The EXPRESS SDN Experiment in the OpenLab Large Scale Shared Experimental Facility

Serge Fdida(1), Thanasis Korakis(2), Harris Niavis(2), Stefano Salsano(3), Giuseppe Siracusano(3)
(1) UPMC Sorbonne Universités & CNRS (2) University of Thessaly (3) CNIT/University of Rome Tor Vergata

Abstract— **In this paper we describe the design and implementation of an experiments dealing with SDN for Wireless Mesh Networks over the OpenLab Facility. The experiment is called EXPRESS: "EXPerimenting and Researching Evolutions of Software-defined networking over federated test-bedS". EXPRESS aims at designing and evaluating a resilient SDN system able to operate in fragmented and intermittently connected networks as needed in a Wireless Mesh Networking environment. The experimental dimension of EXPRESS is to deploy the designed SDN infrastructure over a federation of three testbeds (PlanetLab, NITOS and w-iLab.t) from the OpenLab federation. The experiments consist in the evaluation of a designed solution for the selection of the SDN controller by the Wireless Mesh Routers in intermittently connected networks. The experiment is executed through the OMF framework (cOntrol and Management Framework). OMF provides the ability to describe the distributed experiment spanning over different physical testbeds. Following the experiment description, the OMF framework realizes the configuration of the resources (in our case the Wireless Mesh Routers) and their interconnection, runs the experiment and collects the results.**

Keywords—Software Defined Networking, Open Testbeds, Distributed Tesbeds, Testbed Federation, Wireless Mesh Networks

I. INTRODUCTION

SDN is becoming a preferred networking paradigm for Enterprise Networks and Data centers. Since, the networking community is pushing the envelope of SDN to use it in many other type of environments. The expected benefit is mostly related to the possibility to perform dynamic traffic engineering.

This paper explores the possibility to use SDN in a dynamic heterogeneous environment, such as fragmented and intermittently connected networks. The solution should be able to easily organize together isolated networks, as it may be needed in a dynamic Wireless Mesh Networking (WMN) scenario. We designed EXPRESS, which integrates the basic solutions necessary to discover the network topology and operate the routing protocols in WMNs with an SDN architecture meant to support advanced services (e.g. dynamic traffic engineering).

The complexity of the environment under study makes the evaluation of EXPRESS a complicated task. The OpenLab facility has been used for that purpose. Indeed, the experimental dimension consists in deploying the proposed SDN infrastructure (and the implemented software modules) over a federation of three testbeds (PlanetLab Europe, NITOS and w-iLab.t) from the OpenLab federation and collect performance measurements. OpenLab exploits the concept of federation of testbeds that allows a simplified access to diverse set of heterogeneous resources to the experimenter. The paper briefly presents the basic components and benefits of using OpenLab. It describes the experiments that were carried out and a set of results that demonstrate the ability of OpenLab to provide insight into the designed solution.

II. THE OPENLAB FEDERATED TESTBED

OpenLab comes from a vision originated in 2005, built on several issues related to experimentally driven research. The networking community was facing a few successes in its ability to build testing tools (like PlanetLab or Emulab) but many more failures due to well-identified causes. In addition, a challenge that is still open to our community is to develop reproducible research, meaning that one should be able to reproduce the results that are published and supports a discovery.

This vision considers that an experimenter, namely, the one that uses the facility, should have access to an ecosystem or a "market" of various resources managed by different authorities. For this purpose, the experimenter will register to one such authority that will act as a mediator towards its peers.

The beauty of this model is grounded on the observation that there exist plenty of valuable resources out there that one can benefit if an open access is provided. Some of these resources might be unique, or the sum, or combination of them might be valuable. In addition, it became quickly evident there is not a single testbed that fit all needs and that, solely, a federated model will succeed to embrace the vision.

Enabling this vision requires to define an architecture that supports the underlying concept of federation that was originally introduced in the OneLab EU funded project in 2006. Therefore, it became instrumental to address the following questions:
- What is the right level of abstraction, the minimum set of functionalities to be adopted to share resources owned by various authorities?
- What is the governance model that best supports subsidiarity?

A. The Architecture for enabling an Internet of Testbeds

We benefited from the experience in architecting the Internet to design our model. It is grounded on two principles:
- The "Hourglass" model of the Internet that identifies the IP protocol as the convergence layer. We'll define one such convergence layer for the Federation of testbed resources,
- The peering model of the Internet that relies Customer sand Providers and define peering agreements in a way that there is not a single point of control. Here, we will clearly identify Experimenters, Testbed owners or providers and the Facility itself that rule them all.

We therefore have defined the following abstractions:
- Resource: Testbed ensures proper management of nodes, links, switches, ...
- User/Experimenter: Testbed guarantees the identity of its users
- Slice: A distributed container in which resources are shared (sharing with VMs, in time, frequency, within flowspace, etc.). It is also the base for accountability.
- Authority: An entity responsible for a subset of services (resources, users, slices, etc.)

SFA (Slice Facility Architecture) was designed as an international effort, originated by the NSF GENI framework, to provide a secure common API with the minimum possible functionality to enable a global testbed federation.

The fundamental components for testbed federation were built incrementally, as the understanding about the requirements matured. The first international realization of federation arose in 2007, as a mutual investment from PlanetLab Central, managed by Princeton, and PlanetLab Europe, established by UPMC and INRIA in Europe. It then became of utmost importance to enlarge and extend the federation principle to other type of resources, a more scalable model of federation and an increased ease of use. In parallel, started the important effort to complement and populate the architecture with components mandatory for the entire experiment life cycle.

The experiment lifecycle comprises the following steps:
❶ User account & slice creation
❷ Authentication
❸ Resource discovery
❹ Resource reservation & scheduling
❺ Configuration/instrumentation
❻ Execution
❼ Repatriation of results
❽ Resource release

Step ❶ is handled by the Home Authority of the User, the one the user has registered with. Steps ❷ to ❹ and ❽ concern all involved authorities. Steps ❺ to ❻ are not in SFA but other components such as OMF have been developed for this purpose. OMF is a control, measurement and management framework that was originally developed for the ORBIT wireless testbed at "Winlab, Rutgers University". Since 2008, OMF has been extended and maintained by NICTA as an international effort.

SFA provides a secure API that allows authenticated and authorized users to browse all the available resources and allocate those required to perform a specific experiment, according to the agreed federation policies. Therefore, SFA is used to federate the heterogeneous resources belonging to different administrative domains (authorities) to be federated. This will allow experimenters registered with these authorities to combine all available resources of these testbeds and run advanced networking experiments, involving wired and wireless technologies. The SFA layer is composed of the SFA Registry, the SFA AMs and drivers. The SFA Registry is responsible to store the users and their slices with the corresponding credentials.

MySlice[1] was introduced by UPMC as a mean to provide a graphical user interface that allows users to authenticate, browse all the testbeds resources, and manage their slices. This work was important to provide a unified and simplified view of many hidden components to the experimenter. The basic configuration of MySlice consists on the creation of an admin user and a user to whom all MySlice users could delegate their credentials for accessing the testbed resources. In order to enable MySlice to interact with heterogeneous testbeds, MySlice has to be able to generate and parse different types of RSpecs (Resource Description of the testbeds); this task is performed by plugins.

B. The OpenLab facility

The OpenLab[2] federation of testbeds was launched in august 2014 under the brand of OneLab Facility in order to avoid confusing the Openlab EU funded project that ended in august 2014 with the Facility that we expect to be sustainable). For the sake of clarity, we continue to use OpenLab as the name of the facility in this paper. OpenLab started with the following set of initial federated testbeds:

- Internet-overlaid testbeds: The public fixed-line Internet, at a global scale.
PlanetLab Europe[3], a platform offering virtual machines on over 300 servers located at over 150 locations across Europe.

- Wireless, sensing, and mobility testbeds: Internet of things testing environments.
These platforms offer both fixed nodes and mobile nodes with controlled mobility via robots or model trains. The first testbeds to fit this category are FIT-IoTLab[4] (a French testbed

[1] http://www.myslice.info
[2] http://new.OneLab.eu
[3] http://www.planet-lab.eu
[4] https://www.iot-lab.info

funded by ANR) and the NITOS[5] testbed from the University of Thessaly. The w-iLab.t[6] testbed from iMinds was added for the purpose of the EXPRESS experiment.

As the need for networking research evolves, new testbeds appears or new requirements were expressed. This has been the case for instance for the OpenFlow/SDN developments that trigger new needs and emerging testbeds (such as OFELIA[7] in Europe). The OpenLab project quite early developed a solution named OpenFlow in a Slice that provides the ability to run Openflow vswitches in a slice of a PlanetLab Europe set-up. Experimenters were then able to create an OpenFlow overlay[8] network by specifying the links between PLE nodes, benefiting from the large number of PLE nodes deployed.

Finally, the OpenLab Portal[9] was developed and provides the generic access to the facility. The portal is implemented by the MySlice software component, which allows the users to manage their slices through SFA. The NOC (Network Operation Center) has been installed in the premises of UPMC and allows a full access to the federated testbed, users and experiments managed by the facility. OpenLab is freely accessible to the community at large.

III. THE EXPRESS EXPERIMENT: SDN FOR WIRELESS MESH NETWORKS

The EXPRESS experiment has been selected for funding in the 2nd OpenLab competitive call for additional project partners. EXPRESS stands for "EXPerimenting and Researching Evolutions of Software-defined networking over federated test-bedS" and it includes two main dimensions: scientific and experimental. The scientific dimension considered the design of an innovative, resilient SDN system able to keep operating in fragmented and intermittently connected networks. Such a system should be able to easily glue together isolated networks, as it may be needed in a dynamic Wireless Mesh Networking (WMN) scenario. EXPRESS integrates the basic solutions necessary to discover the network topology and operate the routing protocols in WMNs with an SDN architecture meant to support advanced services (e.g. dynamic traffic engineering). The experimental dimension consists in deploying the proposed SDN infrastructure (and the implemented software modules) over a federation of three testbeds (PlanetLab, NITOS and w-iLab.t) from the OpenLab federation and collect performance measurements.

A. *Scientific questions and technicall challenges*

The main scientific question behind the experiment is whether the SDN paradigm can be applied to networking scenarios where: 1) it is not feasible or reasonable to implement a separate *out-of-band* signaling infrastructure among nodes, therefore SDN signaling will be intermixed at packet level with user data flows following an *in-band* approach; 2) there is a relatively high probability of link failure, the network can become partitioned in disconnected set of nodes, the partitions can later merge back into larger partitions. These conditions may occur in Wireless Mesh Networks (WMNs), like Community Networks ([14]), in which some parts of the network are interconnected by long links that may temporary fail. The reference scenario for our work is shown in Fig. 1, as an example the link between the Wireless Mesh Routers A and B can partition the network in two parts if it goes down. Let us now consider the advantages and the criticalities of using SDN in WMNs.

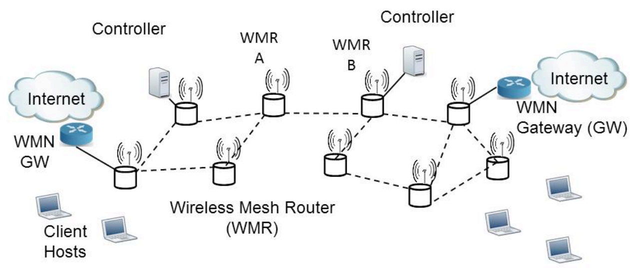

Fig. 1. Wireless Mesh Network reference scenario

The advantage of introducing the SDN paradigm in such environment are mostly related to the possibility to perform dynamic traffic engineering to optimally distribute the traffic over the wireless resources and across the different gateways towards the Internet that could be available in the WMN. The IP best effort routing based on distributed shortest path (e.g. with OLSR [15] or OSPF routing protocols) may lead to poor utilization of the available capacity, with bottlenecks constituted by congested wireless links or gateway nodes. We expect that, using the SDN paradigm will make possible to optimally allocate the user traffic with the needed level of granularity.

On the other hand, using SDN in the considered WMN scenarios has some criticalities. We have identified two main challenges. As for the first challenge, a SDN based approach requires a control connection between the controlled network nodes and the SDN controllers. In a fixed networking environment the *control-plane* communications between the switching nodes and the SDN controllers typically run over *out-of-band* channels, separated from the *data-plane* traffic. For example VLANs can be used in a layer 2 Ethernet network to establish a "signaling" network that will operate independently from the SDN mechanisms used to manipulate the data-plane traffic. Replicating this approach in the WMN scenario will not work, because: i) VLANs are not typically used in WiFi networks; ii) the basic connectivity among nodes of a WMN (referred to as WMR, Wireless Mesh Routers) is established using layer 3 routing protocols. The first challenge

[5] http://nitlab.inf.uth.gr/NITlab/
[6] http://ilabt.iminds.be/wilabt
[7] http://www.fp7-ofelia.eu/
[8] https://www.planet-lab.eu/doc/guides/user/practices/openflow
[9] http://portal.OneLab.eu/

is therefore to design a SDN solution suited to the characteristics of WMNs.

The second challenge that we addressed concerns the applicability of SDN in network partitioning and merging scenarios. Assuming that a SDN controller runs over a set of WMRs, if the network becomes partitioned a subset of WMRs will disconnect from the controller and will need to associate to a different controller (if available in the partition). On the other hand, if two network partitions under the control of two different SDN controllers merge into a single partition, it is desirable that all WMRs fall under a single SDN controller. Clearly, the service logic in the different SDN controllers needs to be coordinated, but as prerequisite we focused on the issue of the establishment of the connection between the WMRs and the most appropriate SDN controller. We can restate the second challenge as "SDN controller selection under network partitioning and merging scenarios". The problem of assigning a SDN controller to each switch in a network with different SDN controllers has been already faced when considering "distributed" SDN solution with multiple controllers, see for example [2]. According to the OpenFlow specifications [4], when a switch is connected to multiple SDN controllers, one of these controllers can act as *master*. The procedure to select the master controller for a given switch is typically referred to as master *election*. The reason is that the procedure is distributed among the controllers that coordinate with each other in order to *elect* the master. The switch is slave in this approach and will be notified by the winner of the election. This procedure works well assuming that there is a stable connectivity among the controllers (in fact in the typical use case the procedure needs to elect a master controller among a set of "replica" controllers operating in the same data center). Using this procedure in the considered WMN scenario may easily lead to inconsistent results. The convergence time of routing protocols used in WMNs is in the order of seconds. During transient phases the different controllers may have different visions of network connectivity. For example two controllers could both believe to be the best candidates to take mastership of a given switch and can both start acting as master for the switch.

A. Solutions to 1st challenge (SDN in WMNs)

In order to address the first challenge identified above, the designed solution foresees to use the OLSR routing protocol [15] to establish the basic IP connectivity in the WMN. Coexistence mechanisms are defined between packets routed using classical IP routing tables (including the OLSR packets) and packets routed using the SDN approach under the instructions of SDN controllers. The forwarding of SDN signaling packets follows an *in-band* approach, i.e. the packets between the switching nodes and the SDN controllers are sent on the same network on which the data plane packets are sent. The signaling packets belonging to the SDN control plane (among WMN nodes and SDN controllers) are forwarded using the basic IP routing information established using OLSR, while the data packets can be forwarded using the basic IP routing or using arbitrary routing under the control of the SDN controller.

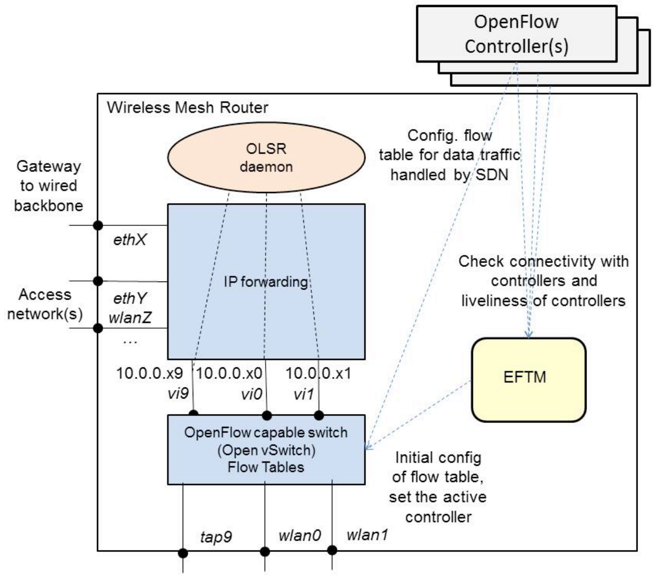

Fig. 2. wmSDN node architecture

The architecture of a node implementing the proposed solution is reported in Fig. 2, the solution was first proposed in [3], referred to as wmSDN (wireless mesh SDN). Then it has been improved and extended in [5] taking into account the OSHI IP/SDN framework [6]. We refer the reader to [5] for the technical details of the solution. In the context of EXPRESS, we have ported and deployed the solution on real wireless nodes in the NITOS and w-iLab.t testbeds.

B. Solution to 2nd challenge (SDN controller selection in network partitioning and merging scenarios)

Coming to the second challenge, the EXPRESS experiment has designed and implemented a solution for SDN controller selection by the Wireless Mesh Routers. The main idea is to assign more responsibility to the controlled nodes (WMRs), letting them take the decision about which switch has to take mastership of the node. Therefore we named this procedure controller *selection* rather than master *election*. The nodes will monitor a set of SDN controllers that can potentially assume the master role and will implement a selection algorithm to choose the preferred controller among the set of reachable controllers (see Fig. 3). Note that WMRs and controllers have the same information about the status of the network (excluding transient conditions), because they share the OLSR vision of the topology. In particular, the WMRs are directly involved in the OLSR topology dissemination while the controller extracts the topology information from a nearby WMR. Therefore, from the topology discovery point of view the WMR acquires topology information even before the controller. Moreover, a WMR can directly check the connectivity with potential controllers trying to establish TCP connections towards them (or monitoring the liveliness of established TCP connections).

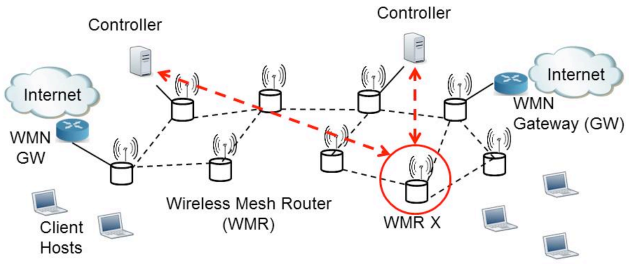

Fig. 3. Controller selection by a Wireless Mesh Router

In the designed procedure a WMR connects only toward a single controller at a given time. This is different from the classical approach where a switch connects in parallel with several controllers. The procedure is performed in the WMR with the help of the EFTM (External Flow Table Manager) entity shown in Fig. 2. The EFTM entity is in charge to perform the master selection procedure and will instruct the switch to connect to the selected controller at a given time.

Performing the master selection on the WMR side has some advantages in our scenario. The first advantage is that each OpenFlow switch will be connected with a single controller at a time, and no conflicting rules can be injected. The second advantage is that a run-time coordination mechanism among controllers is not needed, each controller can operate on its own, obviously all the controllers should follow a consistent service logic.

IV. Experimenting over the OpenLab Testbeds

As described above, we designed and developed novel algorithms and procedures in order to address the aforementioned challenges, thus we needed to test them under real world settings and in the largest scale possible. To this end, we took advantage of the OpenLab facility that provides the unique capability to deploy and evaluate experiments easily in a mid-scale environment exploiting a plethora of different kinds of resources.

A. Taking advantage of the tools provided by OpenLab

The main issue when you conduct an experiment involving several heterogeneous resources is burden related to their control and configuration, as well as their synchronization during the experiment. This was addressed easily by using the OMF framework, which enabled us to configure and control the different kinds of nodes that were part of the experiment, through a single script. In this script, which is written in a simple, domain-specific language provided by OMF, namely the OEDL [7], we described the required initial configuration of the nodes and specified a list of events and associated tasks, as well. The list of tasks includes the drop of a link, the sleep for a specified time period or measurement points for collecting data.

Another big issue that OpenLab tools assisted us to address is the gathering of the measurements generated during the experiment, in a unified way. The collection of all those data is handled by the OML [8], which is again provided by OMF. In this way, we defined measurement points in the experiment description script and OML handled the collection of the experimental results and their storage in a database for further processing.

The necessary steps for conducting the experiment on the federated testbeds and the tools we used are:

- The reservation of the resources through mySlice portal,
- The development of the experiment description using OEDL,
- The development of the experiment scripts through OMF,
- The execution of the experiment through a single script and the collection of the measurements through OML.

In the following subsection, we describe the main challenges faced and the methods followed towards the successful deployment of the EXPRESS experiment on the OpenLab federation.

B. Main challenges during the deployment

The experiment is performed across three different OpenLab testbeds, two wireless testbeds (NITOS [9] and w-iLab.t [10]) that supports different Wireless Mesh Networks and a wired testbed (PLE - Planetlab Europe [11]) that is used to emulate a “backbone” link interconnecting the two Wireless Mesh Networks. The backbone link was implemented through the establishment of an Ethernet over UDP tunnel across PlanetLab testbed (actually two UDP tunnels bridged with a Virtual Switch on a PLE node, as shown in Fig. 4).

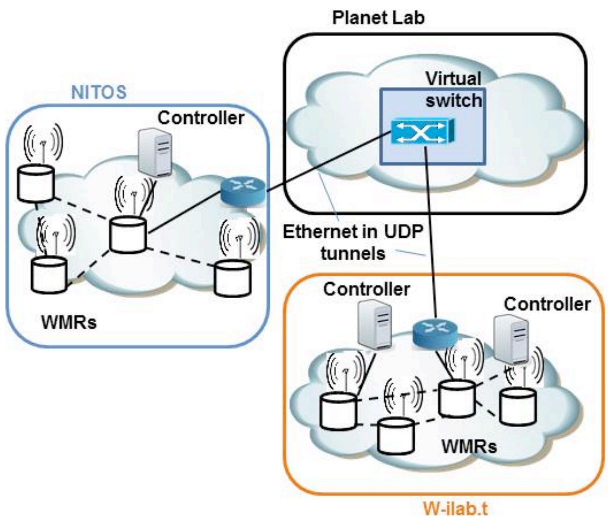

Fig. 4. Experiment over PlanetLab, NITOS and w-iLab.t

In most cases, the description of the experiment in the OMF language is a straightforward procedure, so was in our case. On the other hand, one of the difficulties we faced during the deployment is that most of the wireless nodes in the testbeds are inside the coverage area of all the other wireless nodes in their testbed, making difficult to emulate topologies where nodes lie more than two-hops away from each other. In order to face this difficulty, we filtered the packets at the receiving node in order to emulate the desired experimental topology – thus all packets received by nodes that are not in the mutual communication range were discarded. Further details on the aforementioned procedure can be found in the OpenLab Deliverable D3.11 [12].

Another problem we had to face is the private IP addressing that NITOS and w-iLab.t use for their wireless nodes. Taking under consideration the advantages and disadvantages of all the different options for dealing with this problem, we concluded that a solution based on NAT is the most appropriate and applicable one for our situation. Further details on the aforementioned procedure can be found in the OpenLab Deliverable D3.10 [13].

Since we successfully implemented all challenges during the deployment phase, we designed and developed two different types of experiments.

C. *Description of the experiments*

In the combined OpenLab testbed, we run two types of experiments, respectively denoted as "network merging" and "network partitioning" experiment. In the network merging experiment we start with two mutually disconnected network sets (each one with a SDN controller) and then reactivate a wireless link between two WMRs residing in different sets. In the network partitioning experiment we deliberately deactivate a wireless link that interconnects two sets of the networks, each one including (at least) a SDN controller. In both experiments the following simple control logic is run by the WMR nodes. Each WMR node has a list of SDN controllers that can potentially take control of the node, ordered by priority. The SDN controllers are listed with their IP address. The WMR will periodically check which controller IP addresses are reachable looking at the IP routing table established by means of the OLSR protocol. The WMR will try to connect to all reachable controllers (and will check the liveliness of the connection for the currently selected master SDN controller). Then it will select the highest priority controller among the ones to which it has successfully established a connection. In the experiment, the priority list of the preferred controllers was simply preconfigured in the nodes (using a configuration file). In a real life implementation the priority list could be transferred by a SDN controller to the WMR node and updated when needed.

In both types of experiments, we measured the time needed for the WMRs to connect to the highest priority controller after the event that determined the network merging/partitioning. In the network merging experiment, this time interval can be decomposed into two phases: in the first phase the IP routing (OLSR) will properly reconfigure the connectivity in the control network (OLSR routing procedure). In the second phase our proposed controller selection algorithm will operate to select the highest priority controller (controller selection procedure). In the network partitioning experiment, the WMR node does not rely on OLSR to detect that a controller is no longer reachable, but it will perform its own connectivity check, achieving a faster reaction time.

The rationale of the two experiments is to demonstrate that the controller selection procedure operates within the same time scale than the OLSR restoration procedure. If we accept the performance of OLSR in routing packets over the WMN, we will likely accept the performance of the proposed SDN based approach offering traffic engineering services in the WMN.

D. *Experiment setups*

To perform the network merging experiment, the tunnel between the two wireless testbeds is initially not active, and the WMRs are connected to their respective controllers available within their own local testbed. As soon as the tunnel across Planet Lab Europe is activated, messages start to flow from one wireless testbed to the other and the WMR nodes belonging to w.iLab-t testbed learn the IP route towards the remote controller in NITOS. The EFTM entity implemented into each of the WMR checks if a controller is actually active at that IP address by trying to establish an OpenFlow protocol connection. When this check is positive, the entity chooses to connect to the highest priority controller, in this case the one in the remote wireless testbed (NITOS).

To perform the network partitioning experiments we drop the tunnel established through PlanetLab and measure the time needed by WMRs in each wireless testbed to connect to their local controllers.

V. EXPERIMENTS RESULTS

In this section we present the experiment results. We do not aim to provide an in depth technical analysis of the results. We only illustrate what results we obtained and show how they can support the answers to the questions we have identified in section III.

A. *Network merging experiment*

In this experiment we evaluate the time needed for the WMRs to connect to a higher priority controller after the merging of two network partitions. As shown in Fig. 5, this time is decomposed in two phases, network connectivity and master selection. The former one considers the time needed for the routing protocol to setup the IP routes in all WMRs taking into account the merged network topology. We measure it by trying to send ping requests from a WMR to the controller that was not reachable before the merging event. In our experiments it averages to 15 seconds. In fact, according to the OLSR routing protocol mechanisms, three "Hello" messages need to be received in order to declare a link up and the default interval for sending OLSR Hello messages is 5 seconds. Starting from this time instant we measure the interval needed for the WMRs to disconnect from old controller and connect to the one with higher priority. In our implementation the EFTM periodically tries to establish a connection with all controllers that have been discovered, starting from the highest priority one. The polling period is 3 seconds. In the experiment we measured an average of more than 1.5 seconds for the latest connected WMR, which is consistent with the expectations.

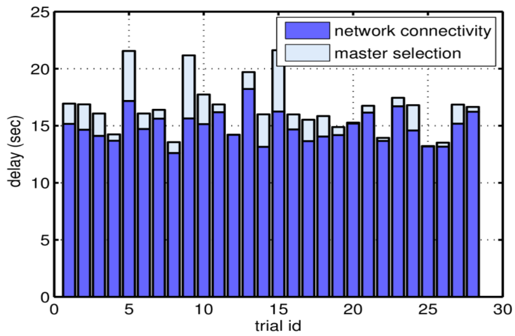

Fig. 5. Network merging experiment

B. Network partitioning experiments

In this second set of experiments we consider the partitioning of the network: starting from a connected network as shown in Fig. 4, we disconnect one of the tunnels across PlanetLab Europe. In Fig. 6 we report the evaluation of the time needed by the WMRs in the w-iLab.t testbed to disconnect from the remote controller in NITOS and connect to the local controller. In this case the WMRs does not rely on OLSR to discover that a controller is not reachable, as it would require more than 15 seconds considering the default OLSR configuration (3 Hello intervals of 5 s needed to declare the link down). The ETFM periodic controller polling procedure (running with a 3 seconds period) considers a 2 seconds timeout before declaring that a controller is down. With this procedure, an average master selection delay of 5.5 seconds is measured.

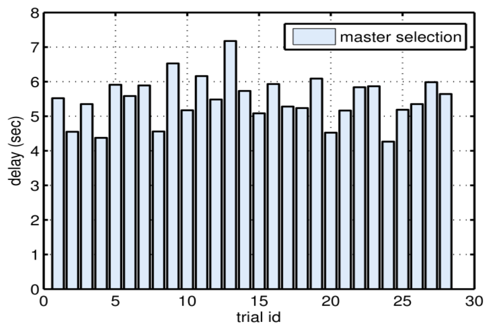

Fig. 6. Network partitioning experiment: master selection delay

VI. CONCLUSIONS

This paper presents two major contributions. The first one is related to the question whether SDN can be efficiently used in a dynamic environment with intermittently connected networks. A solution has been designed for this purpose. Nevertheless, it is critical to evaluate such a proposal in a practical setting. The second value of the paper is to demonstrate that the OpenLab federation of heterogeneous testbeds provides the mean to configure and experiment the solution derived in order to assess its performance. The experiment was conducted with a reduced effort thanks to the tools provided by the OpenLab facility. Despites the fact that the experiment involved three different and heterogeneous testbeds, the performance of the proposed solution has been captured and the results helped to answer the suitability of SDN for this type of complex environments.

ACKNOWLEDGMENTS

This work has been supported by the OpenLab project, "OpenLab: extending FIRE testbeds and tools ", funded by the EC under FP Grant agreement no: 287581.

REFERENCES

[1] T. Rakotoarivelo, M. Ott, G. Jourjon, and I. Seskar, “OMF: a control and management framework for networking testbeds”, SIGOPS Oper. Syst. Rev. 43, 4 (January 2010), pp. 54-59

[2] A. Dixit, F. Hao,S. Mukherjee,T.V. Lakshman, R. Kompella, “Towards an Elastic Distributed SDN Controller”, ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking (HotSDN 2013), August 16, 2013, Hong Kong

[3] A. Detti, C. Pisa, S. Salsano, N. Blefari-Melazzi, “Wireless Mesh Software Defined Networks (wmSDN)”, CNBuB Workshop at IEEE WiMob, France, Lyon, 7 October 2013

[4] Open Networking Foundation (ONF) “OpenFlow Switch Specification”, Version 1.4.0, October 14, 2013

[5] S. Salsano, G. Siracusano, A. Detti, C. Pisa, P. L. Ventre, N. Blefari-Melazzi, “Controller selection in a Wireless Mesh SDN under network partitioning and merging scenarios”, submitted paper, available at http://arxiv.org/abs/1406.2470

[6] S. Salsano, P. L. Ventre, L. Prete, G. Siracusano, M. Gerola, E. Salvadori, “OSHI - Open Source Hybrid IP/SDN networking (and its emulation on Mininet and on distributed SDN testbeds)”, EWSDN 2014, 1-3 September 2014, Budapest, Hungary

[7] OEDL - The OMF Experiment Description Language http://mytestbed.net/projects/omf54/wiki/OEDL-5-4

[8] OML - Measurement Library http://mytestbed.net/projects/oml/wiki/

[9] NITOS testbed http://nitlab.inf.uth.gr/NITlab/index.php/testbed

[10] w-iLab.t testbed http://ilabt.iminds.be/wilabt

[11] PlanetLab Europe (PLE) testbed http://www.planet-lab.eu/

[12] S. Salsano (editor) “EXPRESS final evaluation and overall report”, Deliverable D3.11 of EU FP7 Project “OpenLab”

[13] S. Salsano (editor) “EXPRESS – Final architecture and design, evaluation plan”, Deliverable D3.10 of EU FP7 Project “OpenLab”

[14] http://en.wikipedia.org/wiki/Wireless_community_network

[15] T. Clausen (Ed.), P. Jacquet (Ed.), “Optimized Link State Routing Protocol (OLSR)”, IETF RFC 3626, October 2003