Extending Neutrality to Experimental Facilities

Jon Matias, Eduardo Jacob, Mariví Higuero, Nerea Toledo
University of the Basque Country (UPV/EHU)
Bilbao, Spain
{jon.matias, eduardo.jacob, marivi.higuero, nerea.toledo}@ehu.es

Abstract—**This paper focuses on the topic of Experimental Facilities, and introduces a novel approach which extends the neutrality concept that comes from Access Networks, to the facilities. The architecture of the solution and the implementation details are also described. The OpenFlow technology is used in order to obtain network slices and virtualize the physical infrastructure. The method proposed to define the slices is the MAC address prefix. The isolation of experiments is mandatory and it is done based on these prefixes by means of a modified version of FlowVisor. A new neutral slice is introduced in the solution. No previous configuration or requirement is needed to get access to this slice. Some basic services are also available at the neutral slice (i.e., captive portal, or an authentication and authorization infrastructure). Finally, the EHU OpenFlow Enabled Facility is presented as a proof of the viability of this approach.**

Keywords-Neutral access; Experimental Facilty; OpenFlow; netowork slice; MAC address prefix

I. Introduction

It is undeniable that the Internet is essential in our daily lives, at a level that was never envisaged. The problem is that the Internet was not designed for such widescale deployment. The explosion of services and the number of connected devices show some of the limitations of the Internet. Although IPv6 was designed to solve some of these limitations, IPv4 is still by far the most widely deployed Internet layer protocol.

Neverthless, the Internet is a valuable source of innovation where new services and companies find the perfect place for developing new ideas. However, these proposals should fit in network scenarios with stringent organizational rules and technical limitations (like the use of IP) to assure world wide connectivity.

Future Internet is an initiative that emerges to solve the current limitations of the Internet as we know it nowadays. There are several efforts in this direction: the GENI [1] (Global Environment for Network Innovations) project funded by the NSF (National Science Foundation) in USA, the AKARI [2] project in Japan, and the FIRE [3] (Future Internet Research and Experimentation) funded by the European Commission. Related to this, the Future Internet Assembly (FIA) is a collaboration effort between projects funded by the EU in the topic of Future Internet.

Regarding the FIRE, the European Commission envisions that promoting the innovation in the field of Future Internet is a safe bet. The idea behind this, is that any novel approach requires early experimentation and testing in large-scale environments. The FIRE initiative addresses this need, creating a multidisciplinary research environment for investigating and experimentally validating innovative ideas for new networking and service paradigms. FIRE proposes that this experimentally-driven research should take place in a large scale and sustainable European Experimental Facility.

Therefore, the Experimental Facility becomes one of the EU objectives, building one big facility available for researchers instead of funding one testbed per project. This also reduces the investment needed by new researchers and European companies to test their novel proposals in a large scale platform. The Experimental Facility provides a realistic deployment to validate revolutionary approaches, even though some of them might only be implemented in a long-term basis.

One essential requirement for the facility is to be as generic as possible in order to be able to support new clean slate approaches. Moreover, it is mandatory to accept several experiments at the same time over the same resources. In this context, the isolation between experiments is fundamental. Therefore, some kind of virtualization of the physical infrastructure is needed.

The problem is that the current technology lacks on fully supporting all these requirements. Therefore, a new proposal is needed at technological level in order to support isolation of experiments, virtualization of physical infrastructure (shared) and enabling clean slate approaches. This last requirement is the most relevant, since legacy switches are able to provide isolation and virtualization by using VLANs. However, this alternative imposes some restrictions, such as broadcast support, learning of MAC addresses, STP or Q-in-Q (if VLANs are demanded at the experiment). We are looking for a more generic approach (i.e., Layer 2 without broadcast domain or learning process).

This paper presents a novel approach for providing neutrality on the access to the Experimental Facility. Another contribution relies on the idea of using OpenFlow technology [4] to implement an facility. This means that the facility is available for everyone, and provides some basic services in order to ease and simplify the access to the platform. By adding neutrality, anyone is able to access the experimental facility and no prior step is needed (i.e., registration or authorization). A set of services and resources, such as a captive portal, are available to ease the access to the experiments. In this context, the EHU OpenFlow Enabled Facility (EHU-OEF) is presented as a proof of the viability of the proposal.

The rest of the paper is organized as follows. In Section II, several Experimental Facilities are described to analyze their evolution and how the neutrality is beneficial in those cases. Then, Section III presents the details of the proposed solution, including the system architecture and the implementation with OpenFlow. Afterwards, Section IV introduces the EHU-OEF platform deployed at the University of the Basque Country. Finally, Section V sums up some final conclusions from this paper and outlines future works.

II. Experimental Facilities

This section focuses on some of the most relevant Experimental Facilities funded by the European Commission which are relevant for this paper providing an overview of the work carried out up to now.

A. Evolution of Experimental Facilities in Europe

FIRE combines research into new paradigms with comprehensive test facilities where the ideas are experimented. This creates a key resource for driving European research into future networks. In fact, FIRE provides a core infrastructure and a playground for future discoveries and innovations, combining research with experimentation.

FIRE is built gradually connecting and federating existing and upcoming testbeds related to the Future Internet topic. Regarding the 7th Framework Project (FP7), there are several calls for projects specifically designed for enlarging the FIRE Experimental Facility. The FP7 ICT Call 2 (2007) represents the first wave of FIRE-specific facility projects, with projects like Federica, Wisebed, Onelab2, PII or VITAL++. The FP7 ICT Call 5 (2010) represents the next wave of FIRE facility projects, with projects like Ofelia, BonFire, Smart Santander, Tefis or Crew. Until now, the last wave of approved FIRE facility projects is the FP7 ICT Call 7, with a specific call for collaboration between Europe and Brazil resulted in a new FIRE project, FIBRE-EU. The FP7 ICT Call 8 has recently closed for evaluation.

The most relevant projects related to the topic of this paper are Federica and Ofelia. The recently started FIBRE-EU project is also relevant but there is little information available..

Federica [5] project is a Europe-wide infrastructure based on computers and network physical resources, both capable of virtualization. The facility can create sets of virtual resources according to users' specification for topology and systems. The user has full control of the resources in the assigned slice which can be used for Future Internet clean slate architectures, security and distributed protocols, routing protocols and applications.

Ofelia [6] project is creating a unique experimental facility that allows researchers to not only experiment on a test network, but also to control the network itself precisely and dynamically. To achieve this aim, the OFELIA facility is based on OpenFlow, a currently emerging networking technology that allows virtualizing and controlling the network environment through secure and standardized interfaces. In a nutshell, OpenFlow enables experimenters to change the behavior of the network as part of the experiment rather than as part of the experiment setup.

Both facilities represent important advancements in this field. However, they still suffer from several drawbacks for deploying experiments that concentrate on the protocol layer.

B. Adding Neutrality to the Experimental Facility

This paper proposes to extend the neutrality concept which comes from Access Networks, in order to be applied in the field of Experimental Facilities.

A Neutral Access Network (NAN) [7] is a special type of Open Access Network which grants positive externality to share infrastructure, by making the access network visible to end users rather than transparent. Therefore, some services are available to users within the access network before they get access to the service edge node.

Current Experimental Facilities can benefit from applying neutrality in the access. Most of the European facilities rely on an out-of–the-band procedure to configure the slices, which is done before the slice becomes available. This limits the dynamic nature of the experiment restricting the points of access to previously configured interfaces. Moreover, it is not possible for an end user to select the target experiment, so, the user is pre-configured to access one of them. Furthermore, there is not any facility service available to support the experiments.

The main challenge of adding neutrality is the technical approach which allows the proper virtualization of resources and isolation of experiments, while providing a basic access with a set of services available for all the end users.

III. Solution Description

This section describes our approach for adding neutrality to the experimental facilities based on [8]. First of all, the reference architecture is presented. The Infrastructure as a Service (IaaS) paradigm is the background in which this model is defined. Resource sharing and virtualization of the network elements are the core ideas behind the IaaS concept. Once the architecture is clearly described, the next subsection presents how this is implemented. We are using the OpenFlow technology in order to build the solution. OpenFlow allows us to split up the control and data planes, and isolate and delegate the control plane of each experiment. Finally, the different types of slices that we have defined in our facility are introduced as an example.

A. Architecture

A basic requirement for any experimental facility is to be as flexible and innocuous as possible. The aim of our Experimental Facility is to provide the required support to do research on networking, enabling the testing of new proposals in a realistic scenario.

The idea of having a common experimental facility to run multiple and diverse experiments introduces the necessity of sharing resources among different experiments. Therefore, we must deal with the idea of virtualization, since the same physical elements (i.e., nodes or links) should be available for multiple experiments.

The virtualization of the network and the delegation of the control plane to another entity (in this case the researcher behind the experiment) are common goals between our objective and the IaaS paradigm. As a result, the Generic Architecture for the Cloud IaaS Provisioning Model [9] is valid as a reference for our architecture. This model is partly a result of the EU project GEYSERS [10], which also deals with infrastructure service virtualization. In this context, it is also worth to mention the work done at 4ward project (FP7) [11]. However, this paper does not follow thoroughly their definitions and concepts, so they are properly clarified and extended when needed throughout the article.

The model consists of three layers, each playing a different role. At the bottom, the Physical Infrastructure Provider (PIP) is placed. This entity provides the physical infrastructure, i.e., it is the owner of the network nodes and links, named physical resources, PR. It is very common to find several PIPs on every experimental facility, for instance in European-wide facilities.

In the middle, the Virtual Infrastructure Provider (VIP) is located. This entity requests physical resources to one or several PIPs and generates a common view. The VIP is also in charge of virtualizing the infrastructure in order to provide these virtual resources (VR) to different experiments.

Finally, the Virtual Infrastructure Operator (VIO) is located on top of the architecture and operates the virtual infrastructure provided by one or several VIPs. The operator rules the behavior of the virtual resources and relies on the VIPs to achieve the essential isolation from other VIOs.

From the perspective of an Experimental Facility, the PIP is the owner of the hardware indispensable to build the physical facility. In the research community, it is very common that all the partners involved in the facility contribute with their own resources, while maintaining the ownership of them.

The VIP is the logical entity (i.e., it is composed of one representative of each entity) which has direct access/connectivity to the physical resources. The role of the VIP is crucial for the proper operation of the facility. On the one hand, the VIP is responsible of the virtualization of the physical resources, enabling that multiple experiments share the same network resources at the same time. On the other hand, the VIP guaranties the isolation among all the experiments, protecting each experiment from the interference of the others. A network slice is a group of virtual and interconnected resources isolated from other experiments. The VIP should assure the agreed QoS and avoid any service degradation which could have impact on their performance.

Finally, the VIO is the researcher that manages the experiment. Since we assume that our researchers do research on networking, they need the platform to impose as less requirements as possible in order to be used to prove novel approaches. In our Experimental Facility, the only requirement is the use of MAC addresses to identify the source and destination nodes. Besides, researchers need to have total control over the network slice provided by the VIP, since they want to test new networking paradigms.

B. *OpenFlow Implementation*

This subsection describes how the previously explained architecture is implemented. First of all, the OpenFlow technology has been selected as the enabler to achieve the implementation of the three layer architecture (Figure 1).

OpenFlow is a flow-oriented technology developed at Stanford University, which splits up the control plane and the forwarding process. By doing so, an external entity, known as the controller, is able to modify (i.e., add or remove) the flow table of one or several switches through a standard interface, which implements the OpenFlow protocol.

Due to this reason, the underlying physical infrastructure of the Experimental Facility must support the OpenFlow technology. Currently, there are a great number of manufacturers that support OpenFlow, such as NEC, HP, Juniper or Brocade. The Open Networking Foundation (ONF) composed by the main actors in the networking market, aims at accelerating the delivery and use of software-defined networks (SDN) by standardizing the OpenFlow protocol. This means that the main requirement that PIPs must fulfill is the OpenFlow support by the physical resources.

As previously explained, the middle layer is the most demanding one, since it must support the virtualization of resources and the isolation of experiments. Regarding OpenFlow, there is a special deployment which enables both requirements: the FlowVisor [12]. The FlowVisor is a special type of controller which acts as a transparent proxy between OpenFlow switches (at PIP) and multiple OpenFlow controllers (VIO). The FlowVisor is the tool that allows the definition of network slices and the delegation of their control plane to the corresponding OpenFlow controller, enabling also the isolation between slices. The ability to define network slices relies on the definition of flows introduced by OpenFlow. A flow is defined as a 10-tuple (ordered list of elements) that consists of different fields, such as switch physical port, MAC address, Ethertype, IP address or TCP/UDP port.

Finally, the operator layer is responsible for ruling the actions of the virtual resources delegated to the slice which is assigned to the experiment. Therefore, the resources and the slice should follow the behavior defined by the experiment. In order to achieve this goal, each experiment has its own OpenFlow Controller managed by the researcher responsible of it. By this approach, the researcher is able to control the

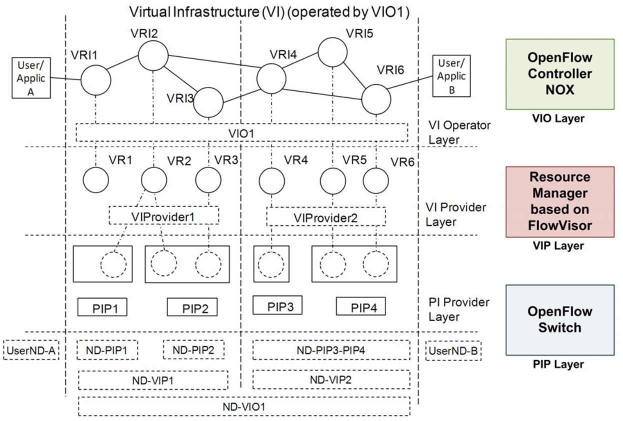

Figure 1. Experimental Facility architecture and OpenFlow implementation

entire slice, with complete isolation from other slices. As previously mentioned, the FlowVisor allows these features.

The isolation of the experiments relies on the isolation of the control plane (done by FlowVisor) of each experiment. When a new packet enters the OpenFlow enabled switch (without a previously defined/installed rule), the packet is forwarded to the FlowVisor. The FlowVisor tries to identify the target controller based on the available information/definition of slices. If there is a match, the packet is forwarded to the corresponding controller associated to the slice/experiment. In fact, all the traffic which matches the slice definition is sent to the experiment's controller. In addition, all the rules and actions allowed to be configured by this controller must be in accordance with conform the definition. Therefore, the infrastructure is shared at the data plane, but only the appropriate controller is able to install the forwarding rules associated to the experiment.

As a summary, the PIPs provide the OpenFlow switches, the VIP relies on the FlowVisor and the VIOs manage the OpenFlow Controllers.

One fundamental decision to take is how network slices are going to be defined. The FlowVisor has the ability to define the slice at different layers (physical, link, network and transport) or as a combination of them. Moreover, each slice can be defined independently of the definition of other slices. However, from the Experimental Facility point of view, it is easier to have a common definition, both for management and traceability.

In our facility, we propose a new method to recognize and distinguish the network slices: the MAC address prefix. This means that the first part of the MAC address univocally identifies the associated slice. This new method only needs one Layer 2 prefix to define and identify the slice, whereas a traditional method needs each and every MAC address to identify the associated slice. Therefore, our new method imposes less administration overhead. For instance, if there are 100 slices with 100 clients per slice, the former method manages 100 prefixes, whereas the latter method needs 10.000 entries to identify the slice associated to each client.

From the management point of view, the setup of a new experiment is very easy: the VIP only needs to find out one free MAC prefix and assign it to the slice associated to the experiment. All the decisions and control actions with regard to the traffic that use that MAC prefix (at the data plane) are redirected to the OpenFlow Controller associated to the corresponding experiment.

From the experiment perspective, all the end nodes involved in the experiment must be configured properly with a MAC address that begins with the delegated prefix. In doing so, there are different options that we have tested: manual or dynamic configuration, and rewriting.

From the user's perspective, the change of the MAC address is not relevant, since the one that they use was assigned by the manufacturer just to assure its uniqueness. From the operator's perspective, the change of their customers' MAC addresses is not a problem, since once they cross a router the MAC changes. Furthermore, the operators will pursue to change the MAC addresses if this allows them to reduce the management effort. It is worth mentioning that trying to identify or authenticate the users by its MAC address is an error. This is because on the one hand, the MACs can be easily spoofed, and on the other hand, they do not identify a user, but just the network card.

The manual configuration depends on the operating system, but there are tools to change the MAC address. For instance, in Linux it can be as easy as changing the IP address. Moreover, when using virtual machines, in some cases, the MAC address can be statically assigned.

We have also developed a service which dynamically configures the MAC address of the end user depending on the experiment that is selected which works on Linux.

The user has the ability to select the target experiment by using the facilities available at the neutral slice. This selection implies that the MAC address to be configured shares the same prefix with the rest of the users of that experiment. Therefore, the MAC is delegated by the experiment slice to the user. This process is similar to the DHCP at the IP layer (nothing in common with SLAAC in IPv6).

The third option, the rewriting, works with independence of the end device, since the MAC address is rewritten at the first OpenFlow switch. The rewriting of the MAC address is a capacity available at OpenFlow specification.

Nowadays, the 1.0 version is the most widely deployed version of OpenFlow, which we are using in our experiments. However, the definition of MAC prefixes is not supported until version 1.1, which is also available at current standard version 1.2. This means that some modifications are needed due to our network slice definition.

In order to implement our solution, the FlowVisor must be modified. The current version of FlowVisor only permits to specify a complete MAC address as a parameter to define the slice, which does not have the needed granularity. So, we decided to adapt the FlowVisor, which is open source and written in Java.

There are two main parts that we have changed: the definition of slices and the matching procedure. On the one hand, the method to describe the slice in order to associate the controller has been upgraded. A new parameter appears: the prefix of the MAC address. On the other hand, the FlowVisor analyzes all the control traffic to identify the associated slice, and corresponding controller to which the OpenFlow packet should be redirected. Therefore, the matching procedure has been also upgraded in order to enable the ability to identify MAC prefixes. With these two upgrades, the FlowVisor is able to work as expected.

C. *Slices*

This subsection presents different types of slices identified at our Experimental Facility. Moreover, the foundations of the neutrality at experimental facilities are explained.

First of all, it is worth mentioning that we have decided to combine both production and experimental traffic over the same infrastructure. This is possible thanks to the OpenFlow technology, which allows us to isolate production traffic from experimentation, and try novel proposals using the same resources.

We have identified three different types of slices: neutral slice, production slice and experiment slice.

The first slice is the neutral slice. As its name suggests, this slice enables the neutrality at the Experimental Facility. This means that anyone is able to get access to this slice. In other words, no prior requirement is needed, no registration process is demanded, and no authentication procedure has to be launched. There are some services that are available at the neutral slice, such as a captive portal with general information about the Experimental Facility, and some other web based services with free access. Therefore, the end user is conscious of the existence of the neutral slice in which several services can be accessed. The neutral slice is also the basic infrastructure to demand access to one of the other slices, either the production slice or one specific experiment slice.

The second slice is the production slice. This slice carries the corporative traffic and gives connectivity to the services provided by the University, both Intranet and Internet access. Since the corporative traffic needs a public IP, a DHCP service is running and IP spoofing is avoided (very easily configured through OpenFlow rules). The production slice relies on a NOX [13] controller with a switch module, that means that all the network elements behave as traditional switches. The access to the production slice is controlled by an authentication and authorization process, which relies on the IEEE 802.1X standard. The production slice is able to identify the EAPoL traffic and redirect all this traffic to the corresponding authenticator (an entity defined by IEEE 802.1X), and then to the RADIUS server. In order to access this production slice, the users must be part of our research group and hence, be subscribed to our LDAP service. Once the AuthN/AuthZ process is successfully completed, the user gets access to the production slice. The first step then is the DHCP service, which is needed to request a valid public IP to get access to Internet.

The last type of slices are the experiment slices. Each experimental slice is requested by a researcher who wants to try a novel proposal, and a new MAC prefix is allocated and assigned to the experiment. As mentioned before, this operation reduces the management overhead, since it is not required to track the MAC addresses of each client. From that moment and on, all the traffic which enters to the Experimental Facility with such prefix is ruled by the Controller owned by the same researcher who asked for the slice. Each experiment has a different behavior and relies on different layers, which can be the commonly known link or network layers or innovative and new layers. This means that each experiment has a unique manner of doing forwarding, and nothing should be assumed. As an example, in one slice a node can be working as a switch, in another one as a router, or in another as something completely new. However, there are some common procedures among all the experiment slices: (1) the user must configure the MAC address (manually, dynamically, or rewriting); (2) an authentication and authorization procedure must be completed before getting access to the experiment slice; (3) from this point and on everything depends on the experiment. The AuthN/AuthZ procedure is similar to the one that takes place at the production slice because both are based on IEEE 802.1X standard. In this case, the authenticator and the RADIUS server can be part of the experiment resources or the experiment can rely on a general authentication service which is provided at the neutral slice. In both options the researcher manages the users who are able to access his experiment slice. In order to manage the users and their credentials, the researcher can rely on the LDAP service (managed by our group) or any other service configured by the researcher.

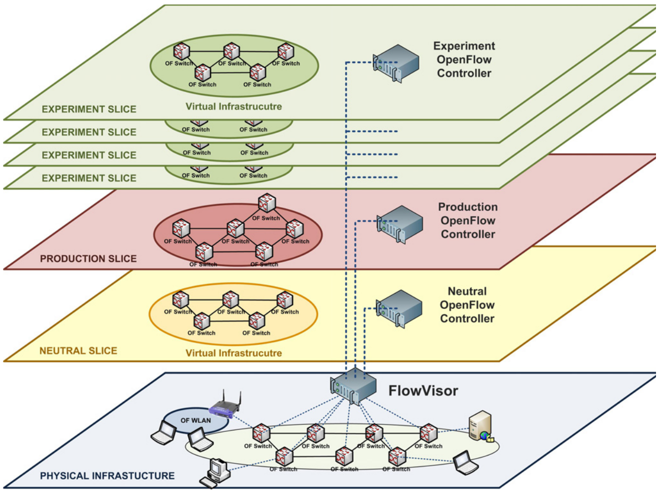

Figure 2. Slices at the Experimental Facility

In conclusion, at this moment, we have one neutral slice, one production slice and as much experiment slices as new good proposals to be tested.

IV. EHU-OEF AS A RESULT

This section introduces the EHU OpenFlow Enabled Facility (EHU-OEF). This facility represents a campus wide infrastructure that supports an early implementation of the previously explained approach. The EHU-OEF validates and demonstrates the feasibility of the proposed solution. This facility also applies the neutrality concept to the Experimental Facilities. This is done by adding a new neutral slice with a set of resources and services (available for anyone), which are helpful for the entire facility.

To the best of our knowledge, the EHU-OEF is one of the biggest OpenFlow infrastructure deployed in Europe for daily operation. Moreover, it is one of the few OpenFlow infrastructures in production, which means that both production and experimental traffic share the same resources.

The EHU-OEF can be categorized also as a Campus Network case, and serve as an example of what can be achieved in both a Campus and an Experimental Facility. Obviously the objectives of both networks are different, but the EHU-OEF presents a wide variety of use cases which can cover both types.

A. Platform description

This subsection describes the EHU-OEF infrastructure and provides a low level detail about the platform.

From the architectural point of view, the three entities have been implemented: PIP, VIP and VIO. Starting from the bottom, there are several PIPs providing infrastructure to the facility. The I2T research group (our group [14]) provides all the OpenFlow enabled switches and some of the fiber and copper links between them. The University of the Basque Country mainly provides the interconnection fiber links at campus level. Euskaltel, a local Telco operator, provides the intercampus fiber links (20 km link at 10 Gbps). The Basque NREN, i2Basque [15], provides a redundant 1 Gbps link for the intercampus connectivity. Finally, the Spanish NREN, RedIRIS [16], provides the 10 Gbps connectivity to the RedIRIS Nova PoP.

On top of the physical infrastructure, the VIP layer is provided by the I2T research group. In particular, a modified version of FlowVisor is managed (and also developed) by the I2T. This enables the virtualization of resources and isolation. In addition, it is based on the approach presented in this paper to define the network slices, the MAC prefixes.

The last layer defines multiple VIOs running on top of the FlowVisor. Last but not least, there is a NOX controller for the neutral slice, another NOX controller with a legacy switch module for the production slice, and several different controllers (most of them are NOX controllers with different specific modules developed by network researchers at I2T) one per each experiment slice.

From the organizational point of view, since the EHU-OEF is located at the University, there are several types of spaces covered by the Experimental Facility. The I2T research laboratory constitutes the core of the facility and the first area deployed. There are also some professor's offices also connected to the platform. Moreover, some of the laboratories used at the telecommunication degree for teaching purposes are also involved in the facility. Finally, there are also Data Centers connected to the EHU-OEF. One of this is owned and maintained by the I2T research group, while another one is run by the UPV/EHU.

From the physical and OpenFlow resources point of view, the EHU-OEF consists of a variety of equipment. There are seven NEC switches (IP8800/S3640 OpenFlow enabled), two NetFPGAS and four WiFi APs (Pantou). The OpenFlow Enabled Facility has a 10 Gbps core, with at least 1 Gbps links to connect the end nodes and redundant paths. Figure 3 shows the current deployment of the facility.

The OpenFlow control traffic is configured as in-band traffic at the NEC switches, which means that the same physical links are used to carry both data and control planes. The VLAN 100 has been defined for this purpose. The in-band configuration has the advantage of not using an extra link to each node (maintaining a parallel network), but the main drawback is that the control plane is affected by the data plane traffic, so congestion, or link failures can happen.

B. *Use cases at I2T*

Several use cases have been identified at the EHU-OEF, which demonstrate the benefits of using this approach. First of all, different profiles appear at the platform: professor, researcher and student. Each type of user has its own permissions and different scope and necessities.

Currently, the profiles are managed by using VLANs, which are statically assigned to physical ports. Nevertheless, it is possible to dynamically configure the VLANs depending on the result of an AuthN/AuhtZ process, but this is not the case at the University. However, by using dynamic configuration, the facility will be restricted by the legacy behavior of the switches, and new proposals will need to deal with procedures like broadcasting, learning or STP. Therefore, it is not possible to fully support the aforementioned use cases.

One of the main advantages of having a procedure to select the target slice and an identity based access control, is that the location is no more a limitation for accessing specific resources. On the other hand, the specific location does not grant indiscriminate access to a set of resources.

If we consider the professor use case, there are multiple alternatives. Professors can demand corporative access not only at their office, but also at the research lab or even in class for some explanation. That is, they demend privileged access to specific resources. They also do research, so they need to get access to the experiment slices not only from the lab, but also from their offices. Finally, the students use a specific slice for setting up a experiment at the lab, after that the professor may need access to that slice to evaluate the work done by the students. The professor has the ability to evaluate the students from its office anytime. Moreover, there can be other students at the lab at the same time.

The second possibility is the researcher use case. The main focus of researchers is doing research. For this use case, we assume that the research lab is their main point of connection. They need corporate connectivity and access to experiment slices. The researchers are the main developers of new modules for the OpenFlow Controllers which rule the different slices. However, due to the distributed nature of the networking experiments, it is expected that they have both types of access at any location throughout the facility. Moreover, the researchers can give lessons as an assistant lecturer or even attend classes as PhD student.

The third option is the student use case. The main point of connection for students is the practice lab. They need both corporative connection to get access to the Internet, and

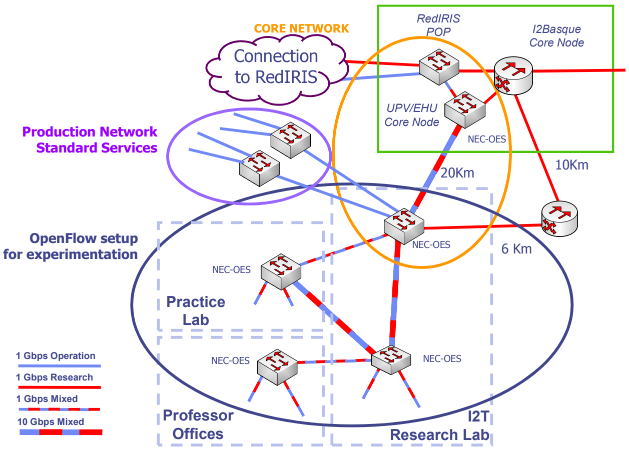

Figure 3. EHU OpenFlow Enabled Facility

access to experiment slices defined by the professors. However, the students can be also be part of research groups, in which they develop and test their own projects (i.e., MSc thesis). Therefore, they also need to get access to some specific experimental slices. Students can develop their own modules for the Controller.

We expect to have performance results that will show the proper operation of the solution and the saving at the operational and administration level.

V. Conclusions

In this paper, the applicability of the neutrality concept, which comes from access network, in the context of Experimental Facilities is analyzed and tested. In addition, we also provide an overview of the efforts conducted hitherto in the field of FIREs and Experimental Facilities.

The main benefit of adding a neutral access to the facility is that the experimental setup and the location of end users can be more flexible and easily evolved over time. Otherwise, the initial setup should be done previously, with some out-of-band tools, as it is done in most of the current facilities. Another important benefit is that the neutral slice provides the platform with a group of basic services that are available for the rest of the slices. Moreover, this neutral slice lowers the barrier of entry to the facility.

The core contribution presented in this paper is the specification of a novel network slice definition. In our approach, the network slices are defined by a unique MAC prefix, which enables a full VLAN space per slice. Each slice defines its own behavior relying on its Controller, can be IP-based, Ethernet-based, MAC prefix-based, with or without support of certain features such as broadcast and so on. Each Controller has total control over the resources assigned to the slice, enabling the path computation and traffic engineering abilities. Regarding the slice isolation, the control plane is isolated by the FlowVisor, and the data plane relies on the flow entry definition and its associated action.

Due to the fact that the facility relies on the MAC prefix to identify the target slice, MAC spoofing could be a problem. However, this problem has a known scope, because the facility only allows traffic with that MAC address on one physical port, as long as it was previously authenticated. It should be noted that this attack only succeeds if it is done at the same physical port and using a MAC address previously authorized. Therefore, the attacker must be at the same port, which it is only possible if the port is shared between multiple end users. Otherwise, the access control is done and the new MAC address should be authorized at the slice. The establishment of the IEEE 802.1AE standard (MACsec) completely solves the problem, since integrity and ciphering support in the access removes the possibility of any spoofing. On the other hand, IP spoofing should be controlled by each slice's controller, whereas the MAC spoofing is controlled by the facility.

The scalability of this approach has two sides. On the one hand, the management is reduced by administering the Layer 2 by using MAC prefixes, instead of the complete list of the client's MAC addresses. On the other hand, the scalability of OpenFlow is a known topic on which people are doing research. The main issue is that the control plane is centralized, with the benefits and also the drawbacks that this implies. A resilience architecture and well dimensioned system could reduce the impact on their scalability.

Currently we are working on adding QoS support to the EHU-OEF. By doing this, we will be able to assure the required network performance for each slice. The main challenge is the lack of QoS support (quite rudimentary) in current versions of OpenFlow. The good news is that it is a known drawback and people are working to solve this issue.

The main proof of the feasibility of this approach is the EHU-OEF infrastructure, which is up and running in a real deployment with production traffic. In the same platform, there are multiple and different experiments which coexist over the same physical resources. We are currently conducting massive tests which are showing the proper behavior and operation of the facility.

Acknowledgment

This work has been partially funded by the Spanish MICINN project A3RAM-NG (TIN2010-21719-C02-01), and the Basque Government Strategic Research Project Future Internet II (IE11-316). It is also supported by the University of the Basque Country's Training and Research Unit for Telecommunications and Electronics (UFI11/16).

References

[1] GENI [http://www.geni.net: April, 2012]

[2] AKARI [http://akari-project.nict.go.jp/eng/index2.htm: April, 2012]

[3] FIRE [http://cordis.europa.eu/fp7/ict/fire: April, 2012]

[4] N. McKeown et al., “Openflow: Enabling innovation in campus networks”. ACM SIGCOMM Computer Communication Review, vol. 38, no. 2, pp. 69-74, 2008

[5] FEDERICA [http://www.fp7-federica.eu: April, 2012]

[6] OFELIA [http://www.fp7-ofelia.eu: April, 2012]

[7] A. Bogliolio, “Introducing Neutral Access Networks”, Next Generation Internet Networks (NGI 09), Aveiro (Portugal), pp. 1-6, 2009

[8] J. Matias, E. Jacob, N. Toledo, and J. Astorga, “Towards Neutrality in Access Networks: A NANDO Deployment with OpenFlow”, 2nd International Conference on Access Networks (ACCESS 2011), Luxembourg, pp. 7-12, 2011

[9] Y. Demchenko et al., “On-demand provisioning of cloud and grid based infrastructure services for collaborative projects and groups”. Collaboration Technologies and Systems (CTS 2011), Philadelphia (USA), pp. 134-142, 2011

[10] GEYSERS [http://www.geysers.eu: April, 2012]

[11] Z, Liang et al., “Virtualization Approach: Evolution and Integration”. Public Deliverable D_3.2.1. 4ward – Architecture and Design for the Future Internet. June 2010.

[12] R. Sherwood et al., “Flowvisor: A network virtualization layer”, Technical Report Openflow-tr-2009-1, Stanford University, October 2009

[13] N. Gude et al., “NOX: Towards an Operating System for Networks”, ACM SIGCOMM Computer Communication Review, vol. 38, no. 3, pp. 105-110, July 2008

[14] I2T research group [http://www.i2t.ehu.es: April, 2012]

[15] i2Basque [http://www.i2basque.es: April, 2012]

[16] RedIRIS [http://www.rediris.es: April, 2012]