



FUTURE INTERNET TESTBEDS  
EXPERIMENTATION BETWEEN  
BRAZIL AND EUROPE



Grant Agreement No.: 288356  
CNPq Grant Agreement No.: 590022/2011-3

## FIBRE-EU

Future Internet testbeds/experimentation between BRazil and Europe – EU

Instrument: *Collaborative Project*  
Thematic Priority: *[ICT-2011.10.1 EU-Brazil] Research and Development cooperation, topic c) Future Internet – experimental facilities*

### D3.3 Final version of the enhanced OFELIA control framework software

Author: WP3  
Revised by: Sebastià Sallent (UPC)  
Due date of the Deliverable: Month 34  
Actual submission date: 31/03/2014  
Start date of project: June 1<sup>st</sup> 2011 Duration: 34 months  
version: v.1.0

Project co-funded by the European Commission in the 7 <sup>th</sup> Framework Programme (2007-2013)		
Dissemination Level		
PU	Public	✓
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

\* This project has received funding from the European Union's Seventh Framework Programme for research, technological development and demonstration



**D3.3 - Final version of the enhanced OFELIA control framework software**

Doc FIBRE – D3.3-v1.0

Date 31/03/2014

<b>FP7 Grant Agreement No.</b>	288356
<b>CNPq Grant Agreement No.:</b>	590022/2011-3
<b>Project Name</b>	Future Internet testbeds/experimentation between BRazil and Europe – EU
<b>Document Name</b>	FIBRE-D3.3-v1.0
<b>Document Title</b>	D3.3 - Final version of the enhanced OFELIA control framework software
<b>Workpackage</b>	WP3
<b>Authors</b>	Carlos Bermudo (i2CAT) Roberto Monno (NXW) Mayur Channegowda (UNIVBRIS)
<b>Editor</b>	Carlos Bermudo (i2CAT)
<b>Reviewers</b>	Sebastià Sallent (UPC)
<b>Delivery Date</b>	31/3/2014
<b>Version</b>	V1.0





***D3.3 - Final version of the  
enhanced OFELIA control  
framework software***

Doc FIBRE – D3.3-v1.0

Date 31/03/2014

## Abstract

This deliverable document complements the information about the enhancements performed over the OFELIA control framework provided in previous documents MS6 [8] and MS9 [9].

Specifically, it contains the enhancements performed and bugs solved since October 2013, date of the previous milestone. Each enhancement is provided in context regarding the project objectives described in the DoW or due to later requirements appeared during the project developments.

The bugs solved are from utilization feedback of the infrastructure or discovered by developers during tests.





## TABLE OF CONTENTS

1	Acronyms .....	7
2	Scope.....	8
3	Reference Documents .....	9
4	OCF enhancements .....	10
4.1	Simplified FlowSpace request process .....	10
4.2	Automatic VLAN assignment .....	11
4.2.1	VLAN Manager .....	12
4.3	Automatic FlowSpace granting.....	14
4.3.1	Integration.....	14
4.4	VM management improvement .....	15
4.4.1	Integration.....	15
4.5	OFAM SFA adaptation .....	15
4.5.1	Integration.....	16
4.6	Version tracker for AMs.....	16
4.7	Increased support for different Virtual Machine images .....	17
4.8	Paving the way for FlowVisor 1.0 and upwards .....	17
4.9	Management of ROADM devices.....	18
4.9.1	OpenNaaS extensions .....	19
4.9.2	ROADM Aggregate Manager .....	20
4.9.3	ROADM Expedient Plug-in.....	22
4.10	OCF Extension to the Optical Layer .....	23
4.10.1	Optical Flowvisor .....	25
4.10.2	Optical FOAM .....	25
4.10.3	Construction and Quality verification of optical slices.....	25
4.10.4	Extended Expedient Openflow plugin.....	26
4.10.5	PCE as a Network Function .....	28
5	Solved bugs .....	30
5.1	List_resources communication error.....	30
5.2	FlowSpace selection enhancement.....	30
5.3	Definition of the VT AM’s address ranges nor properly handled.....	30
5.4	Improved handling of VM allocation and notifications to experimenters .....	30
6	Future work .....	31
6.1	Porting OCF to latest Debian OS (Debian 7).....	31



***D3.3 - Final version of the  
enhanced OFELIA control  
framework software***

Doc FIBRE – D3.3-v1.0

Date 31/03/2014

6.2 LDAP decoupling .....31  
7 Conclusions .....32





## LIST OF FIGURES

Figure 1: Simplified FlowSpace request process ..... 11

Figure 2: Workflow of VLAN Manager ..... 13

Figure 3: Auto-approval information shown to the users..... 14

Figure 4: VM update status transitions state machine..... 15

Figure 5: Enhancements to OCF architecture for ROADM devices..... 19

Figure 6: Advertisement RSpec for ROADM resource ..... 21

Figure 7: Request RSpec for ROADM resource ..... 22

Figure 8: Reserve a cross-connection using the RODM-AM..... 23

Figure 9: Optical Flowvisor..... 24

Figure 10: Optical Flowvisor and extended OCF to the optical layer..... 24

Figure 11: PLI aware virtual optical network composition ..... 26

Figure 12: Optical Network Topology display in Expedient..... 27

**Figure 13: Optical Flowspace selection (choosing wavelengths)**..... 27

Figure 14: PCE and Bandwidth selection options when creating a new slice ..... 28

Figure 15 PCE as virtual network function (NFV) ..... 29

## LIST OF TABLES

Table 1: OpenNaaS extensions for ROADM resources ..... 20

Table 2: ROADM AM GENIv3 interface..... 21



## 1 Acronyms

AM	Aggregate Manager
CIDR	Classless Inter-Domain Routing
GUI	Graphical User Interface
MS	Milestone
MVT	Model View Template
NFV	Network Functions Virtualization
OCF	Ofelia Control Framework
OF	OpenFlow
OFAM	OpenFlow Aggregate Manager
OpenNaaS	Open Network as a Service
PCE	Path Computation Engine
PLI	Physical Layer Impairments
ROADM	Reconfigurable Optical Add-Drop Multiplexers
RSpec	Resource Specification
VLAN	Virtual Local Area Network
VM	Virtual Machine
VON	Virtual Optical Network
VPN	Virtual Private Network
VTAM	Virtualization Aggregate Manager
WDM	Wavelength Division Multiplexing
WP	Work Package
XML	eXtensible Markup Language



***D3.3 - Final version of the enhanced OFELIA control framework software***

Doc FIBRE – D3.3-v1.0

Date 31/03/2014

## 2 Scope

The document reports the work done within the T3.3 Enhancement to the OFELIA Control Framework during the period M25-M30 of the FIBRE-FP7 project which generated the version 0.8 of the software. It is important to note that this Deliverable describes the progress done with respect of MS9 Second version of the enhanced OFELIA Control Framework software [9] delivered in M24.

Section 4 starts with a list and description of the enhancements and new features added to the OCF software, describing its purpose, reason and requirements covered.


Section 5 focuses on the maintenance work performed in order to solve the most important bugs identified during the use of the OCF both by the users and by the development team (those generating an unexpected behaviour of the control framework). It also explains the solutions applied.

Following Section 6 presents a list of features found as interesting during the execution of the project and that may define the way to be followed by future work.

Finally, this document ends with some conclusions derived from the work performed.





	<p><b>D3.3 - Final version of the enhanced OFELIA control framework software</b></p>	<p>Doc FIBRE – D3.3-v1.0 Date 31/03/2014</p>
---	--	--

### 3 Reference Documents

- [1] OpenNaaS manual: <http://opennaas.org/manuals/>
- [2] OCF Architecture: D2.2 - Complete report on federation architecture of OFELIA
- [3] GENIv3 API: [http://groups.geni.net/geni/wiki/GAPI\\_AM\\_API\\_V3](http://groups.geni.net/geni/wiki/GAPI_AM_API_V3)
- [4] GENI RSpecs: <http://groups.geni.net/geni/wiki/GENIExperimenter/RSpecs>
- [5] Mantychore web site: <http://www.mantychore.eu/>
- [6] AMsoil framework: <https://github.com/motine/AMsoil/>
- [7] Hardware-assisted Virtualization definition: [http://en.wikipedia.org/wiki/Hardware-assisted\\_virtualization](http://en.wikipedia.org/wiki/Hardware-assisted_virtualization)
- [8] MS6 First version of the enhanced OFELIA control framework software
- [9] MS9 Second version of the enhanced OFELIA control framework software
- [10] FOAM: <http://groups.geni.net/geni/wiki/OpenFlow/FOAM>
- [11] OCF installation guide: <https://github.com/fp7-ofelia/ocf/wiki/Installation%20manual>
- [12] ROADM AM: <https://github.com/rmonno/RoadmAM>
- [13] Expedient plug-in for ROADM AM: <https://github.com/rmonno/ocf/tree/ofelia.expedient.roadm/expedient/src/python/plugins/opennaas>
- [14] Django framework: <https://www.djangoproject.com/>
- [15] Werkzeug WSGI library: <http://werkzeug.pocoo.org/>
- [16] Ofelia Control Framework source: <https://github.com/fp7-ofelia/ocf>
- [17] OFelia VERsioning tool: <https://code.google.com/p/ofver/>
- [18] GENI Clearinghouse: <http://groups.geni.net/geni/wiki/GeniClearinghouse>
- [19] FP7-OFELIA: <http://www.fp7-ofelia.eu/>
- [20] FP7-Fed4Fire: <http://www.fed4fire.eu/>
- [21] FP7-FELIX: <http://www.ict-felix.eu/>
- [22] S. Peng, R. Nejabati, D. Simeonidou, "Impairment-Aware Optical Network Virtualization in Single-Line-Rate and Mixed-Line-Rate WDM Networks", IEEE/OSA Journal of Optical Communications and Networking, Vol. 5, No. 4, pp. 283-293, April 2013.



## 4 OCF enhancements

As stated in previous section, this section contains a description of the enhancements performed on the OCF since last October 2013 (M25). These enhancements are focused on meeting the project aim for this WP, like easing the usage of the control framework, making it more stable, increasing the diversity of hardware available for experimentation, etc.

### 4.1 Simplified FlowSpace request process

Originally, the FlowSpace request workflow offered two different modes: i) simple mode in which the user does not need to specify any FlowSpace parameter except of the ports and switches he is intended to use; and ii) advanced mode where this possibility is available.

Attending to the need of simplifying this workflow, some enhancements were performed in the advanced mode. Currently, the screen presented to the experimenter contains a number of different forms where he/she can define a FlowSpace. However, only one of these is used most of the times. This screen has been simplified to show initially one form only and allowing more forms under demand.

Also, the user does no longer need to perform two steps (save first the FlowSpace and then click somewhere else) to get back to the slice detail page to continue experimenting; but this is taken care of with a single step, in which the FlowSpace request(s) are automatically saved.

### Set FlowSpace for Slice mySlice

1. Select OpenFlow Ports 2. **Select/Modify FlowSpace**

Each table below describes a class of traffic to be received by your controller. Each field is specified as a range. Empty cells mean any value. The tables are OR'ed together to produce a rule.

For example, if you want to receive all incoming and outgoing port 80 traffic, then you will need two tables. In the first, you would set the "TCP/UDP Destination" field to 80. In the second, you would set the "TCP/UDP Source" field to 80.

If you need to define more FlowSpaces, click "Define another FlowSpace" and an additional empty table will be displayed.

**FlowSpace 1 (saved)**

Field	From	To
MAC Source	<input type="text"/>	<input type="text"/>
MAC Destination	<input type="text"/>	<input type="text"/>
Ethernet Type	<input type="text"/>	<input type="text"/>
VLAN ID	<input type="text" value="2638"/>	<input type="text" value="2638"/>
IP Source	<input type="text"/>	<input type="text"/>
IP Destination	<input type="text"/>	<input type="text"/>
IP Protocol	<input type="text"/>	<input type="text"/>
TCP/UDP Source	<input type="text"/>	<input type="text"/>
TCP/UDP Destination	<input type="text"/>	<input type="text"/>
Ports	<div style="border: 1px solid #ccc; padding: 2px;">           Aggregate OF AM preprod: Port 3 on OpenFlow Switch 00:00:00:00:00:00:0d            Aggregate OF AM preprod: Port 65534 on OpenFlow Switch 00:00:00:00:00:00:09            Aggregate OF AM preprod: Port 1 on OpenFlow Switch 00:00:00:00:00:00:0d            Aggregate OF AM preprod: Port 2 on OpenFlow Switch 00:00:00:00:00:00:0d            Aggregate i2CAT Prod OF AM: Port 3 on OpenFlow Switch 00:10:00:00:00:00:01         </div>	
Delete?	<input type="checkbox"/>	

Expedient by Stanford University and i2CAT foundation © 2011-2013 FP7 OFELIA project & FP7 FIBRE project.  
OFELIA and FIBRE are funded by the European Commission (EC) in the 7th framework programme and supported by the FIRE initiative

**Figure 1: Simplified FlowSpace request process**

## 4.2 Automatic VLAN assignment

Federated environments are usually built on a distributed architecture. These types of environments require some degree of synchronization between federated components, usually managed by a top-level entity.

In the case of FIBRE and OCF, there are several AMs belonging to several islands which conforms a federated environment being the top level-entity, or user portal, MySlice for the case of the federation framework, or Expedient in the native OCF solution. In the case of the OpenFlow AMs, a provisioned slice through several AMs requires to share the same VLAN tag in order to isolate and identify its corresponding traffic.

Until OCF version 0.8 the FlowSpace granting was manually done since it was required that the Island Manager checked that the VLAN tag or the VLAN tag ranges in the requested FlowSpace were not being used by other experiments. In federated environments as FIBRE the situation was less flexible. The user had no information about the available or allocated VLANs on each island that formed part of his FlowSpace, so there was a large chance to request a VLAN/VLAN Range already requested by other users in some of the islands affected. In this situation, the



### ***D3.3 - Final version of the enhanced OFELIA control framework software***

Doc FIBRE – D3.3-v1.0

Date 31/03/2014

Island Managers of every island had to synchronise themselves and find a manual solution formed by free VLAN tags in all the islands affected to fulfil the user request.

A previous work was done by adding a simple mode when the user was booking his OpenFlow resources. Using this simple mode the user retrieved automatically an available VLAN, but this mode only worked when the user only used a single OpenFlow AM in his slice.

In order to simplify the work of the users as well as the Island Managers, an automatic VLAN assignment module was deployed, able to get single/ranges of VLAN(s) independently of the number of OpenFlow AMs (federated islands) selected by the user.

The user may now request the number of desired VLANs rather than asking manually for those. The VLAN manager requests the free VLANs to every OF AM where the FlowSpace is being requested.

The intersection of all available VLANs in the AMs is automatically performed, resulting in a single or a range of free VLANs. If no intersection is possible, the VLAN manager falls back to the advanced mode, where the user can manually request a set of VLANs and in that case Island Manager have to find a solution.

This feature is optional allowing activating or deactivating it at any time by choosing an option on the Opt-in Manager Admin GUI, which ensures backwards compatibility with previous versions of OCF.

#### **4.2.1 VLAN Manager**

VLAN Manager is the module devoted to find the free VLAN(s) between different OFAMs (Opt-in Manager) in a slice. VLAN Manager is able to retrieve a single VLAN tag or a VLAN range and it is enabled or disabled by configuring a setting on OFAM.

##### ***4.2.1.1 VLAN Manager Workflow***

When a user requests a FlowSpace, he may ask for a single VLAN or range of VLANs. To ease the assignation of that range, if the user only selects one OpenFlow AM, VLAN manager will return directly an available VLAN. If the user uses two or more OpenFlow AMs, the VLAN Manager contacts with every OpenFlow AM, gets all the used VLANs and finds a VLAN or VLAN range available in all the islands.

In the OCF context the VLANs are treated as a list of integers, ranging from 0 to 4095. However, due to the VPN configuration and to the island configuration performed by each Island Manager, there are VLANs that may be tagged as reserved and cannot be used for slicing since they form part of the internal networking deployment of the facility (i.e. control network VLAN). These VLANs can be configured so that VLAN Manager will not consider them as candidates.

The VLAN Manager workflow is as follows:

1. Analyse the number of OF AMs that are affected by the FlowSpace request:
  - a. If only one OFAM is used, it asks directly for a random available VLAN.
  - b. If it is more than one, the VLAN Manager retrieves the used VLANs of every Island. It was decided to retrieve the used VLAN for optimization in the data exchanged, since the list of used VLANs is usually shorter than the list of





available VLANs (close to 4096 elements, the total amount of VLANs subtracting the reserved VLANs and the ones used on slices).

2. The list of VLANs can be treated as a set of elements, where the set of used VLANs in the federated facility is the union of all the individual used VLAN sets of each island.
3. To get the free VLANs set, it is only required to make the difference between the general used VLANs set and the full available VLANs (the 4096 VLAN tags).
4. Then it is evaluated whether the user requested a single VLAN or a VLAN range.
  - a. If the user selected a single VLAN, the VLAN Manager assigns one random VLAN from the free VLAN set.
  - b. If the user requested a VLAN range, the VLAN Manager selects a range of the requested length of consecutive VLAN tags from a random position in the vector of free VLANs.

A flowchart describing the workflow of the VLAN Manager is on Figure 2:

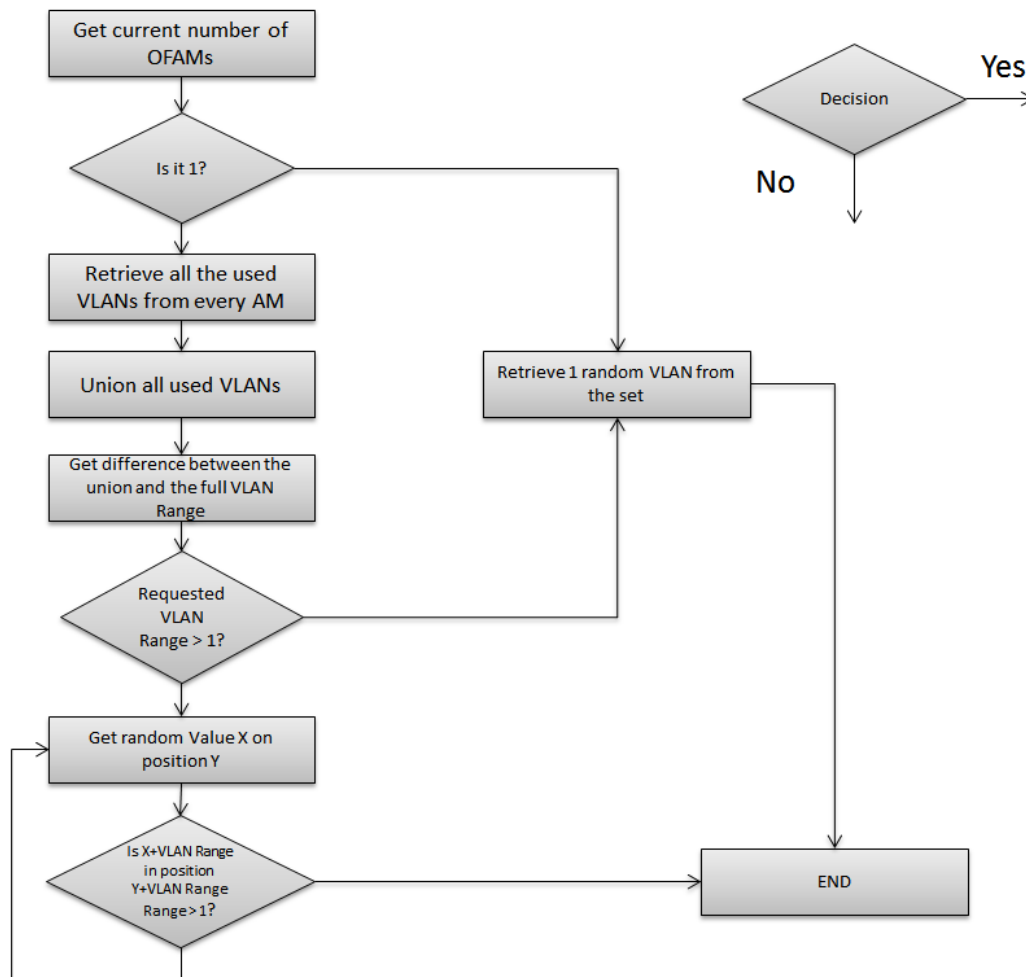


Figure 2: Workflow of VLAN Manager



### 4.3 Automatic FlowSpace granting

The automatic approval of FlowSpaces has always been an important feature which, due to the distributed architecture of the OCF islands, presented a set of constraints or requirements that obstruct the deployment of this feature: the controller URL must be unique, the VLAN(s) selected cannot collide with other slices and there might be some extra requirements of the Island Manager (i.e. restrict a certain type of used packets).

FlowVisor provides an exception-type mechanism to detect when a user is creating a slice with an already existing controller URL. Moreover, OCF assigns a specific CIDR to every island avoiding creating repeated IP ranges among islands and VTAM has track of the used and available IPs at every moment.

Regarding the collision of VLANs, the VLAN Manager, detailed in section 4.2, automatically selects a VLAN or VLAN range, while guaranteeing the isolation of the OpenFlow slices.

If the Island Manager has more requirements than the two above to grant FlowSpaces automatically, this feature can be activated or deactivated at any time using the Opt-in Manager GUI.

#### 4.3.1 Integration

This feature was integrated on Opt-in manager module in the create slice method. The Island Manager can enable or disable the automatic granting of FlowSpaces via the Opt-in GUI. If the automatic approval is selected, when the user requests a FlowSpace and fulfils all the conditions, Opt-in manager approves the FlowSpace immediately after it is requested by the experimenter. Then, it sends an email to the user and the Island Manager informing that the slice was successfully created and granted.

Also, the users can see through Expedient if the AM supports the automatic FlowSpace granting when they select the AMs.

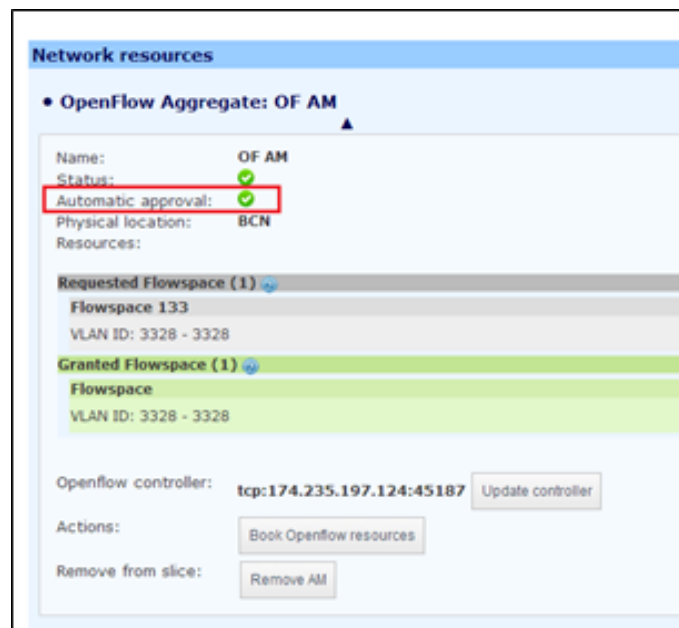


Figure 3: Auto-approval information shown to the users

## 4.4 VM management improvement

The VMs created over OCF can perform the following actions: Start, Stop, Reboot and Delete. Every time an action is executed, the VM remains in a transitory state waiting for an asynchronous message to update its status depending on the result of the operation. In this state no actions can be performed leaving the VM in a stalled state that only the Island Managers can solve by triggering management scripts on the OCF repository.

In order to improve the user experience, a new action was added: “Force Update VMs”. This action connects synchronously to the Agent that manages the VM and retrieves directly the current state of the VM in the host server and then some operations are performed (CRUD) depending on the retrieved state of the VM.

Depending on the transitory state of the VM, the new status will be updated as in the following state machine shown in Figure 4:

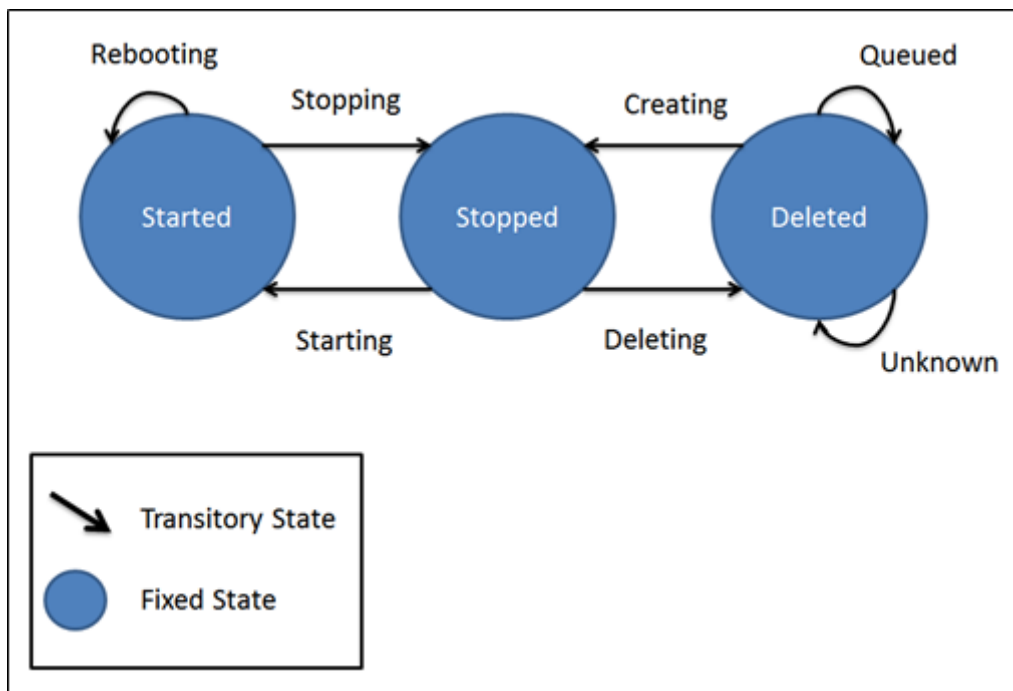


Figure 4: VM update status transitions state machine

### 4.4.1 Integration


As this feature can be triggered by users and Island Managers it was integrated on both the Virtualization AM and Expedient.

The “Force Update” action can be triggered to (forcedly) update a single VM (on Expedient and VT AM), to update all the VMs on a server (VT AM) and the whole VMs on a slice (Expedient).

## 4.5 OFAM SFA adaptation

OFAM is the name given to the OCF integration of FOAM [10]. FOAM is an OpenFlow AM developed in the GENI project used to manage the FlowVisor, as Opt-in Manager does. Nevertheless, FOAM offers more stability and is more adequate for the facility requirements. It



	<p><b>D3.3 - Final version of the enhanced OFELIA control framework software</b></p>	<p>Doc FIBRE – D3.3-v1.0</p> <p>Date 31/03/2014</p>
---	--	---

counts with an extra layer that provides more functionalities and uses the GENIv2 API. Enabling federation with GENI is another reason why it has been incorporated to OCF. FOAM development and maintenance is done outside of the project, but within the OCF development there is effort dedicated to the FOAM integration to OCF with the name of OFAM. Some of the OFAM characteristics are:

- It allows automatic VLAN assignment for a single domain, but it can't manage the VLANs from other federated environment as the VLAN Manager component of the Opt-in Manager.
- It supports automatic approval of FlowSpaces requests.
- It keeps track of the GENI URNs mapping them to an internal (and unique) identifier, making OFAM possible to manage the latest versions of FlowVisor.

On the other hand, since FIBRE and OCF are connected to MySlice portal as a part of the federation framework, OFAM needs to support the SFA API that is very similar to the GENIv2 API but handling a different type of credentials and role-base system.

#### 4.5.1 Integration

Within the integration of FOAM in OCF as OFAM, the most important achievements were:

- Adding the legacy Opt-in Manager API in order to allow Expedient portal to communicate with the OFAM using the existing OpenFlow plug-in in the Expedient GUI.
- Adding the SFA API and authentication methods to support the FIBRE federation framework. A new URL pointing to this API is exposed on the web server and the URN system is updated to suit OCF OFAM AM.

#### 4.6 Version tracker for AMs

The federated environment formed by FIBRE is a distributed architecture where the resource managers (AMs) are geographically distributed along different islands. Every time a new feature is added to one of these AMs, it may be required to synchronize the update/upgrade process of the software in every island in order to enable the new functionality (i.e. add a new method on the AM APIs). For this reason, two methods were added on every AM API: *get\_version* and *get\_info*.


*get\_version*: Retrieves the current version of the OCF installed version.

*get\_info*: Retrieves a list of parameters that the AM has configured as well as the OCF version. The list of parameters can be extended depending of the new requirements and they can be AM- type specific.

The version tracker system can be used to help performing the following tasks as well as others:

- New features can be added without synchronizing all the islands and avoid transitory bad behaviours during the process of upgrading.
- New features may need a specific OCF version to run properly that can be easily checked using one of the new methods.
- Refactors, rebuilds, dependent software updates or maintenance tasks may need all the islands to be synchronized in the version of OCF they are running. By means of the



	<p><b>D3.3 - Final version of the enhanced OFELIA control framework software</b></p>	<p>Doc FIBRE – D3.3-v1.0</p> <p>Date 31/03/2014</p>
---	--	---

Version tracker, workers running in the background can be checking that the version of the island is the required one in order to execute a specific task and trigger it automatically.

#### 4.7 Increased support for different Virtual Machine images

OCF provides a set of different virtual machine templates (images), which are used to generate the final experimenter’s VMs. In order to support the provisioning of a new VM type by OCF (e.g., another OS, a VM containing a modified Kernel, etc.), the Virtualization Aggregate Manager needs two elements:

- A configuration file: describes the VM template specifications such as RAM, interfaces information, etc.
- The image itself: the template image of the file system containing the specific software that justifies the existence of the new type of VMs. This could be the case of a project experimenting in OFELIA that requires that all of its VMs to run a custom kernel.

The new VM template integration process has hugely simplified and enhanced. The process consists on providing the following information to OCF:

1. Both configuration and image files must be accessible by the desired host server(s) (this is the hypervisor configuration file and the template VM image).
2. The OFELIA agent running in the virtualizing servers needs to communicate with the VM configurator(s). This allows the hypervisor to create the VM from the image file. The configurator triggers commands of the hypervisor that depends on the image type (such as its extension, how the image needs to be uncompressed and which virtualization technology is to be used).

Following the latest release of the Debian operating system, support for Debian 7 (Wheezy) VM images was included in the OCF.

The main problem faced to add support for this type of image was that the Agent component is running on Debian 6 servers. Paravirtualization techniques could not be used to create Debian 7 VMs, since the distribution keeps a new kernel version different from the host servers. The solution was to add a Debian 7 configurator that creates VMs using Hardware-assisted virtualization [7].


#### 4.8 Paving the way for FlowVisor 1.0 and upwards

FlowVisor is a key component within the OCF software environment. As already described in previous documentation, FlowVisor is an OpenFlow Controller with a specific objective which is to slice the FlowSpace and multiplex the different slices traffic to its individual OpenFlow controller. Moreover, as already introduced, OCF supports two different OF AMs: Opt-in Manager and OFAM (based on FOAM), both in charge of dynamically configuring the FlowVisor with the user slices.

OCF OF AMs did not support latest versions of FlowVisor, leaving some new functionalities and improvements in stability unusable. Some of the incompatibilities were originated in internal mechanism of the OF AMs. In addition to changes in the FlowVisor API, the most blocking issue was that FlowVisor slice names were composed of the name of the slice plus the site domain identifier where the FlowSpaces request took place, plus an identifier of the slice.

However, that approach had several limitations:



	<p><b>D3.3 - Final version of the enhanced OFELIA control framework software</b></p>	<p>Doc FIBRE – D3.3-v1.0</p> <p>Date 31/03/2014</p>
---	--	---

- Collisions might occur when there are two slices with the same name originating from two different islands which use the same site domain identifier. If those slices are given the same ID by the different clients (e.g. a natural number in Expedient).
- The FlowVisor slice name that is obtained may contain a lot of characters, easily getting to 40 characters and up. The newest versions of FlowVisor only support 40 characters for slice at maximum, making very difficult a conversion of the currently defined FlowSpaces in order to adapt it to the latest versions.
- The federation environment in FIBRE has slice requests both from Expedient and MySlice portals, each one with a different naming style.

To minimize and solve these problems, the new slice name for FlowVisor is a standard UUID, generated every time a new FlowSpace is requested. This UUID or identifier is mapped to the origin of the request (either a slice from MySlice or Expedient portal). In order not to affect the existing slices (those with the “older” names), retro-compatible mechanisms are implemented, allowing managing the already existing slices as well as updating their names with the new identifier.

The latter process occurs in a progressive manner, every time the experimenter updates its experiment. This is completely transparent for users and mostly transparent for administrators, which are not required to perform extra work. Summing up, this process will eventually standardize the data across each island, allowing the development team to eventually ship the latest version of FlowVisor as a part of the OCF.

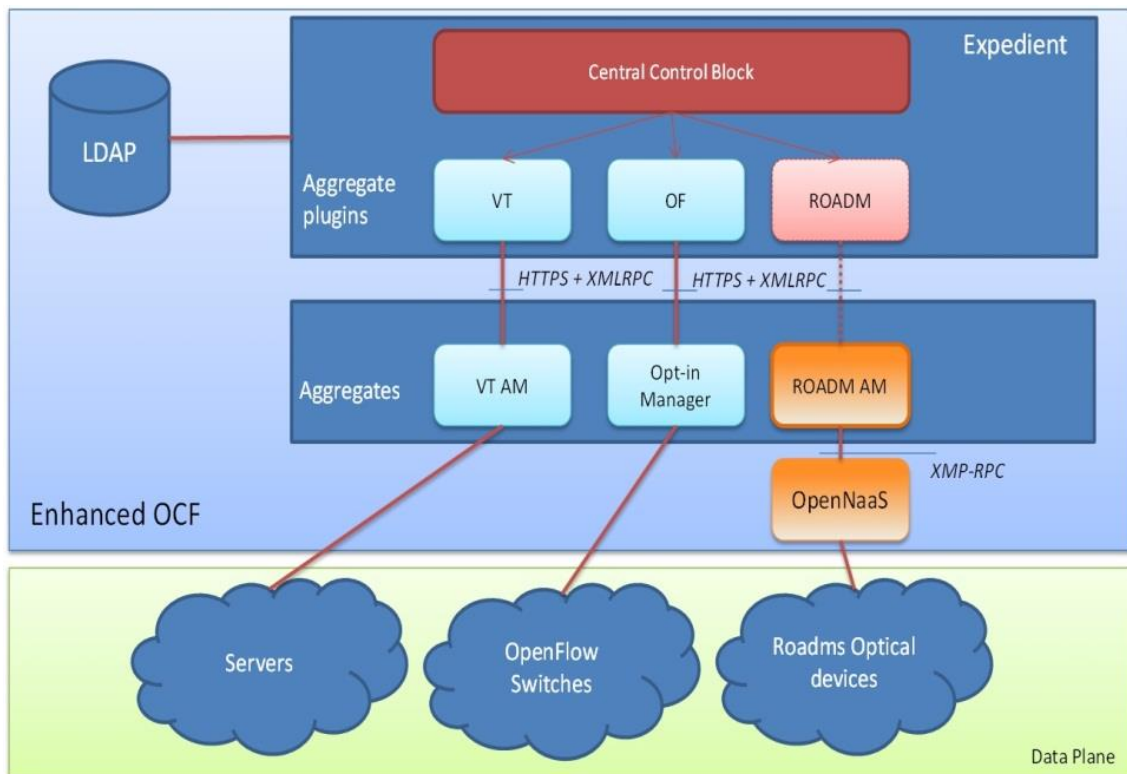
#### 4.9 Management of ROADM devices

In the following section, we describe the extensions and the enhancements to OCF provided to allow the management of the W-onesys ROADM devices.

The ROADMs are optical devices that can add, block, pass or redirect modulated infrared (IR) and visible light beams of various wavelengths in an optical fibre network. These devices can be efficiently used in all the systems which employ WDM. In short, the ROADMs devices can perform the switching operations without optical-to-electrical or electrical-to-optical conversions. To achieve this requirement, they use three main functions: add, drop and cut-through.

The *add* operation allows the generation of an outgoing IR or visible beam. The *drop* operation can be used to terminate an incoming beam. On the other hand, a beam can also be passed through the device without modification (*cut-through*). It is worth to note that the combination of these functions can allow optical signal routing of considerable complexity and that the configuration of these physical devices can be changed remotely.

Figure 5 gives an overview of the OCF architecture [2] highlighting the enhancements for the ROADMs elements.



**Figure 5: Enhancements to OCF architecture for ROADM devices**

The OCF architecture is basically composed of two layers: the **Expedient** and the **Aggregates**. The Expedient architecture is based on a Central Control Block that offers functionalities to manage users, projects and slices information, and on the Aggregates plug-ins which can present the resources details in a homogeneous way. Each Aggregate plug-in is connected to one Aggregate Manager (AM) that is responsible to control and monitor the physical resources using proper and dedicated interfaces. Please refer to [2] for further details.


In the following subsections, we briefly introduce the enhancement to OCF architecture in order to allow the management of the ROADM devices. For this purpose, three new building blocks are developed from scratch or simply extended with more specific functionalities:

- **OpenNaaS** acts as a Resource Manager to abstract the technical details of the physical resources with a high-level northbound interface.
- **ROADM AM** develops a GENiv3 API [3] to list, configure, reserve or release the underlying resources.
- **ROADM** plug-in provides a graphical representation of the ROADM resources (endpoints and labels) and helps the user to perform the cross-connections requests.

#### 4.9.1 OpenNaaS extensions

OpenNaaS [1] is an open source framework created in the Mantychore FP7 project [5] for the provisioning of different kind of network resources, e.g. optical switches, routers, IP networks and BoD-domains.

The OpenNaaS platform is mainly extended with a new type of resource (ROADM), and a new protocol to allow the communication with the remote physical devices. Moreover, its

	<b>D3.3 - Final version of the enhanced OFELIA control framework software</b>	Doc	FIBRE – D3.3-v1.0
		Date	31/03/2014

northbound interface is enhanced to support the control and the management of the optical ports in terms of endpoints and labels.

Table 1 describes the XML-RPC interface.

XML-RPC method	Description
makeXConnection	Create a cross-connection between two endpoints using specified in/out labels
removeXConnection	Destroy a cross-connection
getXConnections	Retrieve all cross-connections identifiers
getXConnection	Retrieve cross-connection details in terms of endpoints and labels
getEndPoints	Retrieve all endpoints of a specified device
getLabels	Retrieve all labels associated to a specified endpoint

Table 1: OpenNaaS extensions for ROADM resources

OpenNaaS provides a CLI for the creation, configuration and startup of its resources. Here, we present a basic workflow to launch OpenNaaS service, to create a new ROADM resource using a proper descriptor file template and to start the configured resource.

Assuming we are into a Linux environment, we can use the following commands to start OpenNaaS:

- `cd <main dir>/platform/target/opennaas-<version>/opennaas-<version>/`
- `./bin/opennaas.sh`

In order to create a ROADM resource, we have to use a resource descriptor file in which we can specify the device's name and type, its supported capabilities and the drivers needed for a correct management of the physical equipment. Then, we can register the protocol used to communicate to the remote device. We can use the following commands to create and configure the ROADM resource:

- `resource:create <main dir>/utils/examples/descriptors/roadm.descriptor`
- `protocols:context roadm:<name> wonesys noauth  
wonesys://<addr>:<port>/?mock=true`


Now, we can simply start the ROADM resource and use its functionalities:

- `resource:start roadm:<name>`

Please, refer to [1] for further details.

#### 4.9.2 ROADM Aggregate Manager

The ROADM AM [12] is basically used to manage, allocate and configure the ROADM resources. It is based on the AMsoil framework [6]: a light-weight pluggable system providing the necessary glue between RPC-handlers and the resource manager. It also supports the

	<b>D3.3 - Final version of the enhanced OFELIA control framework software</b>	Doc	FIBRE – D3.3-v1.0
		Date	31/03/2014

development phase with common tasks such as timers, configuration options, logging system, etc.

The ROADM AM exposes a GENIv3 northbound interface [3] to discover and reserve the resources and to support the “slicing” approach deeply used in the OCF framework. Table 2 describes the GENIv3 interface.

GENIv3 method	Description
getVersion	Query configuration information (e.g. API and RSpec version)
listResources	Retrieve description of available resources. Return an advertisement RSpec
Describe	Describe a single slice or a set of slivers. Return a manifest RSpec
Allocate	Allocate resources as described in the (input) request RSpec
Renew	Request that some slivers be renewed given a new expiration timeout
Status	Get the status of the slivers belonging to a slice
performOperationalAction	Perform start, stop and restart actions on slivers
Delete	Delete slice or slivers and stop them if necessary

Table 2: ROADM AM GENIv3 interface

The resources are described with three types of RSpec (XML-based document):

- **Advertisement RSpec** announces which resources are available
- **Request RSpec** requests the user’s slivers for him/her experiments
- **Manifest RSpec** shows the status of a sliver

Please refer to [4] for a detailed description of RSpecs types and versions.

Figure 6 shows an example of Advertisement RSpec for the ROADM resource:

```
<opennaas:resource>
  <opennaas:name>pedrosa</opennaas:name>
  <opennaas:type>roadm</opennaas:type>
  <opennaas:endpoint>0-17-4</opennaas:endpoint>
  <opennaas:label>648</opennaas:label>
  <opennaas:available>False</opennaas:available>
</opennaas:resource>
```

Figure 6: Advertisement RSpec for ROADM resource

Figure 7 shows an example of Request RSpec for the ROADM resource:



```
<opennaas:resource>
  <opennaas:name>pedrosa</opennaas:name>
  <opennaas:type>roadm</opennaas:type>
  <opennaas:roadm>
    <opennaas:in_endpoint>0-16-4</opennaas:in_endpoint>
    <opennaas:in_label>166</opennaas:in_label>
    <opennaas:out_endpoint>0-1-3</opennaas:out_endpoint>
    <opennaas:out_label>256</opennaas:out_label>
  </opennaas:roadm>
</opennaas:resource>
```

Figure 7: Request RSpec for ROADM resource

#### 4.9.3 ROADM Expedient Plug-in

In short, the ROADM Expedient plug-in [13] is a python code based on the Django framework [14]. It follows the common MVT design pattern allowing the graphical representation of the ROADM resources in terms of endpoints and labels.

The functionalities that the plug-in exposes could be summarized as:

- List the endpoints resources
- List the labels resources
- Reserve a cross-connection
- Describe the resources
- Release the resources associated to a cross-connection

Figure 8 shows a screenshot captured during the booking phase of a cross-connection that involves two ports (specified using ingress/egress endpoint and label) of a ROADM device:





The screenshot shows the OFELIA Expedient web interface. At the top, there are logos for OFELIA and FIBRE. The user is logged in as 'root' in 'NEXTWORKS's island'. The software version is 0.6.1. A navigation menu includes Dashboard, Account, Help, and Logout. A message box displays: 'Successfully created slice roadm-slice.', 'Successfully created project roadm-project.', 'Successfully created/updated aggregate opennaas-am', and '(OpenNaaS) AM exception: [Errno 111] Connection refused'. The breadcrumb trail is 'Home > Project roadm-project > Slice roadm-slice'. The main heading is 'Allocate resources in roadm-slice using opennaas-am'. The form contains the following fields: Name (pedrosa), Type (roadm), Ingress endpoint (0-16-1), Ingress label (822), Egress endpoint (0-17-1), and Egress label (822). There are 'Reserve' and 'Home' buttons at the bottom of the form. A footer contains copyright information: '© 2011-2013 FP7 OFELIA project, FP7 FIBRE project & Stanford University. OFELIA and FIBRE projects are funded by the European Commission (EC) in the 7th framework programme and supported by the FIRE initiative. Development of the OFELIA Control Framework is led by i2CAT foundation. Expedient was adapted from software gracefully provided by Stanford University.'

Figure 8: Reserve a cross-connection using the RODM-AM

## 4.10 OCF Extension to the Optical Layer

OCF has been extended to allow the management of optical devices supporting openflow, and to facilitate experimentation in the optical domain. The goal of this extension is to integrate an Optical FlowVisor that extends original flowVisor with optical device handling. Optical network slicing enabled by a new Optical FlowVisor guarantees isolation between optical network slices taking into account analog features and physical layer impairments (PLIs) of various transport technologies in optical network.

Note: In OFELIA project an optical FlowVisor was developed but it didn't take into consideration the analog features and the physical impairments in the optical layer. In FIBRE we extended the development to include an optical quality of transmission verifier which takes into account the PLIs when creating virtual slice of the heterogeneous optical and packet network resource.

Optical networks are analog in nature, which differentiates optical network slicing from other network slicing technologies, i.e., layer-2 and layer-3 virtualization. Due to the existence of physical layer impairments, the adjacent active optical channels interfere each other, which will impact the isolation of multiple coexisting optical network slices and the way that network slices are composed.

In order take this into account we developed an optical flowVisor which is able to create multiple co-existing and isolated optical network slices. It utilises an "Optical Slice Topology Planner and an advanced "Quality of Transport (QoT) Verifier" to provided virtualization. These two are composed of advanced mathematical modelling techniques and heuristic algorithms which creates optical slices (i.e., allocating wavelength or bandwidth channel, modulation

format and fibre path) such that isolation between them are guaranteed. The Optical FlowVisor creates the optical network slices based on request received for users/applications that includes slice topology, required bandwidth and Quality of Transport.

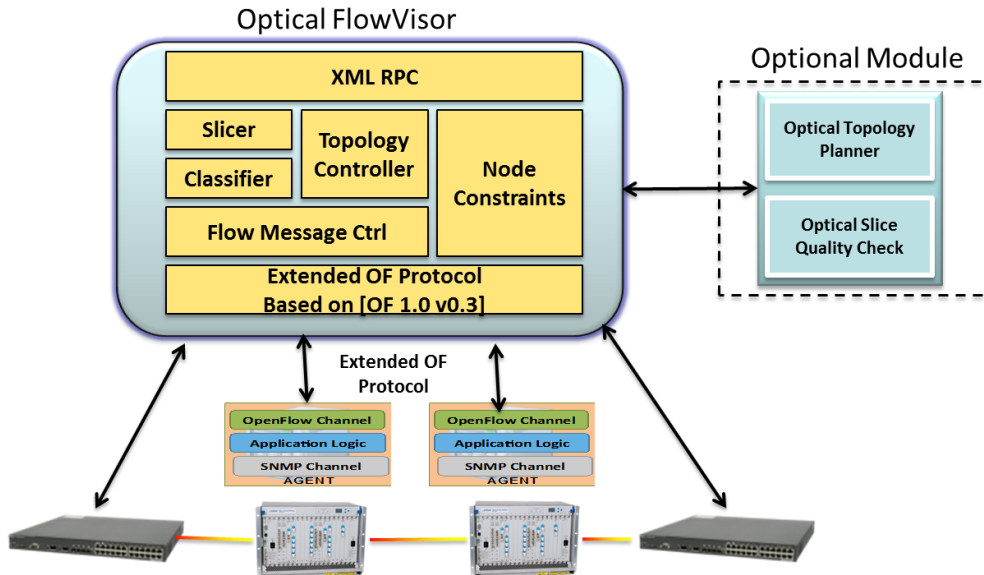


Figure 9: Optical Flowvisor

The components involved in this extension and integration are:

- Optical Flowvisor
- Optical FOAM
- Optical Slice Topology planner and QoT verifier
- Extended Expedient Openflow plugin

Figure 9 shows the interaction between these components.

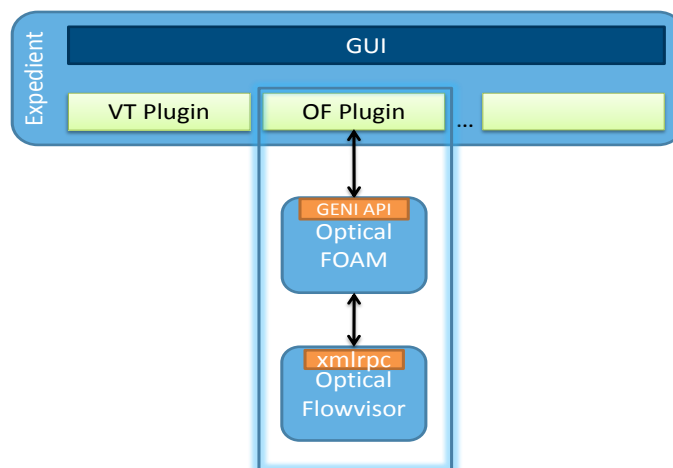



Figure 10: Optical Flowvisor and extended OCF to the optical layer



	<b>D3.3 - Final version of the enhanced OFELIA control framework software</b>	Doc	FIBRE – D3.3-v1.0
		Date	31/03/2014

#### 4.10.1 Optical Flowvisor

Optical Flowvisor can be used for handling optical devices that support openflow in addition to handling packet devices. To achieve this purpose, original Flowvisor functionalities have been extended. Flowvisor is running a server which provides an XMLRPC interface for management. Optical Flowvisor extends this interface to provide the management over optical devices. Extension to the XMLRPC interface are summarized in Table 3:

Method	Description
listCDevices	This method is used to list the optical devices connected to the Flowvisor. It returns the datapath IDs of the optical elements connected to the Flowvisor.
getDeviceInfo <dpid>	This method that exists in the original flowvisor has been extended to return the attributes of optical devices as well. Calling the method remains the same.
getCLinks	This method has been implemented to provide the topology of the optical elements of the network.

Table 4: Extension to the XMLRPC interface

This extended API is accessible to the user by the fvctl tool for testing and debugging.

#### 4.10.2 Optical FOAM

Optical FOAM is the Openflow Aggregate Manager with support of optical devices. The southbound interface is using the extended XMLRPC API that Optical Flowvisor is providing. For communication with the expedient OF plugin, it implements the northbound GENI API.

#### 4.10.3 Construction and Quality verification of optical slices

Due to the interference introduced by PLIs, adjacent active virtual optical networks (VONs) will interfere with each other [1]. So our quality checker VON composition mechanism takes into account the PLIs inherent to the optical layer. The quality of transmission is verified against the effect of PLIs when a VON is successfully composed. For each wavelength channel to be used in a virtual link of a newly composed VON, the quality of all the active wavelengths in the link needs to be verified. If the quality of all the involved VONs is acceptable, the wavelength will be allocated to the new VON, and the quality of all the involved VONs will be updated.

The optical planner and quality checker is external module of the flowvisor that receives slice request information from FOAM and then creates and verifies the optical slice. FOAM uses a XML RPC interface to send the required topology, bandwidth and QoT parameters which is received as an input to the optical planner and checker which then runs the algorithm to get the paths and wavelengths that make up the virtual optical network slice. The algorithm detail is as follows.

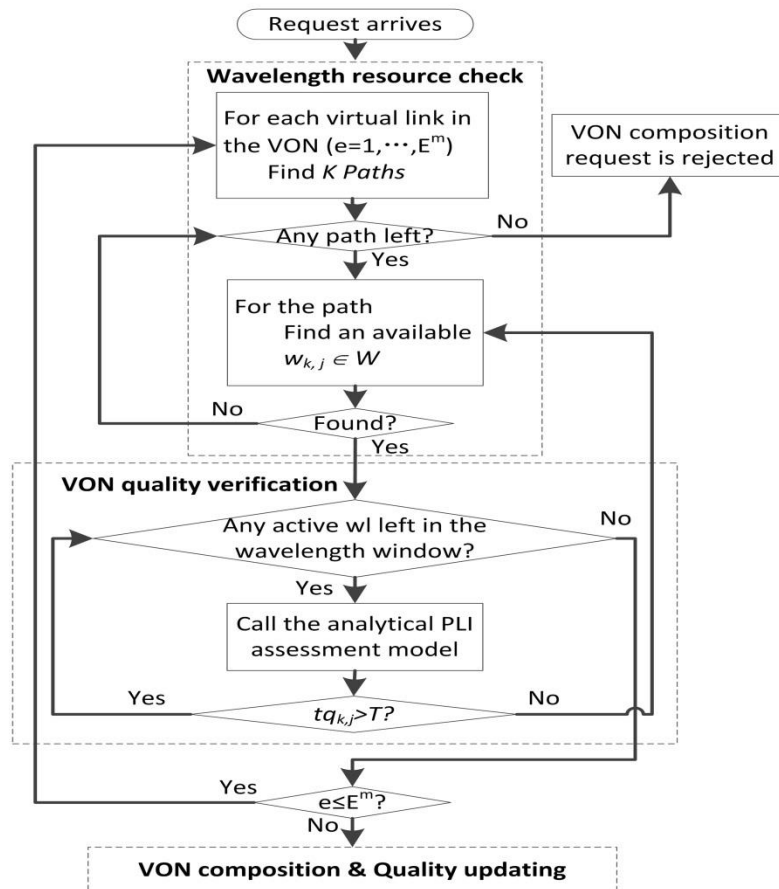


Figure 11: PLI aware virtual optical network composition

Once the VON slice request is received FOAM then pushes the slice information to flowvisor which uses optical openflow extensions to create the slice. The reason for keeping the optical planner and verifier module outside flowvisor is that if in the future flowvisor virtualization is replaced by a new virtualization engine then we can still use the module.

#### 4.10.4 Extended Expedient Openflow plugin

Expedient OF plugin has been extended in order to support calling Optical FOAM API. By calling this API:

- Expedient can display optical network topology.
- User can book optical slices and flowspace (i.e. optical lightpaths).

Slice status	Description	Management
	Optical Slice	Edit slice basic information. Delete slice. <div style="display: flex; justify-content: space-around;"> <span>Update Slice</span> <span>Stop Slice</span> </div>

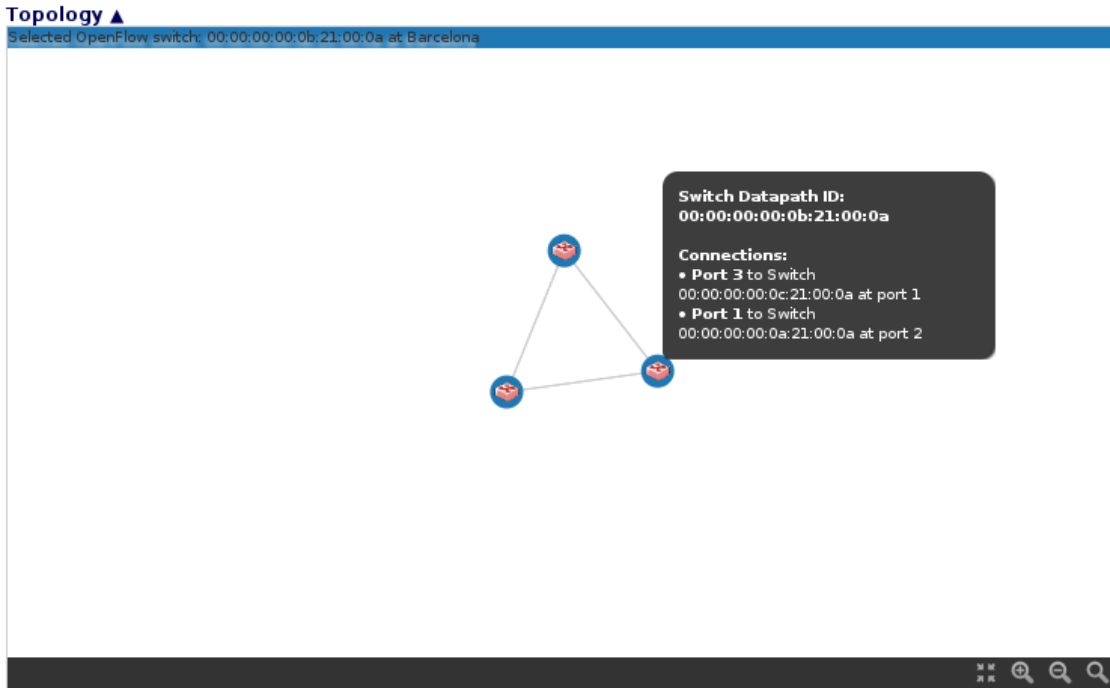


Figure 12: Optical Network Topology display in Expedient

### Set FlowSpace for Slice Optical Slice

1. Select OpenFlow Ports 2. **Select/Modify FlowSpace**

Select wavelength/set of wavelengths to be reserved in the chosen connection.

Frequency	
195.59999	<input type="checkbox"/>
194.59999	<input type="checkbox"/>
194.09999	<input type="checkbox"/>
192.59999	<input type="checkbox"/>
191.8	<input type="checkbox"/>
196.0	<input checked="" type="checkbox"/>
191.7	<input type="checkbox"/>
196.3	<input type="checkbox"/>
196.2	<input type="checkbox"/>

Figure 13: Optical FlowSpace selection (choosing wavelengths)

Extensions to Expedient OF plugin can be summarized as the following:



- Django models and mysql database are extended to support optical network information (e.g. wavelength as a flowspace).
- New web interfaces that deal with optical topology information are added.
- The option where user wishes to book bandwidth instead of selecting wavelength for each optical link is added.
- The option where user can select to have a Path Calculation Element (PCE) with the slice is added.

### Create Slice in Project Project 1

The screenshot shows a web form for creating a slice. The 'Name' field contains 'Optical Slice'. The 'Description' field also contains 'Optical Slice'. The 'PCE' checkbox is checked. The 'Bandwidth' dropdown menu is open, showing '10' as the selected option and '20' as the highlighted option. At the bottom of the form are 'Save' and 'Cancel' buttons.

Figure 14: PCE and Bandwidth selection options when creating a new slice

#### 4.10.5 PCE as a Network Function

The extension of WP5 technology pilot-3 was to provide a PCE as a virtualized network function which can be exploited by a service provider or user for path computation. It extends the NFV principle on to the use case where the PCE acts as virtualized entity which can be shared between many users.

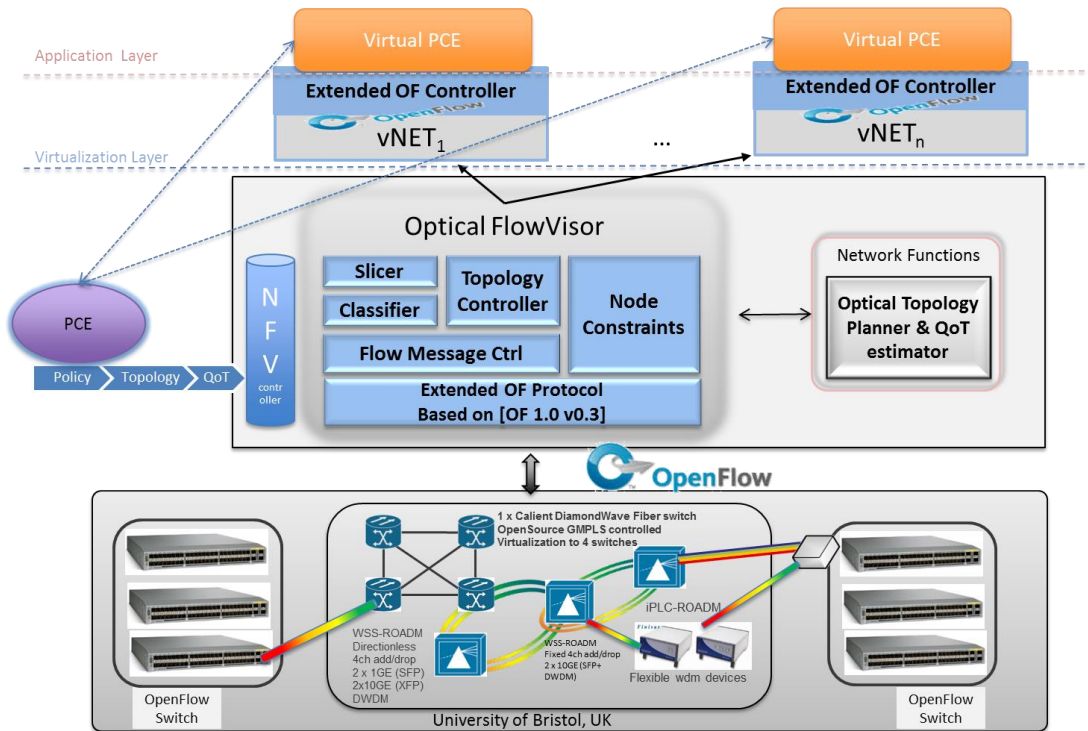


Figure 15 PCE as virtual network function (NFV)

Following the use case #3 users will create multiple optical network slices via optical FlowVisor, which are controlled by different SDN controllers and each of them will run their own instance of the Control Plane with required functionalities (e.g., PCE, load balancing or shortest path routing) and configure the control plane for the specific optical network slice using the PCE. We use a NFV controller to virtualize the PCE and each user will use the PCE API to get paths which are computed only for the virtual slice the user belongs to. The policy control which slices the PCE based on the virtual slices is done via the NFV controller.



## 5 Solved bugs

### 5.1 List\_resources communication error

In previous OCF releases the RSpec handler (VTRSpecInterface) was updated. This update slightly changed the main XML schema used for the intercommunication of the VTAM and the OXA agent. The messages exchanged between these components are based on template XML files. One of these templates is devoted to create asynchronous monitoring messages such as “ListActiveVMs”. The template used was not following the required schema required by VTRSpecInterface after the changes in the schema, causing that when the VTAM received monitoring calls from the Agent, the parser procedure failed due to the wrong schema definition. This error broke the monitoring calls between VTAM, Agent and Expedient and disabled the asynchronous VM update in case of failures.

Due to the asynchronous communication between modules, the parsing error was silent and very difficult to discover. The solution to this bug was to update all the XML templates to the new VTRSpec schema definition.

### 5.2 FlowSpace selection enhancement

Along with other enhancements done in order to simplify the workflow the experimenter goes through when selecting the FlowSpace in the advanced mode, the subset of the network devices and ports previously selected by the user is now automatically checked. Hence, the scenario of accidentally sending some empty FlowSpace request is no more possible.

### 5.3 Definition of the VT AM’s address ranges nor properly handled


One of the features provided by the VT AM is the definition of ranges of addresses (both IP and MAC), which are later used by a server to allow SSH access (IP) and grant a number of identifiable interfaces (MACs) to its hosted virtual machines.

The definition of these ranges was reported to work oddly when an administrator tried to create a range with the same name as one of the existing ranges. To solve this, a proper validation is performed now.

### 5.4 Improved handling of VM allocation and notifications to experimenters

When creating a VM in one of the servers it can happen that the hypervisor’s virtual memory limit is eventually surpassed. Depending on the hypervisor configuration, this may favour either the already existing virtual machines or the new requests. After analysing some incidences, the hypervisor was properly reconfigured to a conservative approach (favouring virtual machines that are already running in the hypervisor). This change was documented in the installation guide [11]. The consequent tests confirmed that this worked as expected as was robust to fit the characteristics of the FIBRE experiments.

As a side improvement, the way to notify the experimenter about any possible error with the VM is improved by showing a descriptive message through Expedient UI.

	<p><b><i>D3.3 - Final version of the enhanced OFELIA control framework software</i></b></p>	<p>Doc FIBRE – D3.3-v1.0</p> <p>Date 31/03/2014</p>
---	---	---

## 6 Future work

Although project requirements have been met; such a big project like this opens the door to a lot of different development ideas. FIBRE has been used as basement for other projects to develop specific functionalities. This section presents some of them that have been started or are in beta version.

### 6.1 Porting OCF to latest Debian OS (Debian 7)

OCF is built under a Debian 6 distribution. However, this distribution is no longer maintained and OCF is growing every day, requiring dependencies that are not updated on Debian 6. For these reasons OCF is being ported to Debian 7.

The porting process is not a trivial task, since OCF must support the new versions of key tools as Django [14] (which requires an important refactor of the core settings of the control framework), Python 2.7 (there are some tools that are only compatible with python 2.6 and at the same time there are errors in official distributions such as the Werkzeug [15] library used to define connections to servers) and Xen (a new version was released that can cause potential errors with the OXA agent), etc.

The update process is being carried on in parallel to support backwards compatibility. This also allows deploying new features on Debian 6 while the porting is being tested.


Another related issue is the way of distributing the OCF. At the moment the source code of the project is being distributed on a GIT repository stored in GitHub [16], and the installation and upgrade procedures are carried out by means of the OFVER tool [17]. Another goal of the porting to Debian 7 is to generate a Debian package with automated scripts for the OCF installation, currently based in OFVER. This way, the work done in OFVER is used to further ease the installation and acceptance of the OCF by providing standard installation mechanisms for all the Debian-based distributions.

### 6.2 LDAP decoupling

OCF uses LDAP to authenticate its users, and it is accessed through most of its modules. However, the access is currently configured separately per each module, and any work that implies modifying the authentication workflow is complicated. In order to improve the access from different entry points, the LDAP access must be decoupled from each module and managed within its own module, a sort of Registry/Clearinghouse that can be easily accessed from any client with the correct certificates and which is more easily extensible than the current model.

It is then useful for federation and standardization purposes to develop a small, independent server that exposes an authentication API (a prototype of ClearingHouse – check [18] for more information on its meaning) to any interested module or island.



	<p><b>D3.3 - Final version of the enhanced OFELIA control framework software</b></p>	<p>Doc FIBRE – D3.3-v1.0</p> <p>Date 31/03/2014</p>
---	--	---

## 7 Conclusions

During the FIBRE-FP7 life-time there has been a huge effort devoted to OCF development as one of the key tools of the facility management and the user experience. This work continued the trace started by OFELIA-FP7 [19] and collaborated with other FIRE projects like Fed4Fire [20] and FELIX [21] enhancing and incrementing the features of the OCF.

This document reported the last important developments in the OCF performed in the WP3 within the T3.3, taking as a source point the MS9 which was not replicated in this document, but which document's reference is a must in order to completely understand the work done in the last year.

The last developments in OCF followed important strategic directions:

- Improve the user experience: the different improvements in the OCF communication and the ability to ask on demand for the status of the computing resources (VMs status notification) allow the user to always know the state of its slices. Most importantly, the automatic procedures added in order to select and grant the OpenFlow resources (through the VLAN manager) translate into a huge improvement in the user experience by getting rid of the waiting time for a slice to be ready for experimentation. Finally, the enhancements to the error handling in the user GUI (Expedient) provide better usability.
- Federation: the addition of the SFA APIs to the different components of the OCF makes it not only possible to integrate in the FIBRE federation framework but also opens the door to any other kind of collaboration. The same can be said for the slow integration of the GENI APIs pointing to the GENI environment.
- Support for new equipment: the ROADM AM and the optical adaptations are a clear example of this, in addition to the generalization of the VMs support in order to be able to manage different VM types of different OS or using different kinds of virtualization.
- Improved robustness: by identifying and solving bugs in the different OCF components we manage to make the framework more stable, thereby enhancing the user experience.

As a final remark, since the FIBRE contributions to OCF has ended, the work performed will be not only used but continued by the project itself and its use cases, but to the rest of the FIRE projects that also have OCF as one of its components.





***D3.3 - Final version of the  
enhanced OFELIA control  
framework software***

Doc FIBRE – D3.3-v1.0

Date 31/03/2014

*"This work makes use of results produced by the FIBRE project, co-funded by the Brazilian Council for Scientific and Technological Development (CNPq) and by the European Commission within its Seventh Framework Programme."*

END OF DOCUMENT

