol
AS	Autonomous System
QoS	Quality of Service
QoE	Quality of Experience
RSSI	Received Signal Strength Indication
TP	Telekomunikacja Polska
FT	France Telecom

2. Introduction

The European OneLab project builds on the PlanetLab distributed networking testbed. One of OneLab's principal goals is to put in place a robust and reliable European centre for PlanetLab, that can be used by academia and industry in Europe for testing new networking applications and concepts.

Three OneLab work packages make technical contributions. These are: WP2 Integration, WP3 Monitoring, and WP4 New Environments. It is these contributions that require validation, to ensure that OneLab delivers a working platform, with components that function as described in the project's Description of Work.

The Integration work package will build a centralized software development infrastructure. All OneLab partners will use it in order to integrate the different OneLab components as they are delivered throughout the course of the project. To ensure the consistency of the OneLab software, the integration work package will provide a private test platform to validate all the new contributions from OneLab in a realistic environment.

The Monitoring work package aims to deepen PlanetLab's monitoring capabilities in order to provide a better view of the underlying network. This work package adds two components to PlanetLab. One provides an API that allows applications to interrogate a passive monitoring system called CoMo [14], to obtain information about packets as they pass between PlanetLab hosts. Another provides a system that allows applications to obtain topology information about the underlying network by interrogating a topology monitoring component that gathered active measurements data such as traceroutes.

The New Environments work package adds five testbed components: WiMAX links, UMTS connections, multihomed environments, a wireless ad hoc network, and an emulation component. Each of these components is independent from the others.

With the combination of these work packages, OneLab aims to:

- Build a sophisticated monitoring system.
- Extend PlanetLab into new environments.
- Federate PlanetLab.

The validation process proposed for OneLab will provide a system-wide validation in real conditions.

First, the Monitoring, and New Environments work packages will benefit from either standalone tests or integrated tests (involving many components at once). The validation tests can also be iterative as each component is independent from the others. Second, the Integration work package will provide a useful loop back to make the tests of New Environments and Monitoring as realistic as possible thanks to the private test platform. Finally the migration of the OneLab components from the private test platform to a publicly available PlanetLab Europe platform will have to be tested by a general validation process in order to ensure the robustness and stability of the combined OneLab contributions.

This deliverable D2.6 provides a step-by-step guide through a validation process intended to assure the reliability and the robustness of the integrated components, without being overly burdensome. The validation plan specifies a set of tests that will ensure the reliability and robustness of the platform. Validation will proceed by comparing the measurements gathered in a series of real experiments against validation criteria specified in this deliverable.

The validation plan is divided into four phases:

Phase 1: Unitary validation (new environments)

Phase 2: Monitoring validation

Phase 3: Iterative integrated validation

Phase 4: Generalized PlanetLab Europe validation

For each phase, we will:

- Specify the validation requirements.
- Describe experiments and tests that are based upon the specified validation requirements.
- Define validation tools to perform tests and experiments.

3. PlanetLab Europe platform validation overview

PlanetLab is a geographically distributed platform designed to support the deployment and evaluation of planetary-scale network services. PlanetLab is extended in OneLab by introducing new components. OneLab's goals are to provide a PlanetLab Europe platform that will extend PlanetLab into new environments beyond the traditional wired Internet, deepen PlanetLab's monitoring capabilities, and federate in order to provide a European administration for PlanetLab nodes in Europe. These enhancements will change the actual PlanetLab platform in important ways. It is therefore important to validate the new components in order to ensure that a robust platform is delivered at the end of the project.

3.1. Overview of the PlanetLab architecture

PlanetLab provides *distributed virtualization* and services that run in *slices* on the platform. A slice corresponds to a distributed set of virtual machines (VMs) that allocate resources distributed across a set of PlanetLab nodes. A PlanetLab node is a dedicated server, hosted by an organization, that runs PlanetLab software.

A VM is implemented by a *vserver*[4]. Vservers are the principal mechanism in PlanetLab for providing virtualization on a single node. It provides virtualization at the system call level by extending the non-reversible isolation for filesystems to other operating system resources.

Each PlanetLab node runs a *Virtual Machine Monitor* (VMM) that implements and isolates VMs. The *Node Manager* is a privileged "root" VM running on top of the VMM. It monitors and manages all the VMs on the node and enforces policies on creating VMs and allocating resources to them.

Fig. 1 shows the architecture of a node and the relationships that exist between VM, VMM, and the Node Manager.

The PlanetLab platform has to maintain a database of nodes, slices, resource allocations, and policies on a central server. For this, PlanetLab uses a centrally controlled brokerage service, developed by Princeton, called PlanetLab Central (PLC). It helps users and site administrators to create and delete slices and nodes, specify a boot script, and set user keys.

In adding new components, OneLab will modify the kernel and the behaviour of the VMM and VMs could change. The role of the validation Work Package is to avoid any anomalies and dysfunction between the existing and new components.

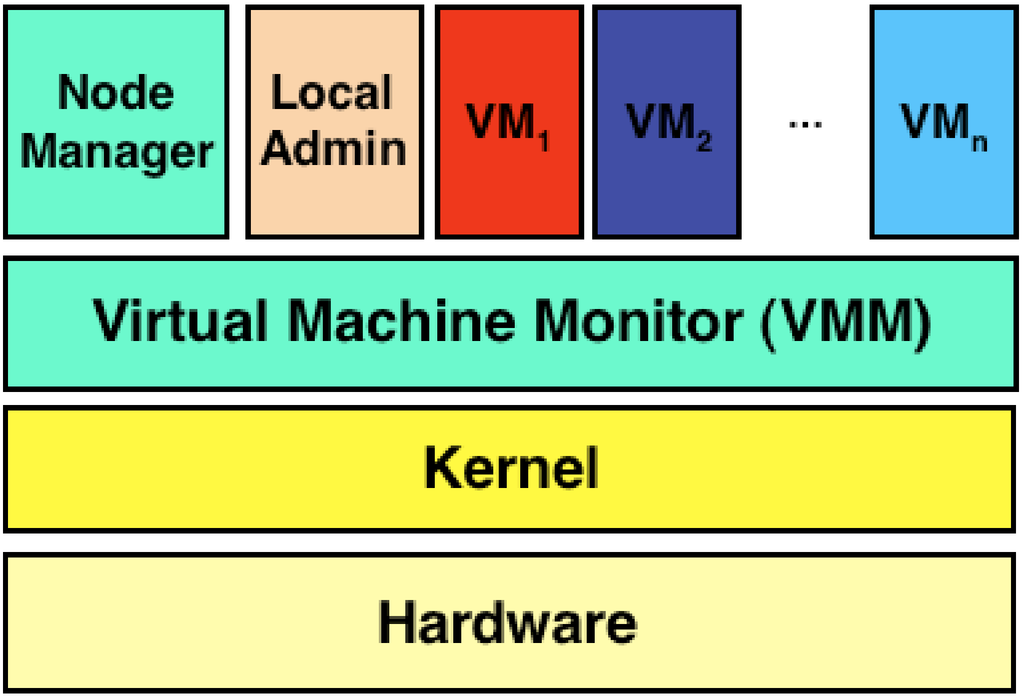


Figure 1: Overview of PlanetLab node architecture
(graphic courtesy of Larry Peterson, Princeton University)

3.2. Naming conventions

In this section we introduce our formal naming conventions approach. Well-known industrial validation models, such as CMMI and ISO 15504, inspire our approach. In particular, we draw upon the CMMI model's format for naming requirements, experiments and validation tools in order to have a better visibility on the validation results.

Requirements will be named as follows:

REQ_[The first four letters of the component]_[the number of the requirement]

For example, the first requirement for the UMTS subcomponent will be: **REQ_UMTS_01**

Experiments will be named as follow:

EXP_[The first four letters of the component]_[the number of the experiment]

For example, the first experiment for the WiMAX subcomponent will be: **EXP_WIMA_01**

The validation methods not only define the technical approach to perform validation, but also specify the needs for the facilities, equipment, and environments.

As introduced for the requirements and the experiments, we will use a formal denomination to associate each validation tool with a specific name.

Validation tools will be named as follows:

VAL_[The first four letters of the component]_[the number of the validation tool]

For example, the first validation tool associated with WiMAX validation will be: **VAL_WIMA_01**

3.3. Overview of the validation process

The quality of a system is highly influenced by the quality of the process used to acquire, develop, and maintain it. The purpose of our validation process is to demonstrate that PlanetLab Europe components fulfil their intended use and requirements when placed in their intended environment. To be acceptable to users, PlanetLab Europe components must perform as expected in their intended operational environment.

The new components of OneLab will be built on the current structure of PlanetLab. Therefore, it is important to have a standalone approach in the first validation phase to test the new contributions of OneLab separately from the current PlanetLab platform. This step is called the unitary validation.

As the integration of new technological components will change the PlanetLab kernel, we have drawn up a system wide validation that will progressively test each new component integrated into the platform in a realistic environment. Each new environment and monitoring component need feedback using a private test platform. The private test platform is a dedicated testing platform that allows each partner to experiment with their components and applications in a realistic integrated environment. This private test platform will integer specific drivers and functionalities that new environments component needs. This step is call the iterative integration validation phase. It will begin in Month 18 and will end in Month 24.

Furthermore, we will perform a homogeneous validation process and will test the entire platform API and services before the migration of the integrated OneLab components to the publicly available platform as operated by WP1; this last validation step is called generalized validation. The validation process is summarised in Figure 2.

In each of the previous validation steps, each partner involved in a specific component or subcomponent has to define:

- Requirements and measurements for each component.
- A set of experiments to be performed to validate all the components
- Tools and middleware to perform validation tests.

The results of the validation process will be gathered into a single document called the OneLab validation report, which is D5.2. All the partners involved in the validation process will have to send their results to UPMC, which is in charged of collecting all the validation data for this work package. Finally, as OneLab validation requires the collective effort of all partners, the responsibilities of each partner during the validation process are set down in this document, as shown in Table 1.

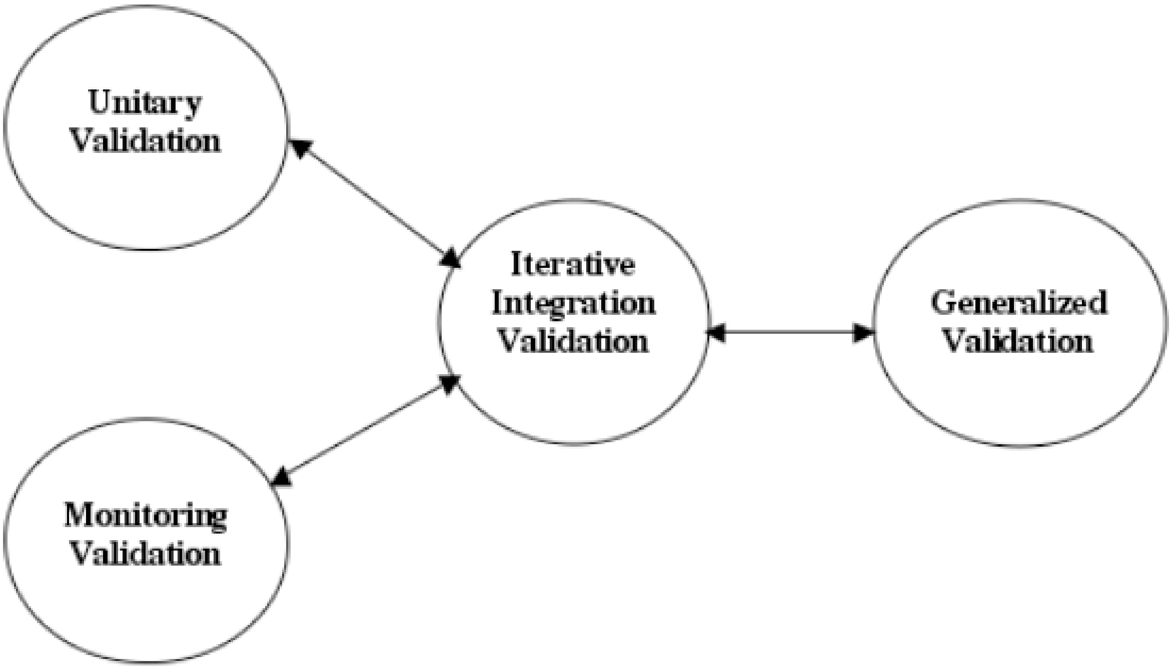


Figure 2: Overview of the validation process

Partners	Action expected during the validation process
UPMC	• Ensure consistency during the validation process. • Validation of the wired Topology Monitoring Component (Sec. 5) • Generalized PlanetLab Europe validation (Sec. 7)
INRIA	• Iterative integrated OneLab validation (Sec. 6)
FT	• Validation of the Wireless ad hoc component (Sec. 4) • Validation of the ad hoc topology monitoring (Sec. 5)
TP	• Validation of the Wireless ad hoc component (Sec. 4) • Validation of the ad hoc topology monitoring (Sec. 5)
UCL	• Validation of the WiMAX component (Sec. 4)

	• Validation of the wired topology monitoring (Sec. 5)
UniPi	• Validation of the Emulation component (Sec. 4)
CINI	• Validation of the UMTS component (Sec. 4)
UC3M	• Validation of the Multihomed component (Sec. 4)
Quantavis	• Validation of the passive monitoring component (Sec. 4)

Table 1. Role of each partner involved in the validation process

4. Unitary OneLab validation

The Unitary OneLab validation will include each new environments component to be tested. This validation will occur on the private PlanetLab test platform run by INRIA.

The New Environments work package provides a technological extension to PlanetLab, which adds a total of five testbed components. Each of these new components are described in several deliverables as follow:

- WiMAX: D4A.1.
- UMTS: D4B.1, D4B.2.
- Multihomed: D4C.1, D4C.2, D4C.3.
- Wireless ad hoc: D4D.1, D4D.2.
- Emulation: D4E.1, D4E.2, D4E.3.

In this section, we will define all the requirements and experiments that are required for the New Environments work package.

4.1. Requirements

4.1.1. WiMAX

REQ_WIMA_01	Evaluate the transmission delay, packet loss ratio, and delay jitter for the fixed node.
REQ_WIMA_02	Verification of the model and performance of the WiMAX network.

4.1.2. UMTS

REQ_UMTS_01	Characterisation of the impact of the radio link on the end-to-end QoS.
REQ_UMTS_02	Connect UMTS nodes to a different technology node as a local Wi-Fi hotspot.

4.1.3. Multihomed

REQ_MULT_01	Evaluate the ability of creating simulated link failures that affect traffic flows as perceived by applications inside a sliver.
REQ_MULT_02	Test automatic detection of path failures by REAP.
REQ_MULT_03	Test exploration of alternative paths, and selection of a working path.
REQ_MULT_04	Test re-configuration to use a new path (change of locators) for ping traffic.
REQ_MULT_05	Test re-configuration to use a new path (change of locators) for UDP and TCP traffic.

4.1.4. Wireless ad hoc

REQ_ADHO_01	Test a video streaming application on an ad hoc network with the source and the destination in 2 different ad hoc networks connected via a wired network
REQ_ADHO_02	QoS measurement to validate the video streaming application on the ad hoc network
REQ_ADHO_03	QoE measurement to validate a video streaming application on ad hoc network

4.1.5. Emulation

Emulation extensions for OneLab nodes are provided by an external box (which can be real or simulated) that implements the desired bandwidth limitations and packet delays. We call this box a DBox.

The DBox sits on the link between the OneLab node and the rest of the network, and its presence is recorded in the Node Networks table of the database hosted on the central site of the platform.

REQ_EMUL_01	Availability of a DBox on the link between a node and the rest of the laboratory network.
REQ_EMUL_02	Availability of PLC software modified with the changes to support the DBox

4.2. Experiments and tests

4.2.1. WiMAX

EXP_WIMA_01	Perform active measurements, with the fixed WiMAX-connected node to better characterise the behaviour of the WiMAX interface.
EXP_WIMA_02	Use measurements to tune the WiMAX parameters used in the link emulator
EXP_WIMA_03	Use measurements via traceroute and delays to show the differences when a connection passes through a wired connexion to a WiMAX connexion
EXP_WIMA_04	Compare the TCP WiMAX model results to the real deployed WiMAX results.

4.2.2. UMTS

EXP_UMTS_01	Measure the characteristics of the end-to-end path between a mobile node with UMTS connectivity and fixed nodes located remotely at another site of OneLab
EXP_UMTS_02	Measurement of the characteristics of the alternate path.
EXP_UMTS_03	Test the interoperability at the service level provided by the UMTS video gateway.

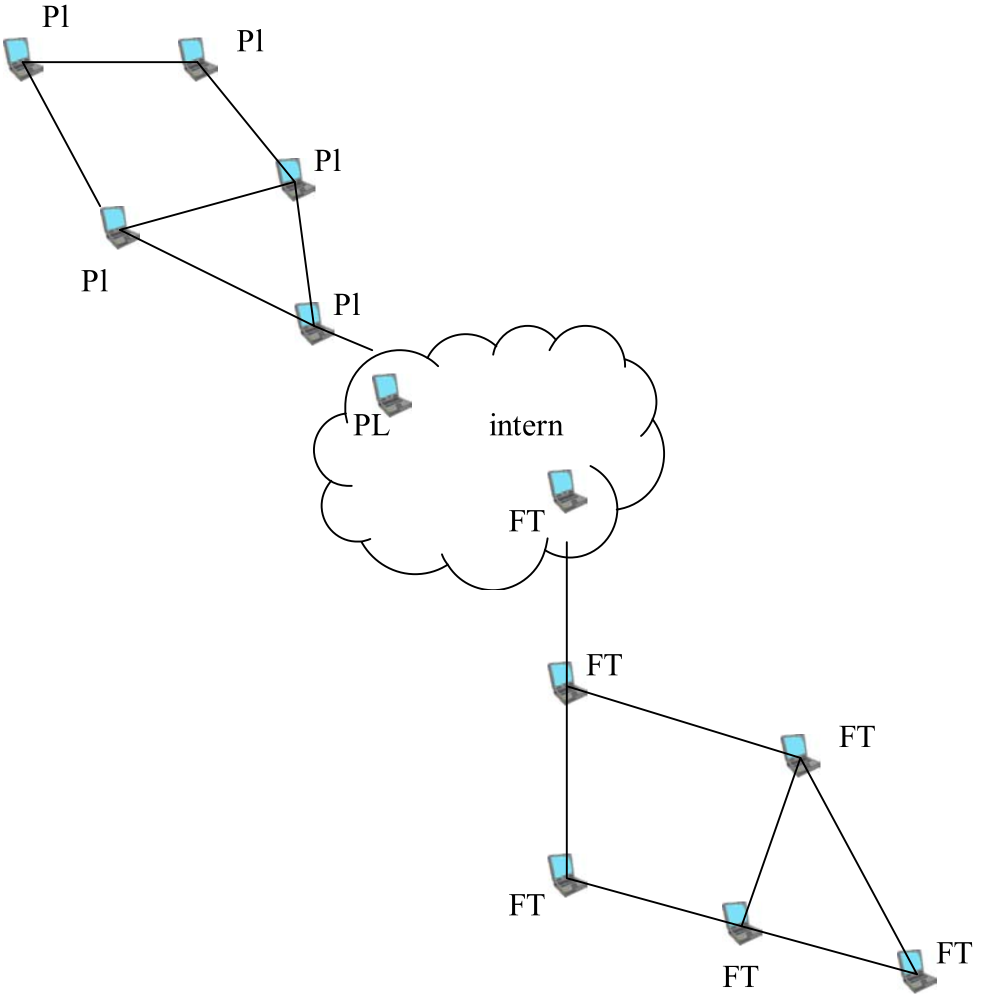
4.2.3. Multihomed

EXP_MULT_01	• Initiate an application. • Create a simulated path failure (in the path used by the traffic of the application). When REAP is deactivated, check that the application traffic is disrupted.
EXP_MULT_02	• Initiate an application with all the multihoming functionalities enabled. • Create a simulated path failure (in the path used by the traffic of the application). Check that REAP is able to detect the path failure (reaches the exploration path state).
EXP_MULT_03	• Initiate an application with all the multihoming functionalities enabled. • Create a simulated path failure (in the path used by the traffic of the application). • Check that REAP explores new paths (probe messages can be used to detect this) and chooses a working one.
EXP_MULT_04	With all the multihoming functionalities enabled: • Send ping traffic between two nodes. • Create a simulated path failure (in the path used by the traffic of the

	application). • Check that the traffic flow continues in the new path (packets can be captured to check new locators, ping working again (ECHO ICMP messages received).
EXP_MULT_05	Initiate a TCP application with all the multihoming functionalities enabled: • Create a simulated path failure (in the path used by the traffic of the application). • Check that the traffic flow continues in the new path (packets can be captured to check new locators, application working again). • Repeat with a UDP application.

4.2.4. Wireless ad hoc

EXP_ADHO_01	Tests will been performed between ad hoc nodes, located in Lannion, in France, and in Warsaw, interconnected though the PlanetLab network in order to evaluate both the integration of ad hoc networks within PlanetLab, as well as to identify any delays and bandwidth limitations. In addition QoE information will be evaluated. The tests aim at evaluating the quality of video streaming with the source and/or the destination located in an ad hoc network.



The tests are performed for video streaming over HTTP and RTSP:

- The tests are performed in the same way in TP's network and in FT's network. The video server is located in the same country than the ad hoc network and is on a wired network (PLI for polish tests and FTI for French tests). The client is in the ad hoc network. The tests are carried out with several hops between the server and the destination.
    - One hop between client and gateway
    - 2 hops between client and gateway
    - 3 hops between client and gateway

- The tests are performed in the same way in TP's network and in FT's network. The video server and the client are on the ad hoc network. The tests are carried out with several hops between the server and the destination.
    - One hop between client and server
    - 2 hops between client and server
    - 3 hops between client and server

	• The server is located in TP's ad hoc network and the client is located in FT's ad hoc network. The total number of hops between the server and the gateway plus the client and the gateway are: ▪ 2 hops ▪ 3 hops ▪ 4 hops

4.2.5. Emulation

In order to test the operation of the DBox we can run the following tests.

EXP_EMUL_01	Manual configuration of the box: This test is aimed at testing the procedures to configure the PLC database with information to connect the node to the DBox, and pass configuration information to the node and the DBox. In this test we do the following steps: • Using the PLC interface, add a node to the system (using the menu ….) • Using the PLC interface, add a DBox to the system (using the menu ….) • Using the PLC interface, declare a connection between the two (using the menu ….) • Download the image for the DBox and store it onto the hard drive or flash memory card of the DBox. • Start the DBox, which at this point will contact the PLC to update its databases (including tables of users, passwords, keys) • Wait for the node to update its tables and become aware of the presence of the DBox.
EXP_EMUL_02	Basic operation of the DBox: This test is aimed at testing that the connection set up in the previous step works properly and does not alter traffic in absence of external configuration. This is done as follows: • Remove the DBox and replace it with a cross link • Run a simple test, e.g., ping or something else, and record the results. • Put the DBox back in place • Re-run the previous ping test, and check that the delays and bandwidth does not change significantly (an additional delay in the order of 1ms in RTT is considered acceptable; bandwidth should be 30Mbit/s or more).

EXP_EMUL_03	Emulation at work: This test is aimed at testing that the node can actually configure the DBox and that the latter responds to the commands. This is done as follows: • Open a window on the node, running a ping test, and keep it running. • Issue a 'netconfig' command, specifying a bandwidth limitation. • Observe that the delay reported by the ping test changes accordingly to the bandwidth and packet size.

4.3. Validation tools

This section presents the validation tools needed to run the experiments defined in the previous section. Some experiments will be validated. These tools will also help to collect the results data during the validation process.

VAL_WIMA_01	UCL will use traceroute and ping tools in order to measure the WiMAX network. Furthermore some dummynet examples will be used to correlate the emulated model with the WiMAX measurement.
VAL_MULT_01	UC3M will use a network analyzer such as Wireshark in order to run their experiments.
VAL_ADHO_01	FT will use tcpdump to measure the delay, the jitter and the packet loss. The NTP protocol is needed to synchronise the network nodes. In addition the RSSI will be gotten to measure the link stability. The RSSI could be gotten from the iwspy tool or directly from the driver.
VAL_EMUL_01	UniPi will use standard network tools as ping and netconfig to measure the delays and bandwidth.

5. Monitoring validation

5.1. Passive Monitoring

The goal of the passive monitoring validation is not to propose new overlay applications but rather to understand the ease of use of the OneLab passive monitoring component and its performance.

5.1.1. Requirements

REQ_PMON_01	Availability of monitoring systems on various locations in the OneLab infrastructure
REQ_PMON_02	Availability of the passive monitoring service (CoMo) on the OneLab infrastructure
REQ_PMON_03	Availability of a web-based visualisation interface to receive query results and present them to the user.

5.1.2. Experiments and tests

EXP_PMON_01	Use Jaiswal03 [1] and Jaiswal04 [2] techniques for measuring the passive monitoring performance in terms of response time and processing overhead.
EXP_PMON_02	Demonstration of rapid definition and implementation of a custom query by a generic user (i.e., not a passive monitoring component developer)
EXP_PMON_03	User authentication and query execution on selected CoMo nodes
EXP_PMON_04	Demonstration of protection against malicious users

5.1.3. Validation tools

VAL_PMON_01	Web browser
VAL_PMON_02	Compiler

5.2. Topology monitoring

In order to test the topology monitoring component provided by OneLab in a heterogeneous environment, we will divide the topology monitoring validation into two parts: validation of wired network topology monitoring and validation of ad hoc wireless network topology. UPMC and UCL will be focused on the wired topology monitoring while testing the topology information component API [D3B.1] and the active measurement subcomponents [D3B.2] developed on the wired network. Then FT will test the active measurement subcomponent [D3B.2] and the Topology Information API on an ad hoc wireless network with a gateway linked to the wired network. Thus, we will have a better vision of the performance provided by the OneLab topology monitoring component in a heterogeneous network.

5.2.1. Requirements

5.2.1.1. Requirements for wired topology monitoring

REQ_TMON_01	Provide topological data to pass through given nodes that perform specific processing.
REQ_TMON_02	Use the information provided by the topology monitoring component to react to changes in underlying network conditions.
REQ_TMON_03	Test the active measurement subcomponent, traceroute@home, and send probe packets into the network
REQ_TMON_04	Test the AS-level [D3B.3] subcomponent on the private PlanetLab.

5.2.1.2. Requirements for ad hoc network monitoring

REQ_TMOA_01	Make network QoS information available to an application via a database (bandwidth, packet loss, RSSI, volume of data, etc..). The QoS information can be detailed for each application.
REQ_TMOA_02	Make network QoS information available to users via a GUI (bandwidth, packet loss, volume of data, CPU consumption, volume of management packets, etc..)
REQ_TMOA_03	Provide topological data

5.2.2. Experiments and tests

5.2.2.1. Experiments for wired topology monitoring

EXP_TMON_01	Test the reconfiguration of the overlay routing tables by application when changes occur in the underlying network.
EXP_TMON_02	Compare the path obtained by the new measurement tools (traceroute@home) and the existing measurement tool (Scriptroute, RIPE-RIS) to a fixed known network topology.
EXP_TMON_03	Compare the results obtained by the Topology Information Component API [D3B.1] with the topology data extracted from a Cisco tool called NetFlow [10]
EXP_TMON_04	Analyse the results of the AS-level subcomponent on a known AS and test the behaviour of the tracking of BGP announcements.

5.2.2.2. Experiments for ad hoc network monitoring

EXP_TMOA_01	FT will test the monitoring tool (QoS information for an application and QoS information for users (GUI)) with the tests described in section 5.2.4
EXP_TMOA_02	FT will test the wired topology monitoring tool on the wireless ad hoc network to provide topological data.

5.3.1. Validation tools

VAL_TMON_01	UPMC and UCL will validate the Topology monitoring component for a wired network with known topological tools as Skitter [11], RIPE-RIS [12]and Team Cymru [13] tools, such as a whois client or Asinuse.
VAL_TMOA_02	FT will validate the Topology monitoring component for wireless ad hoc networks with wireless tools Iwspy for the RSSI Information, routing table editing and by comparison with the tests results of section 5.2.4

6. Iterative integrated OneLab validation

6.1. Main goals and features

One of the important features of the integrated test framework for validation of the OneLab platform is that it execute automatically on a daily basis. This allows us to incrementally validate the daily changes to the OneLab code base.

The test framework is based on the API methods and executed as a package of remote requests; for that the API server connection is used for connecting and communicating with the MyPLC shell (the API libraries). This feature’s objective is to perform early detection of the issues that a user would face when remotely interacting with the OneLab test bed.

As the integration of new components in the core system should not cause performance to deteriorate, the main goal of the iterative integrated OneLab validation is to provide:

- An iterative and automatic mechanism to test the OneLab code base and then validate it based on the existing result from the test framework.
- An automatic solution to precisely simulate all the steps of creating a new MyPLC (testbed) [9], the result will be a realistic testbed with some active nodes ready for use.

- An independent iterative way to debug and potentially validate the most important behaviour of the OneLab testbed, taking into account the applications that can potentially be run there.

To achieve all these objectives, the test framework provides tests for validating the various functionalities of the OneLab testbed. An overview of what is currently tested is described in the next sections.

6.2. Requirements

REQ_IIOV_01	Initial installation and system configuration.
REQ_IIOV_02	Use of the API methods in order to create real objects.
REQ_IIOV_03	Configuration and installation of real nodes.
REQ_IIOV_04	Creation and deployment of slices.

6.3. Experiments and tests

In the next sections, we describe all the steps taken by the test framework for validation purposes.

6.3.1. MyPLC installation and configuration

EXP_ IVIC _01	• Uninstall of the old MyPLC based on the latest version of successful compiled build code.
EXP_ IVIC _02	• Clean of all traces related to the old MyPLC under /plc/root and /plc/data. • Reinstall of a new MyPLC using rpm install with the appropriated path to the latest version on the http://build.one-lab.org/onelab/.
EXP_ IVIC _03	• Mount the MyPLC. • The configuration of the new installed MyPLC is done silently by the way of the plc-config-tty: <plc> -bash-3.00# plc-config-tty Enter command (u for usual changes, w to save, ? for help) l ========== Category = PLC_WWW PLC_WWW_HOST = the hostname of the box handling the web part(server, html pages …) ========== Category = PLC PLC_ROOT_USER = root@myplc-box PLC_NAME = The MyPLC name PLC_ROOT_PASSWORD = ******** ========== Category = PLC_BOOT

	PLC_BOOT_HOST = the hostname of the box handling the boot of the new myplc ========== Category = PLC_MAIL PLC_MAIL_SUPPORT_ADDRESS = support@onelab.org ========== Category = PLC_API PLC_API_HOST = the hostname of the box where is stored the API library PLC_API_MAINTENANCE_SOURCES = 127.0.0.1 PLC_API_MAINTENANCE_PASSWORD = 9f680107-b5be-4d22-9c9c-44265ff94706 ========== Category = PLC_DB PLC_DB_HOST = the hostname of the box where the Data Base is stored. PLC_DB_PASSWORD = 549b41dc-9bd1-4aa5-904a-66367f525556 ========== Category = PLC_NET PLC_NET_DNS2 = The DNS IP address PLC_NET_DNS1 = The second DNS IP address • Restart the newly-installed and configured MyPLC.

6.3.2. MyPLC population

6.3.2.1. Sites and Persons Management:

EXP_IVPM_01	• Add a Site and configure it with all the needed information (Name, description, Address). This Site is owned by the root user.

EXP_IVPM_02	• Add Persons:
	  o PI: principal investigator, responsible for overseeing the site's participation in OneLab.
	  o Tech: Technical, responsible for installing or administering OneLab nodes.
	  o User: researchers, educators, and students who run experiments, develop applications, or maintain services on OneLab.
	• Add a Role for each Person previously added (PI, Tech and User).
	• Add Persons to the previous created Site.
	• Enable the added Persons.
	• Upload keys for those Persons (for later eventual authentication).

6.3.2.2. Nodes Management:

EXP_IVNM_01	• Create two Nodes: test*@onelab.org. • The PI and the Tech Persons each create one node and install it. • Add the installed Nodes to the Site. • Update the Node's status from "install" to "reinstall" (to avoid to manually confirm the Nodes 's installation later on). • Add the regular user to the two Nodes.

6.3.2.3. Slices Management:

EXP_IVSM_01	• Create a Slice (ts_slicetest1). • Add the Slice to the Nodes.

	• Add the regular user to the newly created Slice.

6.3.3. Node Installation and simulation:

For this step, we require to have the VMware Player 1.0.4 installed, a free desktop virtualization that makes it easy to operate virtual machines (Nodes in our test framework):

EXP_IVNI_01	• Kill any other instance of previous virtual machines running there (it's generally the case because we intentionally leave the Nodes running after the tests complete). • Run the Node installation simulation via the VMplayer's virtual machines.
EXP_IVNI_02	• Iteratively check the Node's status all along the install steps, until the install succeeds or a timeout expires.

6.3.4. Test the testbed access:

The MyPLC testbed is running now with two Nodes ready for use:

EXP_IVTA_01	• Testing the access to the Slice through an SSH connection. • Restart the node manager on the two nodes, to refresh the information exchanged with the MyPLC previously. ssh –i plc-root-private-key root@test*.one-lab.org service nm restart • Access slice and ask for the date on the Vservers dedicated to the user via: ssh –i user-private-key SliceName@test*.one-lab.org date

6.4. Integration with the build system

The build repository is browsed at http://build.one-lab.org/onelab/ that gets updated after each build.

The build system is instrumented with tags related to each run:

--onelab.build.ok: successful run of the OneLab code base.

--onelab.ok: successful run of the test framework.

--onelab.broken: failure on the test framework. This file contains the details of the last code's instructions causing the bugs.

2007.07.15--onelab.build-ok	15-Jul-2007 04:51	0
2007.07.15--onelab.ok	15-Jul-2007 04:57	0
2007.07.15--onelab/	15-Jul-2007 04:51	-
2007.07.17--onelab.broken	17-Jul-2007 04:57	20K
2007.07.17--onelab.build-ok	17-Jul-2007 04:50	0
2007.07.17--onelab/	17-Jul-2007 04:50	-

Display from the build system repository

All the steps details of each run are listed on the log-build.txt into the appropriate OneLab directory:

```
=========>Try to connect to the ts_slicetest1@test2.one-lab.org at 04:57:13
+ ssh -i /root/.ssh/slices.rsa ts_slicetest1@test2.one-lab.org echo 'The Actual
Time here is;' date
The Actual Time here is
Sat Jul 14 04:57:23 UTC 2007
=========>connected to the ts_slicetest1@test2.one-lab.org--->
all is alright
+ '[' 0 '!=' 0 ']'
+ set +x
+ touch /build/onelab/2007.07.14--onelab.ok
+ '[' -n onelab-build@one-lab.org ']'
+ echo http://build.one-lab.org/onelab/2007.07.14--onelab
++ date
+ mail -s 'Successfull build in /build/onelab/2007.07.14--onelab' onelab-
build@one-lab.org
+ echo 'Completed on Sat Jul 14 02:57:14 UTC 2007'
+ cd /build/onelab
+ rm -f latest
+ ln -s 2007.07.14--onelab latest
+ exit 0
```

The log-build's tail of a successful test framework.

7. Generalized PlanetLab Europe validation

7.1. Global process

This phase of the validation process is the ultimate step before delivering PlanetLab Europe as a stable and secure platform. This validation has to be done in a real environment and with the latest build of MyPLC. Thus we will not use any simulation tool to validate this part. Every test will be performed on the hosting server used by the PlanetLab Europe platform. We will have the ability to use PlanetLab tools and services to test the experiment scenarios in real time.

7.2. Requirements

7.2.1. MyPLC installation and configuration for PlanetLab Europe

REQ_ GVIC _01	Provide a complete and stable version of MyPLC RPM package that match with a Fedora Core distribution.

REQ_ GVIC _02	Check the good performance of the installation and uninstallation process of MyPLC on PlanetLab Europe server.
REQ_ GVIC _03	Verify the integrity of files, directories and services involved in MyPLC.
REQ_ GVIC _04	Verify that MyPLC is working correctly on PlanetLab Europe server: • Start MyPLC • Stop MyPLC • Disable and Re-enable automatic startup • Change the configuration
REQ_ GVIC _05	Evaluate the behaviour and the access of the API, web server, boot server and the database server.

7.2.2. PlanetLab Europe population

7.2.2.1. Sites and Persons Management:

REQ_GVPM_01	Check the availability for a root user to create a site and assign role (PI, Tech, User) to real users through the XML-RPC API or through the web server.
REQ_GVPM_02	Evaluate the creation of real users and their access and authentification.
REQ_GVPM_03	Verify the good behaviour of the registration process for a site and a regular user.

7.2.2.2. Node Management:

REQ_GVNM_01	Test the creation of nodes via the PlanetLab Europe API.
REQ_GVNM_02	Check the availability for responsible user to administer nodes.
REQ_GVNM_03	Evaluate the installation and uninstallation process for PlanetLab Europe Nodes
REQ_GVNM_04	Verify the availability to access to New Environment nodes.

7.2.2.3. Slice Management*:*

REQ_GVSM_01	Control the opportunity to create a slice.
REQ_GVSM_02	Test the management process for different slices.

7.2.3. Federation:

REQ_GVFE_01	Check the federation process while installing PlanetLab Europe.
REQ_GVFE_02	Control the read access on PLC database and the good behaviour of peering methods. .

7.2.4. PlanetLab Europe Access:

REQ_GVAC_01	Provide an access to the web server with multiple browser**s.**
REQ_GVAC_02	Check the availability of SSH connections to different slices.

7.2.5. Monitoring services

REQ_GVMO_01	Provide a global view to users and administrators of slice and node behaviour.
REQ_GVMO_02	Use monitoring tools to respond to users and administrator complaints.
REQ_GVMO_03	Put in place a monitoring tool to control PlanetLab Europe server health.
REQ_GVMO_04	Check the availability of the Topology Monitoring service on PLE.
REQ_GVMO_05	Verify the availability of the Passive Monitoring component on PLE.

7.3. Experiments and tests

7.3.1. MyPLC installation and configuration for PlanetLab Europe

EXP_ GVIC _01	- Installation of the latest MyPLC version of successful compiled build code.
EXP_ GVIC _02	- Uninstallation of the previous MyPLC installed - Clean of all traces related to the old MyPLC under /plc/root and /plc/data. - Reinstallation of a new MyPLC for PlanetLab Europe using rpm install with the appropriate path to the latest version on the http://build.one-lab.org/planetlabeurope/latest/
EXP_ GVIC _03	Verify that the main files and directories are installed: • Root filesystem: /plc/root.img • Mount point: /plc/root • User data: /plc/data • Configurations files: /etc/planetlab

	• Database: /var/lib/pgsql • Node package update: /var/www/html/download • XML files: /var/www/html/xml • Root user: /root
EXP_ GVIC _04	• Start **MyPLC:** # service plc start • Stop MyPLC**:** # service plc stop • Disable automatic startup of MyPLC**:** # chkconfig plc off • Re-enable automatic startup of MyPLC**:** # chkconfig plc on • Mount the MyPLC**:** # service plc mount Mounting PLC: [OK] • Change the configuration of the new installed MyPLC using plc-config-tty tool**:** # chroot /plc/root su - <plc> # plc-config-tty Config file /etc/planetlab/configs/site.xml located under a non-existing directory Want to create /etc/planetlab/configs [y]/n ? y Created directory /etc/planetlab/configs Enter command (u for usual changes, w to save, ? for help) u == PLC_NAME : [PlanetLab Test] PlanetLabEurope == PLC_SLICE_PREFIX : [pl] ple == PLC_ROOT_USER : [root@localhost.localdomain] root@planet-lab.eu == PLC_ROOT_PASSWORD : [root] ******** == PLC_MAIL_ENABLED : [false] true == PLC_MAIL_SUPPORT_ADDRESS : [root+support@localhost.localdomain] support@planet-lab.eu == PLC_BOOT_HOST : [localhost.localdomain] planet-lab.eu == PLC_NET_DNS1 : [127.0.0.1] 138.96.250.248 == PLC_NET_DNS2 : [None] 138.96.250.249 Enter command (u for usual changes, w to save, ? for help) w Wrote /etc/planetlab/configs/site.xml Merged /etc/planetlab/default_config.xml
EXP_ GVIC _05	• Access to the API on PlaneLab Europe server via the command *plcsh* • Connect to the web server : http://planet-lab.eu/ and try to login as root. • Access to the database server and browse the table installed. • Test the boot server with the API function AddBootState.

7.3.2. PlanetLab Europe population

7.3.2.1. Site and Person management:

EXP_GVPM_01	• Connect to the web server as root user. • Add a Site and configure it with all the needed information (Name, description, Address). • Specify role (PI, Tech, User) for a real user • Check that this Site is owned by the root user. • Do the same operation using the API command: AddSiteAddress, AddSite.

EXP_GVPM_02	• Add Persons:
	  o PI: principal investigator, responsible for overseeing the site's participation in OneLab.
	  o Tech: Technical, responsible for installing or administering OneLab nodes.
	  o User: researchers, educators, and students who run experiments, develop applications.
	• Add a Role for each Person previously added (PI, Tech and User).
	• Add Persons to the previous created Site.
	• Enable the added Persons.
	• Upload keys for those Persons (for later eventual authentication).

EXP_GVMP_03	• Create a new site with the site registration form • Validate the site registration process
EXP_GVMP_04	• Create a new regular user account for a specific site. • Check the good behaviour of the validation user account process.

7.3.2.2. Node management:

EXP_GVNM_01	• Create several Nodes by using Add Node under the Nodes tab on the web server. • Test the creation of nodes with PI and the Tech persons authentification. • Install the nodes in real condition. • Fill the node configuration parameters on the Node Details page. • Create a Boot CD image from PlanetLab Europe. • Verify that nodes installed boot correctly. • Add the installed Nodes to the Site. • Update the Node's status from "install" to "reinstall" (to avoid manually confirming the Node's installation later on). • Add regular users to the two Nodes.
EXP_GVNM_02	• Try to administrate the nodes created as PI.

EXP_GVNM_03	• Access nodes via ssh (ssh –i /etc/planetlab/root_ssh_key.rsa root@node) • Check the good behaviour and stability of the nodes via the log file for Node Manager: /var/log/pl_nm
EXP_GVNM_04	Verify the access and the performances of the New Environment nodes: • WiMAX • UMTS • Wireless ad hoc • Multihomed • Emulated.

7.3.2.3. Slice management:

EXP_GVSM_01	• Create different Slices (The PI must have the slice creation permission). • Add Nodes to slices by clicking *Manage Nodes* on the *Slice Details* page on the web server. • Add the Slice to the Nodes. • Add regular users to the newly created Slices.
EXP_GVSM_02	Test to force slice creation on a node: • Connect to the PlanetLab Europe server as PI or root. • Update slices.xml with the *crond* *service plc start crond* • Launch the slice creation service on a particular node*:* *ssh –I /etc/planetlab/root_ssh_key.rsa root@node* *vserver pl_conf exec service pl_conf restart*

7.3.3. Federation:

EXP_GVFE_01	Check that PlanetLab Europe platform can access to PlanetLab nodes located in the United States.
EXP_GVFE_02	• Verify that PlanetLab Europe can get information of nodes, sites and persons that belongs to PlanetLab in the United States. • Check that PlanetLab Europe don't have write access into the PlanetLab database located in the United States.

7.3.4. PlanetLab Europe access:

EXP_GVAC_01	Test the access to the PlanetLab Europe web server with several browsers: • Internet Explorer 7 • Firefox

	• Safari
EXP_GVAC_02	• Test the access to the Slice through an SSH connection. • Restart the node manager on the two nodes, to refresh the information exchanged with the MyPLC previously. ssh –i plc-root-private-key root@node service nm restart • Access slice and ask for the date on the Vservers dedicated to the user via: ssh –i user-private-key SliceName@node date

7.3.5. Monitoring services

REQ_GVMO_01	• Alter a specific node that was installed. • Verify with the CoMon [5] interface the characteristics of the altered node. • Do the same experiment with a slice.
REQ_GVMO_02	• Simulate an abuse access on a slice located on a specific node by generating one SSH access per second on this node address. • Record the time when this simulated attack was done. • Track the source of the attack using PlanetFlow [6] service.
REQ_GVMO_03	Use Nagios [7] software to monitor the heath of PlanetLab Europe server: • Check PlanetLab Europe memory. • Check load. • Check the http server. • Ping tests. • Check users.
REQ_GVMO_04	Verify the performance and the access of the Topology Monitoring component: • Query a path toward a source to a destination • Query BGP information.
REQ_GVMO_05	Try to query the passive monitoring CoMo component and verify its availability

7.4. Validation tools

VAL_GPEV_01	UPMC will need to validate PLE web server with several browsers, such as Internet Explorer 7, Firefox, Safari.
VAL_GPEV_02	UPMC will use CoMon [5], PlanetFlow [6], Nagios and CoMo as monitoring tools.

8. Conclusion

In this document we set out the validation process for the OneLab platform, from the unitary level up to the general level of the validation platform. This document will serve as the reference for validating the OneLab platform. The experiments and measurements described in the validation plan have to match the requirements defined in order to ensure the quality of all tested components. The next step of the validation process will be to collect the experiment results into a Validation Report D5.2, due on Month 24, in order to demonstrate the stability and robustness of the PlanetLab Europe platform.

9. Relation to other deliverables

This document, the D2.6 Validation Plan, is produced by the Integration work package (WP2). The final goal of WP2 is to provide to WP1 Operations a robust, stable and secure publicly operational PlanetLab platform. As the validation plan focused on the validation of WP2, WP3, and WP4, the relationship of each deliverable within these work packages is described in the following table. Moreover, the validation process is part of the validation workpackage (WP5) and starts in Month 18 and runs through Month 24. The result of the validation process will be the validation report that will collect all the data gathered during the experiments.

Nb	DOCUMENT	Partner
D5.2	Validation report	UPMC
D4A.1	WiMAX component	UCL
D4B.1	UMTS node	CINI
D4B.2	UMTS gateway	CINI
D4.C.1	Multihoming architecture document	UC3M
D4C.2	Multihoming mechanisms document	UC3M
D4C.3	Multihoming component	UC3M
D4D.1	PlanetLab compliant ad hoc network	FT
D4D.2	Gateway between ad hoc network and PlanetLab	FT
D4.E.1	Modified dummynet code	UniPi
D4E.2	Integrated dummynet and PlanetLab	UniPi
D4E.3	Emulation component	UniPi
D3.A.2	Passive monitoring component	Quantavis
D3.B.1	Topology information component API	UPMC

D3.B.2	Active measurement subcomponent	UPMC
D3B.3	AS-level measurement subcomponent	UPMC

10. References

[1]	S. Jaiswal, G. Iannaccone, C. Diot, J. Kurose, D. Towsley, ‘‘Measurement and Classification of Out-of-Sequence Packets’’, In Proc. Infocom, 2003.
[2]	S. Jaiswal, G. Iannaccone, C. Diot, J. Kurose, D. Towsley, ‘‘Inferring TCP Connection Characteristics Through Passive Measurements’’, In Proc. Infocom, 2004.
[3]	Akihiro Nakao, Larry Peterson, and Andy Bavier, ‘‘A Routing Underlay for Overlay Networks’’, In Proc. ACM SIGCOMM, 2003.
[4]	Linux VServers Project: http://linux-vserver.org/.
[5]	CoMon a monitoring infrastructure for PlanetLab: http://comon.cs.princeton.edu/
[6]	PlanetFlow**,** a network auditing service for PlanetLab: http://planetflow.planet-lab.org/
[7]	Nagios a monitoring solutions for hosts, services, and networks: http://www.nagios.org/
[8]	Jeremy Sugerman, Ganesh Venkitachalam and Beng-Hong Lim , "Virtualizing I/O Devices on VMware Workstation’s Hosted Virtual Machine Monitor" , In Proc. USENIX Annual Technical Conference VMware virtual machine technology: http://www.vmware.com
[9]	MyPlC User’s Guide: http://www.planet-lab.org/doc/myplc
[10]	Cisco Systems, *Cisco CNS NetFlow Collection Engine*, http://www.cisco.com/en/US/products/sw/netmgtsw/ps1964/products_user_guide_chapter 09186a00801ed569.html , 2004