fibre

FUTURE INTERNET TESTBEDS
EXPERIMENTATION BETWEEN
BRAZIL AND EUROPE

Grant Agreement No.: 288356 (FP7)
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

D5.4: Content-delivery controller application software

Author: WP5
Revised by: Sebastià Sallent, Carlos Bermudo

Due date of the Deliverable: Month 34
Actual submission date: 12/04/2014
Start date of project: June 1st 2011 Duration: 38 months
Version: v.1

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

FP7 Grant Agreement No.	288356
CNPq Grant Agreement No.:	590022/2011-3
Project Name	Future Internet testbeds/experimentation between Brazil and Europe – EU
Document Name	FIBRE-D5.4
Document Title	D5.4: Content-delivery controller application software
Workpackage	WP5
Authors	Mayur Channegowda (UNIVBRIS)
Editor	Mayur Channegowda (UNIVBRIS)
Reviewers	Sebastià Sallent (I2CAT) Carlos Bermudo (I2CAT)
Delivery Date	12/04/2014
Version	V1.0

Abstract

The aim of the WP5 is to develop local and federated technology pilots and validate the deployed FIBRE facilities through showcases, which have been pre-selected from a number of use cases (UCs): Seamless Mobility, High Definition content delivery and Bandwidth on Demand through OpenFlow and GMPLS. The D5.1 and D5.2 deliverables [**¡Error! No se encuentra el origen de la referencia.**, **¡Error! No se encuentra el origen de la referencia.**] present a global and homogeneous view of these UCs, mainly focusing on their goals and objectives, on the involved technologies, on the developed building blocks and on the integration of these modules taken as a whole.

This deliverable starts with a quick report on the Content Delivery pilot use case #2 then elaborates on the modules, software packages, installation and the user steps to execute the use case over the FIBRE control framework.

Note: The pilot uses FOGO 4K solution, developed by LAVID group, as the content delivery software which are available in University of Sao Paulo and University of Bristol.

TABLE OF CONTENTS

List of Figures

1 Acronyms

AM	Aggregate Manager
API	Application Programming Interface
BoD	Bandwidth on Demand
CDN	Content Delivery Network
CDS	Content Delivery Server
CLI	Command Line Interface
CORBA	Common Object Request Broker Architecture
CRUD	Create, Read, Update and Delete
CP	Control Plane
DPID	DataPath IDentifier
DPS	Discovery Packet Switches
E-OFC	Extended-OpenFlow Controller
FIBRE	Future Internet testbeds / experimentation between Brazil and Europe
F-PCE	Flow-aware Path Computation Element
FPGA	Field-Programmable Gate Array
FV	FlowVisor
GMPLS	Generalized Multi-Protocol Label Switching
GUI	Graphical User Interface
HTTP	HyperText Transfer Protocol
IM	Island Manager
LLDP	Link Layer Discovery Protocol
LSP	Label Switched Path
MS	Milestone
NE	Network Element
NMI	Network Management Interface
NOX	OpenFlow Controller
OCF	OFELIA Control Framework
OF	OpenFlow
OFELIA	OpenFlow in Europe: Linking Infrastructure and Applications
OFV	Optical FlowVisor
OMF	cOntrol, Management and Measurement Framework
OML	ORBIT Measurement Library
OSPF-TE	Open Shortest Path First - Traffic Engineering
OSS	Open Source Software
p.	Page
QoS	Quality of Service
ROADM	Reconfigurable Optical Add-Drop Multiplexer
RSVP-TE	Resource Reservation Protocol - Traffic Engineering
SDN	Software Defined Networking
SFA	Slice-based Federation Architecture

SNMP	Simple Network Management Protocol
TCP	Transmission Control Protocol
TED	Traffic Engineering Database
UDP	User Datagram Protocol
UNI	User Network Interface
VLAN	Virtual Local Area Network
VM	Virtual Machine
VON	Virtual Optical Network
VT	Virtualization Technology
WP1	Project Management
WP2	Building and operating the Brazilian facility
WP3	Building and operating the European facility
WP4	Federation of facilities
WP5	Development of technology pilots and showcases
WP6	Dissemination and collaboration

2 Scope

This deliverable presents the FIBRE High Definition content delivery across different sites use case workflow from a user perspective. The document gives a brief high-level overview of the UC, focusing on its objectives and on its architecture. The main part of the deliverable is the software description, in terms of developed modules and technical specifications. Moreover, some details on the package dependencies and on the installation procedure are introduced. Finally ending with the user workflow, describing various steps to be taken by an experimenter to run the use case over the FIBRE control framework.

CNPq Conselho Nacional de Desenvolvimento Científico e Tecnológico 60 ANOS
Ministério da Ciência, Tecnologia e Inovação
GOVERNO FEDERAL BRASIL PAÍS RICO É PAÍS SEM POBREZA

3 Reference Documents

1. [FIBRE] http://www.fibre-ict.eu/
2. [OFELIA] http://www.fp7-ofelia.eu/
3. [NOX] http://www.noxrepo.org/
4. [NOX REPO] https://github.com/noxrepo/nox-classic
5. [NOX WIKI] https://github.com/noxrepo/nox-classic/wiki
6. OpenFlow Switch Specification version 1.3.0 - https://www.opennetworking.org/images/stories/downloads/specification/openflow-spec-v1.3.0.pdf
7. OpenFlow Switch Specification version 1.0.1 - https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-spec-v1.0.1.pdf
8. Extensions to the OpenFlow Protocol in support of Circuit Switching - http://www.openflow.org/wk/images/8/81/OpenFlow_Circuit_Switch_Specification_v0.3.pdf
9. FP6 PHOSPHORUS – Deliverable D2.3 "Grid-GMPLS high-level system design"
10. [IETF RFC 4655] – "A Path Computation Element (PCE)-Based Architecture", Informational RFC, Aug. 2006
11. Milestone MS5.1: Report on technology building blocks
12. M. Channegowda, P. Kostecki, N. Efstathiou, S. Azodolmolky, R. Nejabati, P. Kaczmarek, A. Autenrieth, J.P. Elbers, D. and Simeonidou, "Experimental Evaluation of Extended OpenFlow Deployment for High-Performance Optical Networks," in Proceedings of ECOC, Tu.1.D.2 (2012).
13. Okung-Dike Ntofon, Mayur P. Channegowda, Nikolaos Efstathiou, Mehdi Rashidi Fard, Reza Nejabati, David K. Hunter, and Dimitra Simeonidou, "Experimental demonstration of OpenFlow-enabled media ecosystem architecture for high-end applications over metro and core networks," Opt. Express 21, 4811-4816 (2013)
14. [D5.1]: Report on the detailed design and development of technology pilots
15. [D5.2]: Report on integration of local and federated showcases

3.1 Technology Pilot 2: High definition content delivery across different sites

This use-case takes advantage of the functionalities provided by the OpenFlow Control Plane. In particular, it exploits the flexibility provided by "OpenFlow" and "Flow Routing" to properly distribute the traffic and load balance requests from clients/users of a high-definition video streaming service between delivery sites (CDSs located in different sites).

An OpenFlow-based application (i.e. a POX application) can be interfaced to one or more Content Delivery Servers (CDSs) that form a Content Delivery Network (CDN). The POX application [3] will be able to monitor the CDS performance by retrieving the related status, load and failures. When certain thresholds are exceeded (e.g. the load on CDS or its energy consumption), NOX application will re-route one or more clients to another CDS located in another site. The re-routing will be performed and facilitated by NOX application which will change the flow tables of the OF switches under its control.

In this setup the UK site is responsible for delivering high quality video (e.g., 4K streaming) and the site in Brazil receives a copy of the content (cache).

Assuming that equipment (packet and optical) in both sites are OF-enabled, the European and Brazilian CDSs can be interfaced with a NOX application capable to monitor their performance. Therefore, in a scenario where football matches are streamed during the World CUP 2014, clients in UK (or even Europe) will be served by the UK site. When a certain server usage threshold is exceeded (e.g. the load on the UK-CDS), NOX application will re-route some of the content-delivery traffic to another (less crowded, but not necessarily closer) CDS (e.g. Brazil site). In order to take proper re-routing decisions, NOX applications require to retrieve a detailed description of the network topology and the updated information regarding the performance and failures of the controlled CDS. The considered thresholds and their values will be specific and configurable parameter of the NOX application. So depending on the criteria of interest (load balancing, energy savings, cheaper electricity bills, QoS, failures), NOX application will take the decision and trigger the appropriate actions. It is worth noting that a close interworking of NOX and monitoring applications is also required (e.g. for migrating some users from their current site to another one 'less-loaded').

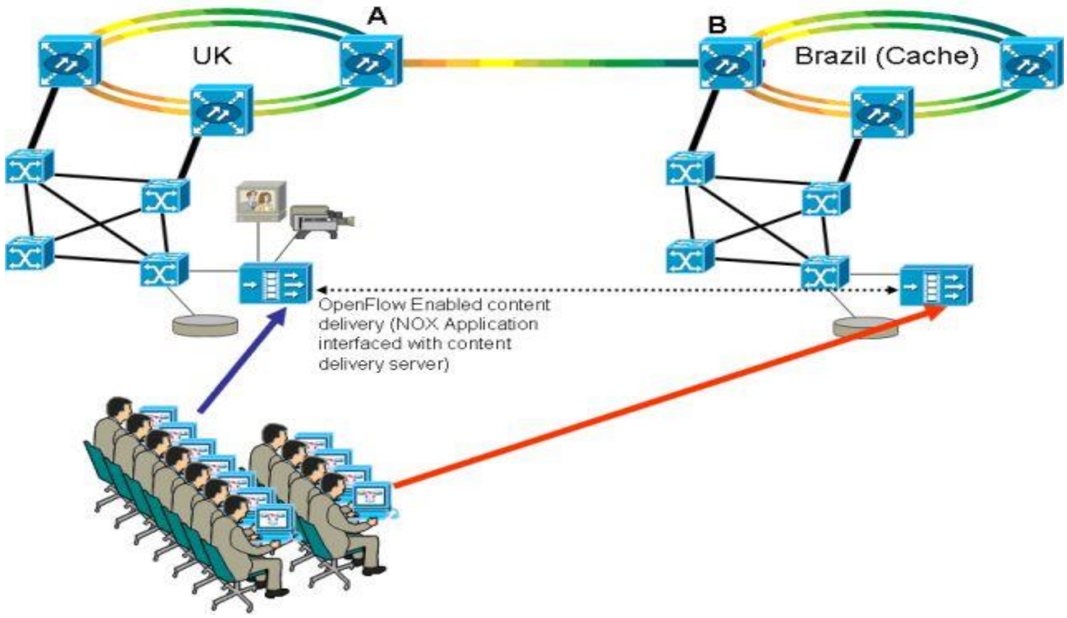


Figure 1: High-definition content delivery

3.1.1 Functional Description

The FIBRE infrastructure provides OpenFlow enabled devices for the use case which provides the topology and current network state information to the POX based SDN controller. A database(db) maintains the topology information along with the node information. Another media broker module on the controller fetches all the information pertaining to the media servers e.g. the capabilities, GPU/CPU information, memory etc. This media related information is combined with the network topology information to provide a media-network aware topology which is maintained and updated in the db. The media broker has another important function i.e it maintains all the information pertaining to the media content (media type, location etc.). So upon request from the user for a particular media content access the media broker will evaluate the media player capabilities of the user and then find the best content (from different repository sources) and the network path to satisfy the request. A flow-PCE will assist the media broker in providing the best available path between the player and the content.

But the focus of the use case is not only to serve new requests but also adapt request to changing network and server conditions. A monitoring module in the POX controller will collate all server & network resource informations, evaluate them if there are any possible problems (congestion, utilization etc.) and then reroute users media players to appropriate media content streamers.

3.1.2 Content Delivery Software Architecture

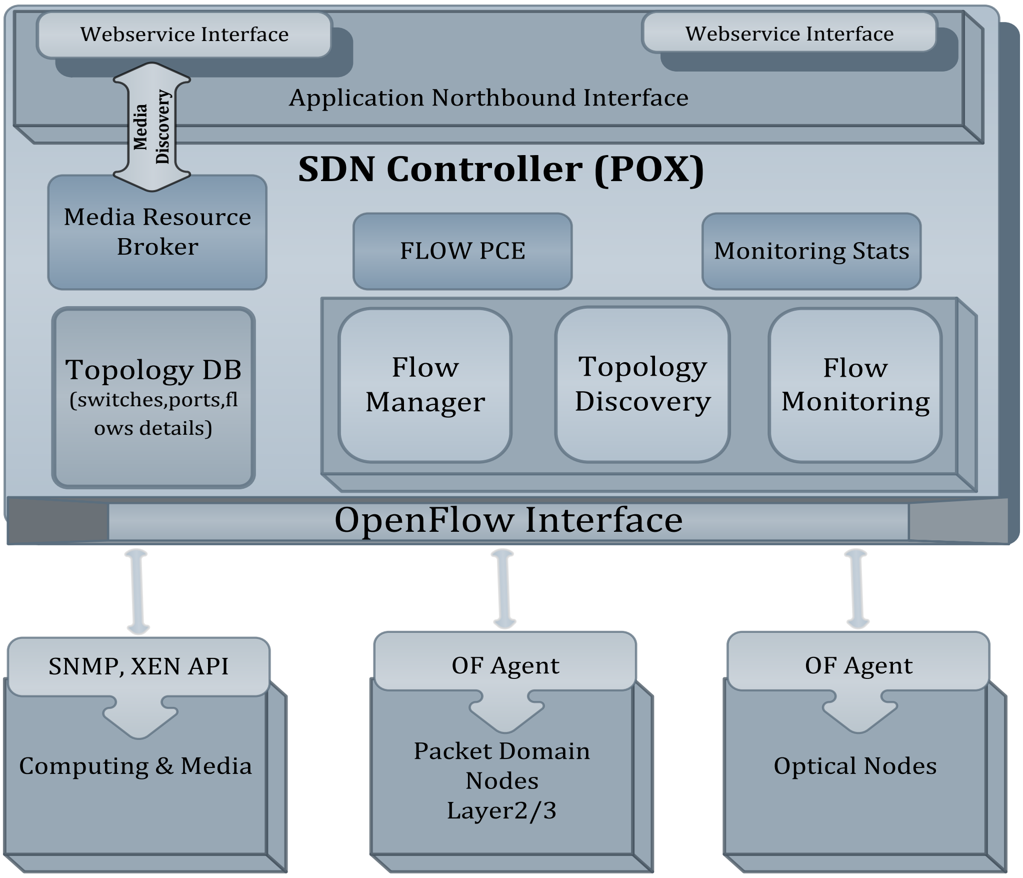


Figure 2: Software Architecture

The content delivery software consists of following main components:

- FIBRE control framework to provide the slice of OpenFlow enabled devices (packet, optical), media server and virtual machines to host software's.
- Media Solution: FOGO 4k Player and Streamer. The FOGO is a proprietary solution and will be available till the end of the FIBRE project.
- POX controller: this is the SDN controller which runs on top of the experimenter slice. It utilizes the OF Interface to control the slice resources. The controller application is hosted over python DJANGO framework. The DJANGO framework consists of application called POX_CW which houses the content delivery software and interfaces. The main modules of the controller are as follows.

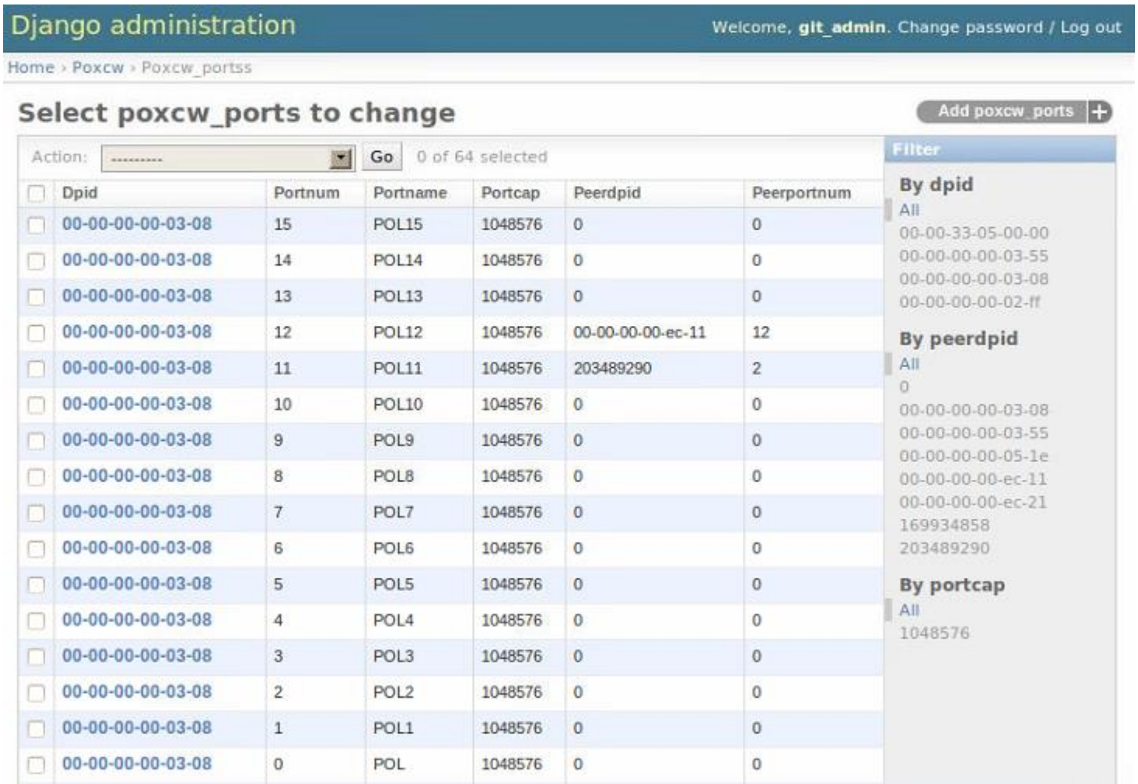
3.1.2.1 Modules description

The POX controller uses following modules for the use case:

Topology Discover: It uses the OpenFlow abstractions to read device information and construct the entire topology. It also supports optical devices discovery which is documented in the libopenflow.py file in the pox controller. The discovered topology is stored in the mysql based topology DB and it is represented via the DJANGO web application which can retrieved using the following link.

http://10.2.9.22:8080/admin/poxcw/poxcw_sw/

Figure 3: Controller showing discovered switches and switching type



Dpid	Portnum	Portname	Portcap	Peerdpid	Peerportnum
00-00-00-00-03-08	15	POL15	1048576	0	0
00-00-00-00-03-08	14	POL14	1048576	0	0
00-00-00-00-03-08	13	POL13	1048576	0	0
00-00-00-00-03-08	12	POL12	1048576	00-00-00-00-ec-11	12
00-00-00-00-03-08	11	POL11	1048576	203489290	2
00-00-00-00-03-08	10	POL10	1048576	0	0
00-00-00-00-03-08	9	POL9	1048576	0	0
00-00-00-00-03-08	8	POL8	1048576	0	0
00-00-00-00-03-08	7	POL7	1048576	0	0
00-00-00-00-03-08	6	POL6	1048576	0	0
00-00-00-00-03-08	5	POL5	1048576	0	0
00-00-00-00-03-08	4	POL4	1048576	0	0
00-00-00-00-03-08	3	POL3	1048576	0	0
00-00-00-00-03-08	2	POL2	1048576	0	0
00-00-00-00-03-08	1	POL1	1048576	0	0
00-00-00-00-03-08	0	POL	1048576	0	0

Figure 4: Discovered topology showing port details of optical switch

Monitoring Stats: this function gets monitoring stats from two places one from the flow stats of OpenFlow and from the SFlow stats.

Flow Manager and Monitoring: these functions assist in pushing flows and then set the intervals to receive the flow stats (packet and byte counts).

Media Broker: Uses the rest API to receive media server and client requests. Has information on the location of the media servers along with their capabilities.

3.1.2.2 Technical specifications

These are the technologies involved on the implementation

Experiment Side:

FIBRE Control Framework

User Side:

POX OpenFlow controller
DJANGO framework
Mysql Database
Python REST API

3.1.2.3 Installation and Configuration instructions for Experiment side

Experimenter is expected to have FIBRE login and he is able to create and start the appropriate slice. Details of creating FIBRE slice is on the FIBRE website.

3.1.2.4 Installation and Configuration instructions for User side

Following are the steps to follow:

Step 1: Install all necessary packages

We use PIP to install packages. Pip is a package management system used to install and manage software packages written in Python

Install Django relates packages:

```
$pip install django=="1.5"

$pip install mysql-python

$apt-get install mysql-server,mysql-client

        Access mysql via: mysql –u root –p

$apt-get install python-mysql

$pip install requests

$pip install django-evolutions

Create Project

$django-admin.py startproject poxcw_django

REST Interface

pip install bottle
```

The project created in DJANGO is called POXCW_django which will hold the POX controller

Install POX SDN controller

```
$ git clone http://github.com/noxrepo/pox

$ cd pox
```

Now move the pox controller into the django project folder i.e inside poxcw_django

Install sFlow-RT for network flow monitoring: sFlow-RT requires Java 1.6+. The following commands download, install and run the software:

```
wget http://www.inmon.com/products/sFlow-RT/sflow-rt.tar.gz
tar -xvzf sflow-rt.tar.gz
cd sflow-rt
./start.sh
```

Install F-PCE software, that is a NXW proprietary code

```
# obtain the code from NXW
git clone <nxw git server address>/gmpls-build
cd gmpls-build
./bootstrap
./configure -prefix=/opt/fpce
./builder -bootstrap fibre
./builder -build
```

Environment set:

Set DJANGO setting using the path to the POXCW application path.

```
Export DJANGO_SETTINGS_MODULE=poxcw_django.settings
```

Configuration and running of the controller:

The POX controller is invoked by pox.py along with optional arguments. POX functionality is provided by components which are specified on the command line following any of the POX options above. Here we describe the components:

Component l2_l1_fullsetup file: Reads all packet and optical switch information. The abstracted topology information is stored in the mysql database which is integral part of the DJANGO web application. This module is run using the following command

```
./pox.py forwarding.l2_l1_fullsetup
```

This module also initializes the sflow module to start monitoring for flows using the trig() function in sflow_triggers.py file

It is also possible to retrieve the POX controller acquired topology using the curl utility:

- curl -X GET http://10.2.9.22:8080/get_topology

Component bottle_test file: The bottle_test component uses the python Bottle package to expose a REST API

```
./pox.py forwarding.l2_l1_fullsetup forwarding.bottle_test
```

FOGO media 4k configuration:

It is assumed that FOGO proprietary solution is already installed.

Before configure the player and start to reproduce a movie, make sure that the PlayerController and the StarterServer has been started.

```
fogo-playercontroller -p 1234 --interface-server-port 1235

fogo-playerstarter -a 127.0.0.1 -p 7010
```

The following steps use tools prefixed with "ui-". They can be found in "fogoutils" directory. We recommend that these steps be executed remotely using ssh.

Next, configure the PlayerSystem:

```
./ui-setps PLAYER_SYSTEM_XML 127.0.0.1
```

In 4-screens setup, you can use the file “fogo-config-ps-2x2wall.xml”. If the movie partitions are misplaced during the reproduction, try to change the DISPLAY environment variable in Environment xml nodes.

The next step is to set the ExecutionParameters:

```
./ui-setpars EXEC_PARS_XML 127.0.0.1
```

In 4-screens setup, use the sample file “fogo-config-pars-4k-sample.xml”. To set the vídeo source, see the Source xml node in each BrickPlayer instance.

Now, you can start the reproduction:

```
./ui-play 127.0.0.1
```

3.2 User Workflow

In the following sections we give further details on the steps that an experimenter can follow to reproduce the use case into the FIBRE infrastructure.

Some assumptions are taken here:

1. the reference control and management framework (CMF) is an enhanced version of OCF developed for the FIBRE project;
2. the experimenter has already received the credentials to access the user portal;
3. the role of the experimenter is "administrator".

3.2.1 Create a slice using the CMF

The experimenter can point the browser to the server address and can access using the provided username and password. Then:

- create a new project giving a name and a short description
- add the aggregate managers to the project. At least we need:
  - a VT Aggregate Manager (computing and media resources)
  - an OpenFlow Aggregate Manager (network resources)
- create a slice giving a name and a short description
- add the aggregate managers to the slice

3.2.2 Configure the test environment

The experimenter can choose the resources and can create the desired topology. E.g.:

- create 1 large VM(1GB RAM) in the managed physical servers
  - VM is used to install the DJANGO based POX controller along with its components
  - There is also a pre-installed server and VM which can be used
- Choose the media server streamer and player.
- Choose the physical and virtual ports of the managed OpenFlow switches
  - User can choose a combination of packet and optical resources.
- Create and grant the flow-space
  - use a VLAN identifier = n (e.g. n=2)
- Access to the hosts VMs and configure the data interface to send VLAN tagged traffic with VLAN identifier = n (e.g. n=2)

3.2.3 Deploy POX controller and the Media player

The experimenter can deploy the controller software following the instructions given in **¡Error! No se encuentra el origen de la referencia.**

After that, the experimenter can start all the components:

- (F-PCE) cd /opt/fpce/bin && sudo ./fibreCtrl start
- (POXCW_Django controller) cd /poxcw_django/pox && $./pox.py forwarding.l2_l1_fullsetup forwarding.bottle_test

3.2.4 Running the Demo:

- ./ui-play 127.0.0.1 will start the player.
    - This internally will request the openflow controller about the best streaming repository to stream the media content to the user. The streaming repository is fixed by the controller and the controller replies with the path and also sets the corresponding network path.
- 2 ways to congest the server or the network to redirect traffic to different streaming servers. Add more traffic by initiating more clients to request content access or use iperf to generate traffic.
- Once the server runs short of memory or CPU cycles the controller will automatically pick the next best streaming server and set the network path.
- Or if the flow size of the media content is categorized as an elephant flow (high bandwidth long duration flow) then another server along with a different path is selected and configured.
- Flow stats retrieval will confirm the network path change.

4 Conclusions

In this deliverable the Content-delivery controller application software in the FIBRE facility use case has been presented.

It is important to note that this use case has been also used to demonstrate the FIBRE infrastructure during two public events:

- 2nd FIBRE open workshop, premises of UPC in Barcelona on 5 November 2013;
- FIA meeting, International Conference Center in Dublin, 2013.

"*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