Towards Neutrality in Access Networks: A NANDO Deployment with OpenFlow

Jon Matias, Eduardo Jacob, Nerea Toledo, Jasone Astorga
University of the Basque Country (UPV/EHU)
Bilbao, Spain
{jon.matias, eduardo.jacob, nerea.toledo, jasone.astorga}@ehu.es

***Abstract*—A next step in the evolution of Access Networks introduces a scenario in which the fair competition among service providers is enabled through the sharing of access infrastructure. CAPEX savings or regulatory aspects are currently promoting such a scenario. By adding neutrality, the positive feedback loop includes customers, service providers and network operators. The NANDO project implements a new layer 2 approach for Neutral Access Networks. This NAN proposal includes a network operator selection mechanism, a secure instantiation of services and a prefix-based forwarding approach (Ethernet-PF). The OpenFlow technology has been selected for its deployment. OpenFlow is a protocol by which an external entity (controller) can control/modify the flow table of a switch, which rules the forwarding process. This paper is focused on describing the NANDO scenario and the most relevant implementation details related to OpenFlow. In addition, a detailed description of the developed controller and its operational model are shown, including some representative examples. Finally, the functional feasibility of NANDO is validated in a scenario where multiple operators share the same physical infrastructure for service delivery.**

Keywords–Neutral Access Networks; OpenFlow; Access Control; Authentication and Authorization; Carrier Ethernet

I. INTRODUCTION

This paper introduces the NANDO project (Neutral Access Network Demonstrator over OpenFlow), which has been accepted for its deployment over Federica [1]. The experiment was proposed by the I2T Research Group from the University of the Basque Country (UPV/EHU, Spain) in collaboration with the TSLab from the Royal Institute of Technology (KTH, Sweden). A new Federica slice was defined and assigned to this project, once it was accepted by the User Policy Board. The Federica facility provides a virtual infrastructure for researchers who want to test their proposals for Future Internet, such as protocols or applications, in a large-scale scenario.

The NANDO project introduces a pure layer 2 (Data Link Layer) approach [2] to get neutrality in next generation access networks. The main idea behind this proposal is that each customer is able to select the network operator when requesting access to a service. The service is anything (video, voice, data) that demands a specific handling in the access/aggregation network. This means that the service provider is able to impose some requirements to the network operator. Therefore, the delivery of certain services, such as video on demand (VoD) or videoconferencing, requires the support from network operators to ensure a certain QoS.

The Neutral Access Network (NAN) approach includes a network operator selection mechanism, a secure instantiation of services and a prefix-based forwarding proposal (Ethernet-PF). Moreover, network virtualization and resource sharing are fundamental issues to overcome in order to achieve neutrality on the access. This means that the same access network is shared simultaneously by multiple network operators.

The main focus of this paper is the implementation of NANDO proposal by using the OpenFlow technology [3]. Due to its commercial support by major vendors (such as Juniper, HP or NEC), this technology enables the NAN approach to be deployed in real scenarios and production environments. OpenFlow has been developed at Stanford University (Clean Slate) and selected by several projects as the enabler for innovation in future networks, such as E-GENI in USA or Ofelia, Change and Sparc in Europe.

OpenFlow is a flow-oriented technology which splits up the control plane from the forwarding process. In this context, an external entity – the controller – is able to control the switching of packets by defining the forwarding table through a standard interface, the OpenFlow protocol. It was originally conceived as a way of supporting research experiments in production networks, but has evolved into network service architectures such as the NOX [4] and Software Defined Networks (SDN) [5].

The rest of the paper is organized as follows. In Section II, several OAN/NAN proposals are analyzed and related to our approach. Then, the NANDO scenario is described and the main contributions are briefly described in Section III. Afterwards, Section IV describes the platform implementation process and how each of the aforementioned contributions has been developed and deployed with OpenFlow. Finally, Section V sums up some final conclusions from this paper.

II. OAN/NAN PROPOSALS

Open Access Network (OAN) has been proposed [6] to bridge the digital divide and enhance the Internet penetration by enabling the fair competition among service providers on a shared access infrastructure (between users and services). The access network is shared with independent edge nodes for all service providers. Neutral Access Network [7] (NAN) is a special type of OAN which grants positive externality to share infrastructure, by making the access network visible to

 ISBN: 978-1-61208-142-7

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.

In the context of OAN and NAN, there are several technical proposals [8] [9], which try to deal with the requirements imposed by an open or neutral approach. The first proposal [10] is based on DHCP relay, which forwards the DHCP related traffic to the DHCP server associated to the service provider (selected by a captive portal). The same layer 2 network is shared by all the users, which are configured with an IP address from a different subnet depending on the service provider. The second proposal is based on Linux policy routing [9], which makes use of a captive portal for selecting the service provider and source routing for transmitting the packets to the appropriate provider. The third proposal [11] makes use of tunneling to establish a point-to-point connection between the user and the service provider. The available tunnel servers and the service provider associated to each of them should be provided to end users. The forth proposal [8] is only available for WLAN, since it uses multiple SSIDs to distinguish among different service providers. The fifth proposal is based on the CAPWAP protocol (RFC 5415), which allows to assign a different VLAN to the user depending on the 802.11i authentication process. Finally, the sixth proposal [8] is related to the use of IMS in the context of WLAN, which is supported by the 3GPP.

The NAN proposal presented in NANDO is not designed for a wireless scenario. So, the last three proposals are not considered for a further analysis. The first and second proposals make use of a captive portal to select the service provider and then the layer 3 information (e.g., subnet or source IP) is used to distinguish among the target providers. Opposite to these approaches, NANDO is based on layer 2 information, Ethernet services (as described by the Metro Ethernet Forum [12]), to differentiate the target provider. Finally, instead of tunneling all the traffic in point-to-point connections, like in the third proposal, the NANDO approach does not encapsulate the Ethernet frames; thus, reducing the overhead and being more efficient in multipoint scenarios.

Apart from the previous considerations, the main difference is that NANDO enables an environment not only for multiple service providers, but also for multiple network operators providing services over the same access network infrastructure. For this purpose, a new approach for network virtualization is introduced, which is based on the MAC addressing scheme. In this context, the user is able to select the network operator before accessing the services. The selection process is similar to the WLAN association and the use of SSIDs.

If we take a look to the current broadband access architecture (described by the Broadband Forum [13]), the PPPoE is used between the users and the L2TP Network Server (LNS) that is connected to the service providers, and L2TP is used between the L2TP Access Concentrator (LAC) and LNS. The LAC checks the provider, whereas the LNS checks if the user is valid by using a RADIUS server. The architecture supports the introduction of multiple service providers or LNS in the same infrastructure by using multiple point-to-point connections. The main concern is how the services are discovered by users.

On the other hand, the NANDO project introduces a scenario in which no PPP encapsulation is used between users and service providers. The flow definition, introduced by OpenFlow, is used to identify each service and forward traffic from users to the corresponding provider. This approach enables the use of Ethernet frames from end-to-end, reducing the overhead present in tunneling solutions.

Comparing the current broadband proposals and NANDO, there are some key aspects to consider. The former is based on PPP/PVCs for service delivery, which introduces some overhead at data path, whereas the later is based on flows and VLANs. The multicast support is another relevant issue when delivering services, which is not well supported by PPP, since multipoint requires complex mesh networks. On the other hand, multicasting and multipoint capabilities are well supported by Ethernet technology. Furthermore, the authentication capabilities of PPP are quite limited (PAP/CHAP), whereas the IEEE 802.1X introduces an extensible framework for authorization (EAP).

III. THE NANDO PROJECT

The NANDO experiment consists of the deployment of a neutral access network based on OpenFlow switches in which two different network operators are deployed sharing the same physical infrastructure and resources. Each operator provides a set of services to authorized users. At first step, at least one service provider is located in the TSLab at KTH (Sweden), while the customers are located in the I2T Lab at UPV/EHU (Spain). Figure 1 shows the interconnection scenario between both labs, which consists of a not fully meshed group of 5 OpenFlow user space switches (Open vSwitch [14]) along with 2 more machines with the Openflow controllers and the servers (provider selection and AuthN/AuthZ services). At the UPV/EHU premises, NetFPGA [15] devices and IP8800 NEC [16] commercial switches are employed to support experimental traffic in the campus network without disturbing production traffic.

The idea behind this platform is to test he operational issues related to a neutral scenario under real conditions, in which two different network operators (represented by KTH and UPV/EHU) share the same physical infrastructure (from Federica) to deliver the services provided by a third party. Although for implementation feasibility reasons there are

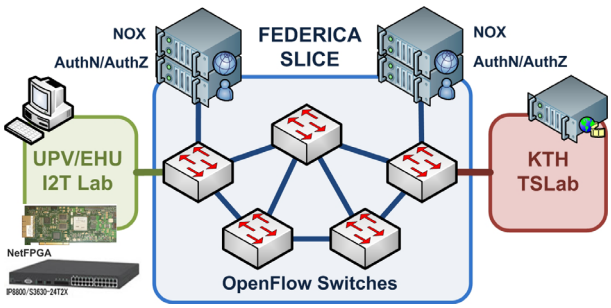

Figure 1. The NANDO Slice

only two entities (KTH and UPV/EHU), the network operator and service provider are supposed to belong to different entities.

Due to space limitation, this section just briefly describes the main contributions of NANDO proposal presented in [2].

A. *Provider Selection Mechanism*

The provider selection mechanism allows customers to freely decide which network operator they want to use before accessing the network and requesting the service. It is a layer 2 service discovery mechanism composed by a scanning phase and a setup phase. First, the customer scans the list of available network operators at its current location. Then, depending on the operator's identity or the services available behind each operator, the customer selects one of them and starts the setup phase. In this step, the customer requests a layer 2 configuration according with the selected network operator. Finally, the end user equipment creates a new virtual interface configured with the leased parameters, such as the MAC address.

B. *Secure Instantiation of Services*

A new procedure for a secure instantiation of services is also implemented as an extension to the Generic AAA Architecture (RFC 2903). The new AAA environment introduces new requirements to deal with. In this context, the service provider must authenticate its customers when requesting the services and impose a set of requirements to the network operator in order to successfully deliver the services. Here, aspects like policy enforcement and obligation handling are addressed.

The basic idea is that after a successful authentication of the customer, the service provider generates a set of profiles which describe the requested service. These profiles are submitted to the network operator to check if all the requirements can be fulfilled. In affirmative case, the operator must establish the path for the service and configure the edge node with the associated access control rules.

C. *Prefix-based Forwarding (Ethernet-PF)*

As a technical solution to overcome the virtualization of the neutral access infrastructure and the distinction (and even isolation) of traffic from one network operator to another, the prefix-based forwarding approach is proposed. By using this approach, the traffic can be easily associated to its operator just by inspecting the prefix (e.g., first byte) of the MAC address. For this, the mechanism described in Section III.A is essential. Once the customer configures the new MAC address leased by one operator, the traffic generated by this interface is handled by the same operator.

If all the MAC addresses leased by a certain edge node have the same prefix (e.g., first 3 bytes), all the traffic to/from users behind this edge node can be identified with a single forwarding rule, thus reducing the flow tables of core nodes. With such a solution, the complete VLAN range is available for each network operator, which eases the network management and inter-operator agreements. The only agreement needed per NAN domain is the very first prefix used to identify each operator.

IV. PLATFORM IMPLEMENTATION

This section is focused on the implementation aspects and OpenFlow related issues concerning NANDO. As previously mentioned, in OpenFlow the flow table is managed by an external entity, the controller. The NOX is the main open-source project which develops an OpenFlow controller, and is used by NANDO. Each and every packet that enters an OpenFlow switch is compared to the flow table, and if no previously defined rule is matched, the packet is encapsulated and transmitted to the controller. Once the packet arrives to the controller, it is analyzed to determine what should be done, and maybe to activate new rules in the flow table in order to handle the packet.

Figure 2 shows a high level view of the platform implementation which represents the main functional blocks and interactions among them. The OpenFlow protocol is used to exchange information between two different entities: the OpenFlow switch and the OpenFlow controller. The OF switch is responsible for forwarding both data and control traffic, which is sent through the downstream interface to/from the users. The upstream interface is used to send the control packets to the appropriate servers (e.g., AuthN/AuthZ or Provider Selection) and the data packets to/from the aggregation network. The NOX controls the behavior of the switch by inserting/removing forwarding rules.

Mainly there are 3 different types of traffic handled by a separate functional block at the NOX. The Provider Selection Traffic Handler receives the selection control packets and inserts the forwarding rules which enable its later exchange between the user and the Provider Selection server. On the other hand, the AAA Process Traffic Handler receives the AAA packets and inserts the forwarding rules which enable the AAA protocol between the user and the AuthN/AuthZ server. This module has also an interface with the AuthZ server to receive the final authorization decisions, which carry the access control rules for inserting/removing new forwarding rules (depending on the AAA process). Finally, the Prefix-based Data Forwarding Decision module

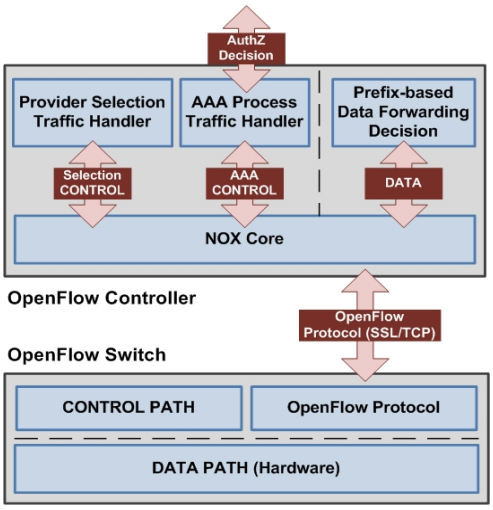

Figure 2. High Level View of the Platform

is the responsible for the data plane. Following the Prefix-based Forwarding approach (Ethernet-PF), the forwarding is based on the first 3 bytes (depending on the prefix length) of the destination MAC address, instead of the whole address. The complete approach relies on the previous controlled distribution of locally administrated MAC addresses provided by the selection mechanism to the end users before accessing the network. Figure 3 introduces a simplified view of the entities and exchanges involved in this process.

A. *Provider Selection Traffic Handler*

As previously introduced in Section III.A, the network operator selection mechanism is based on a client-server protocol. The final outcome is the creation of a new virtual network interface at the client side, with the appropriate configuration for later access to operator. In Linux, this is easily done through the following command:

```
# ip link add link eth0 name veth0 address Leased_MACAddress type macvlan
```

In order to integrate this selection mechanism with OpenFlow, a detailed description of the associated traffic is necessary: a specific ethertype (0x0110) and a multicast address (01:00:5E:90:00:00). First of all, the client sends the scanning request with the aforementioned ethertype to the known multicast address and from its globally administered MAC address (assigned by the NIC's manufacturer). There could be multiple responses to the initial scanning request, with same ethertype, from the server MAC address (unicast) to the client's NIC. After some time (configurable), all the offers from available operators are presented to the client in order to select among them. A unicast setup request/response is sent between the customer's NIC and the selected server.

Regarding OpenFlow, the previously described exchange involves the definition of at least three different flows: the initial multicast flow, both unicast responses (which are exactly the same) and the unicast setup request. The flow registration process is as follows. When a new scanning request comes into the OF switch from the new customer, the packet is redirected to the NOX controller, which checks the ethertype (identified as selection protocol). Then, the NOX enters the learning process and register the physical port associated with the calling station (the NIC address from the customer). At this point, it must be said that all the servers (from network operators) must be previously registered at NOX, which means that their MACs are known.

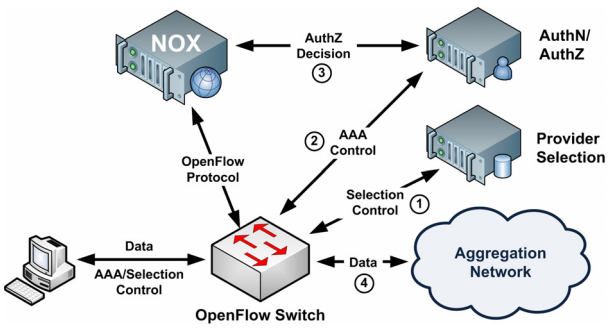

Figure 3. Entities and Exchanges

Due to the temporary nature of the selection process, the rules are charged for a short period of time. In the simplest scenario, the selection process related rules are:

```
DstAddr: 01:00:5E:90:00:00, SrcAddr: NIC_Address, Ethertype: 0x0110, InPort: Learned_Port => Action: Server_Port
DstAddr: NIC_Address, SrcAddr: Server_Address, Ethertype: 0x0110, InPort: Server_Port => Action: Learned_Port
DstAddr: Server_Address, SrcAddr: NIC_Address, Ethertype: 0x0110, InPort: Learned_Port => Action: Server_Port
```

B. *AAA Process Traffic Handler*

This section describes the handling of AAA related traffic with OpenFlow. The secure instantiation of services is based on a modified extension of IEEE 802.1X standard. The standard defines three entities that take part in every AAA process: the supplicant, the authenticator and the authentication server. The traffic relevant to OpenFlow is the exchange of packets between the supplicant and the authenticator, by using the EAPoL protocol. This traffic is completely identified by its ethertype (0x888E) and multicast address (01:80:C2:00:00:03). However, the implemented and deployed solution extends the EAPoL protocol to enable multiple simultaneous AAA processes from the same customer's MAC address. This is essential for requesting multiple services by using the same network operator. Standard and non-standard EAPoL could be distinguished by the version field.

The flow registration process is as follows. Once the client has the new delegated MAC address from the network operator, the authentication process starts on the new virtual interface. Therefore, a new packet with ethertype 0x888E is sent from the new address to the multicast address. Since there is no previously defined rule to handle this traffic, the packet is encapsulated and redirected to the NOX controller. The NOX identifies the ethertype as AAA traffic and checks the multicast address. Then, the learning process takes place and registers the physical port associated with the leased MAC. The authenticator is supposed to be an internal process from the switch, but this is not an option when dealing with an OF switch. Consequently, the authenticator is running on an external machine directly attached to one of the known physical ports of the switch. With this information both rules can be easily defined:

```
DstAddr: 01:80:C2:00:00:03, SrcAddr: Leased_Address, Ethtype: 0x888E, InPort: Learned_Port => Action: Authenticator_Port
DstAddr: Leased_Address, SrcAddr: 01:80:C2:00:00:03, Ethtype: 0x888E, InPort: Authenticator_Port => Action: Learned_Port
```

Up to this point the EAPoL is the only traffic allowed. However, after the AAA process takes place, a predefined traffic from the customer to the provider needs to be enabled.

After a successful authentication (AuthN) process, the authorization (AuthZ) stage takes place. In this step, the AuthZ server determines if the customer is allowed to get access to the requested service and generates a profile describing the service and associated parameters. This profile is needed both for controlling the access and setting up the

 ISBN: 978-1-61208-142-7

connectivity. The former is described in this section, whereas the latter is dealt with in the next Section IV.C.

First of all, a new communication channel with the NOX controller is needed. Opposite to previously described mechanism of sending packets from the OF switch to the NOX, in this case an asynchronous channel from an external entity (the AuthZ server) is required. For that purpose, a web service based on REST is enabled at the NOX controller (Figure 4). Therefore, a new REST client is implemented and launched once the AuthZ generates the profile. For optimization reasons, the profile is transmitted to the NOX controller in JSON format. Once the JSON profile gets into the controller, it is parsed to obtain the needed parameters to activate the new access from the client to the service.

Let us consider a simple example: the client requests a new service to get access to Internet. In this case, the service can be identified at flow level as IP traffic from the client's MAC address to the gateway's MAC address, and vice versa. The rules that should be activated are the following:

DstAddr: Gateway_Address, SrcAddr: Leased_Address, Ethtype: 0x0800, InPort: Learned_Port => Action: Gateway_Port
DstAddr: Leased_Address, SrcAddr: Gateway_Address, Ethtype: 0x0800, InPort: Gateway_Port => Action: Learned_Port

C. *Prefix-based Data Forwarding Decision*

One of the main contributions from this NAN proposal is the prefix-based forwarding mechanism (or Ethernet-PF), built upon the delegation of addresses from network operators to its customers. In this context, only a certain part of the MAC address, defined by a prefix (e.g., first 3 bytes), is enough to determine the physical output port for the packet. That is why in NANDO project, it is only necessary to identify the edge switches for forwarding decisions, which drastically reduces the size of forwarding tables.

Let us consider that the 5 OF switches from the platform are edge switches. This means that only five rules are needed at each OF switch to locate all the possible customers behind those edge switches. Furthermore, each network operator is able to define multiple paths among edge switches by using VLAN tagging. To get the total number of rules those five rules must be multiplied by the number of VLANs. On the other hand, in current switches, the Spanning Tree Protocol is used to get a loop free topology by blocking ports that generate loops. Then, the learning phase takes place to learn all the MAC addresses (one rule per customer) and its associated port. Also, Multiple STP could be enabled to get different topologies associated to different VLAN id's. The learning process is repeated per VLAN. Consequently, Ethernet-PF drastically reduces the forwarding tables and the time, since the MAC's learning process is not further needed.

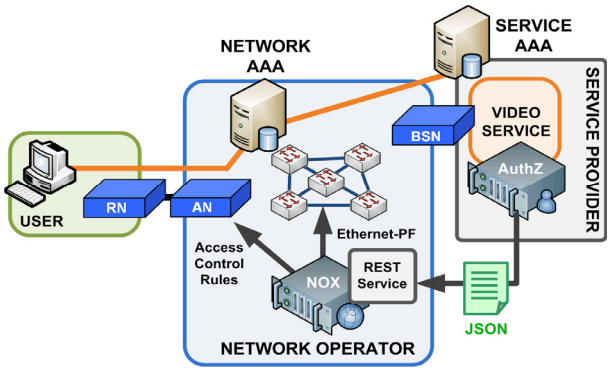

Figure 4. REST Service at NOX (JSON profile)

Regarding the Ethernet-PF implementation, the current OpenFlow version 1.0 does not support MAC-subnetting, but it is to be supported by the next release, version 1.1. So, three options have been considered to get prefix-based forwarding up and running in NANDO. First option is to use the current Ethernet-PF prototype developed with Click tool [17]. Although there is a basic support for OpenFlow in Click (OpenFlowClick) the OFv1.1 is not yet supported. The second option is to hack both NOX controller and OF switch software to add this support. But adding this support to vendor switches (IP8800 NEC) is not possible. The third option is to process all packets by the controller, since at this point the complete packet is available and wildcard matching could be done even at bit level. The problem is the performance of the final platform, since all packets are relayed to the controller. Finally, and mainly due to the mixed infrastructure, with both vendor and software switches, the third alternative is selected.

First of all, a loop free topology is generated with a function developed for NANDO. In this case, the nodes' physical layout is known in advance, so this simplifies the implementation. Then, the controller should register the rules at the OF switches pointing out the defined physical output for each edge switch. Since there is no option for this until version 1.1, the rules are defined and recorded at NOX. A new function has been developed to return the physical output port associated with those records, the idea is to emulate the desired behavior. At this point, each packet that comes into one of those OF switches is relayed to the NOX, and thanks to that function, the output is easily obtained. Then, the packet is forwarded through the predefined port. Of course, the prefix from MAC address (which defines the network operator) and the VLAN id (which defines the topology) are taken into account by this function.

Since network behavior is centralized at NOX, if any failure (link, switch) is detected and reported to the NOX, an alternative topology is computed and reconfigured. There are several proposals to avoid its scalability problems, such as HyperFlow or Maestro.

D. *NANDO validation*

The previously described platform has been validated at functional level. At the beginning, the idea was to validate the proposal over the Federica infrastructure, but due to several unexpected events the original deployment has been very limited. At the end, the main part has been deployed over the I2T Lab (UPV/EHU) and adapted to be extended over a layer 2 platform from the Spanish NREN (RedIRIS).

Figure 5 shows the validation environment that proves the functional viability of NANDO proposal. There are two users (user A and user B), two service providers (service 1 and service 2) and two operators (operator A and operator B)

 ISBN: 978-1-61208-142-7

sharing the same physical infrastructure. Four different setups have been established at the same time. First, user A selects operator A through the provider selection mechanism and creates a new virtual interface with a MAC address delegated from operator A. Then, user A requests a connection to service provider 1 through the AAA process. After a successful authorization the path1 is configured by the NOX and the flow from user A to service 1 is enabled. In a second stage, user A makes use again of the provider selection mechanism to select operator B. At the end, user A has two virtual interfaces, one per operator. By using the new virtual interface, user A launches a new AAA process to request a new connection to service provider 2. Finally, the path2 from the operator B is enabled for this purpose. A similar procedure takes place at user B. At the end, user B has also two virtual interfaces, one from operator A and another one from operator B, which are used to connect to service 2 (path2) and service 1 (path1), respectively.

As previously mentioned, the lack of support for OF v1.1 invalidates the performance evaluation of the proposal, since due to this limitation all the packets must be processed by the NOX, instead of being forwarded at data level.

V. CONCLUSIONS

As the first and main conclusion, the NANDO project has proved the functional viability of the NAN proposal introduced in ACCESS 2010 [2]. Moreover, the OpenFlow technology has been very useful in a mixed scenario such as NANDO, in which vendor switches, NetFPGAs, Open vSwitch and even WiFi devices (OpenWRT) have been integrated under a common controller (NOX), becoming the technological convergence factor. However, the limited set of wildcard options defined in the current OFv1.0 has significantly restricted the fully support for Ethernet-PF. It is not possible until next release (future work) to implement the complete solution over OpenFlow.

Regarding access control, OpenFlow has confirmed to be an ideal technology to unify forwarding and access control rules, since only defined flows are granted, while the rest of the traffic is discarded.

As presented in Section IV.C, the prefix-based forwarding proposal has demonstrated drastic reductions in the number of forwarding rules. For instance, considering a scenario with 20 edge nodes, 35 core nodes and 100 customers behind each edge node, a fully meshed connectivity with Ethernet-PF needs only 20 rules at core nodes. Furthermore, those rules are isolated from customers and the connectivity is enabled just after configuring the rules (since learning process is disabled).

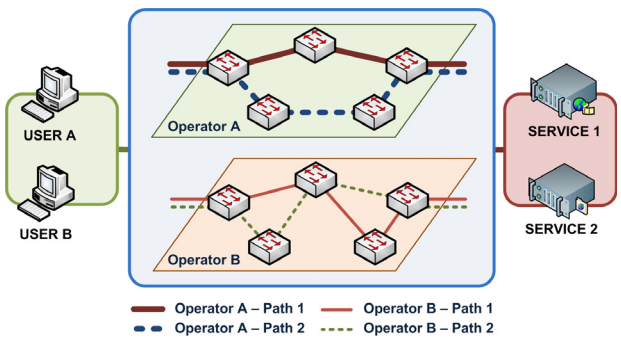

Figure 5. Validation Environment

To conclude, the NANDO project enables a scenario in which customers are able to choose the network operator that they want to use before requesting a service to the provider. Moreover, the same customer is able to select another network operator at the same time to deliver a new service from a different provider. But even more, at the same location (residential or business premises) another customer can select a third network operator to get access to a different service offered by any other provider.

ACKNOWLEDGMENT

This work has been partially funded by the Spanish MICINN project A3RAM-NG (TIN2010-21719-C02-01).

REFERENCES

[1] Federica Project, http://www.fp7-federica.eu, 2010.

[2] J. Matias, E. Jacob, Y. Demchenko, C. de Laat, and L. Gommans, “Extending AAA Operational Model for Profile-based Access Control in Ethernet-based Neutral Access Networks”, The 2nd International Conference on Evolving Internet, pp. 168-173, 2010.

[3] N. McKeown, et. al., “OpenFlow: Enabling Innovation in Campus Networks”, ACM SIGCOMM Computer Communication Review, vol. 38, no. 2, pp. 69-74, April 2008.

[4] 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.

[5] K.-K. Yap, T.-Y. Huang, B. Dodson, M. S. Lam, and N. McKeown, “Towards Software-Friendly Networks,” in Proceedings of the 1st ACM Asia-Pacific Workshop on Systems (APSys ’10), pp. 49-54, 2010.

[6] R. Battiti, R. Lo Cigno, M. Sabel, F. Orava, and B. Pehrson, “Wireless LANs: From WarChalking to Open Acess Networks,” Mobile Networks & Applications, pp. 275-287, 2005.

[7] A. Bogliolo, “Introducing neutral access networks,” International Conference on Next Generation Internet Networks (NGI 2009), pp. 1-6, 2009.

[8] J. Barceló, A. Sfairopoulou, and B. Bellalta, “Wireless open metropolitan area networks,” SIGMOBILE Mob. Comput. Commun. Rev., vol. 12, no. 3, pp. 34–44, 2008.

[9] A. Seraghiti and A. Bogliolo, “Neutral Access Network Implementation Based on Linux Policy Routing,” The 1st International Conf. on Evolving Internet, pp. 158-162, 2009.

[10] A. Escudero, B. Pehrson, E. Pelleta, J. Vatn, and P. Wiatr, “Wireless access in the flyinglinux.NET infrastructure: MobileIPv4 integration in a IEEE 802.11b,” in 11-th IEEE Workshop on Local and Metropolitan Area Networks, pp.51-53, 2001.

[11] A. Bogliolo, “Urbino wireless campus: A wide-area university wireless network to bridge digital divide,” in Proceedings of AccessNets’07, pp. 1–6, 2007.

[12] Metro Ethernet Forum, http://metroethernetforum.org, 2011.

[13] Broadband Forum, http://www.broadband-forum.org, 2011.

[14] Open vSwitch, http://openvswitch.org, 2011.

[15] NetFPGA devices, http://www.netfpga.org, 2011.

[16] NEC IP8800, http://www.nec.co.jp/ip88n/ip8800_s3630, 2011.

[17] Click tool, http://read.cs.ucla.edu/click, 2010.

 ISBN: 978-1-61208-142-7