

Integration of Virtual SDN and NFV Laboratory with NEWTELP Platform

Martin Medvecký¹, Radoslav Vargic¹, Juraj Londák¹, Pavol Podhradský¹, Peter Trúchly²

¹Faculty of Electrical Engineering and Information Technology,

²Faculty of Informatics and Information Technologies,

Slovak University of Technology in Bratislava Ilkovičova 3, 812 19, Bratislava, Slovakia

martin.medvecky@stuba.sk, radoslav.vargic@stuba.sk, pavol.podhradsky@stuba.sk

Abstract – The paper provides a concept and architecture of the Virtual SDN and NFV Laboratory and its integration with NEWTELP platform – a learning platform developed with the EU Horizon 2020 NEWTON Project. The Virtual SDN and NFV Laboratory was proposed as a virtual laboratory for teaching and research activities in the field of Software Defined Networking (SDN) and Network Function Virtualization (NFV) technologies. The paper presents the concept, implementation and testing of integration of Virtual SDN and NFV laboratory with NEWTELP platform.

Keywords - SDN; NFV; Network virtualisation; Virtual laboratory experiments; NEWTELP platform, xAPI

I. INTRODUCTION

Software-Defined Networking (SDN) [1] is an emerging ICT architecture that decouples the network control and forwarding functions enabling the network control to become directly programmable and the underlying infrastructure to be abstracted for applications and network services.

Unlike existing packet network architectures, which use nodes in which features, packet control logic and packet forwarding hardware are integrated the concept of SDN is based on separation of the control and data forwarding plane in network. The centralized control plane implemented in software executed on general purpose processors can bring many advantages to networking - especially speeding up innovation, new network features development and deployment, configuration of all network devices, as well.

Network functions virtualization (NFV) [2] is a network architecture concept that uses the technologies of IT virtualization to virtualize entire classes of network node functions into building blocks that may connect, or chain together, to create communication services. Virtualization means that a network function and part of the infrastructure are implemented in software and therefore the NFV software architecture is an important aspect of the NFV architectural framework.

The architecture of NFV technology [3] is based on the following components:

- NFVI (NFV Infrastructure) [4] - provides virtual resources needed to support the implementation of virtualized network.

- VNF (Virtualized Network Functions) [4] - software implementation of network functions that is able to run by NFVI and may be accompanied by EMS (Element Management System), which manages the VNF.
- NFV MANO (Management and Orchestration) [5] - covers orchestration and lifecycle management of physical and/or software tools that support the virtualization and infrastructure lifecycle management VNFs.

II. SYSTEM DESCRIPTION

A. Virtual SDN and NFV Laboratory

The Virtual SDN and NFV laboratory was designed for teaching and research purposes to enable study, verification, testing and examination of SDN and NFV technologies by a secure and controlled manner for both local and remote users [6]. The block diagram of the Virtual SDN and NFV laboratory is shown in Fig. 1.

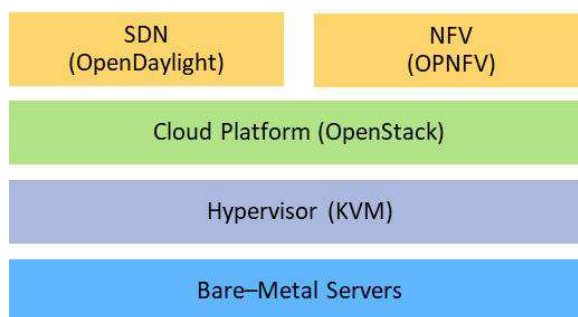


Figure 1. Virtual SDN and NFV laboratory - functional view

The basis of the Virtual SDN and NFV laboratory is a cloud created by the bare-metal servers, hypervisors and the OpenStack cloud platform. The interconnection of individual servers is accomplished via VLAN-enabled Ethernet switches, allowing for the creation of three separate networks (management network, instance network and storage network). In addition to these cloud networks, software-defined networks will be created as needed, using virtually open vSwitch switches implemented through the virtualization platform. The connection to external networks is made using a router.

B. NEWTELP platform

NEWTELP platform [7] main goals are the development, integration and dissemination of innovative technology enhanced learning (TEL) methods and tools. The platform intends to create or interconnect existing state-of the art teaching labs and to build a pan-European learning network platform that supports fast dissemination of learning content to a wide audience in a ubiquitous manner. The platform is based upon a multi-tier, client-server architecture. Multi-tier approach provides the flexibility to support ever-changing business (often legislatively driven) requirements. Its key feature is that it separates out business logic, client access technology and centrally held data into discrete layers with standard, open interfaces. The architecture has securely managed interfaces to isolate systems from illegal access.

Functional view of the NEWTELP platform is depicted in the Fig. 2. The main component is the Course Management System and acts as a coordinator for all the functionalities that compound the system. This module combines most of the content and services defined in the learning and teaching process and come as a bridge for different components of the platform. The Innovative TEL Solutions (ITS) component consists of three sub-components: Gamification, Personalisation and Adaptation. Personalisation provides the required supports to learners in achieving the course specified learning goals. This is performed by identifying the learner needs (knowledge gaps) and recommending a set of relevant contents to address that existing gap. Adaptation provides content adaptation to the learners in order to increase learner perceived quality, learning outcome and improve the learning process when multimedia and mulsemmedia educational content is delivered. Gamification provides configuration of gamification elements/rewards (e.g. points, badges, leaderboards and so on) and mechanics (e.g. rewarding rule based on the defined elements and what learning actions or activities must be connected). The component that interacts with the Gamification component is course management system.

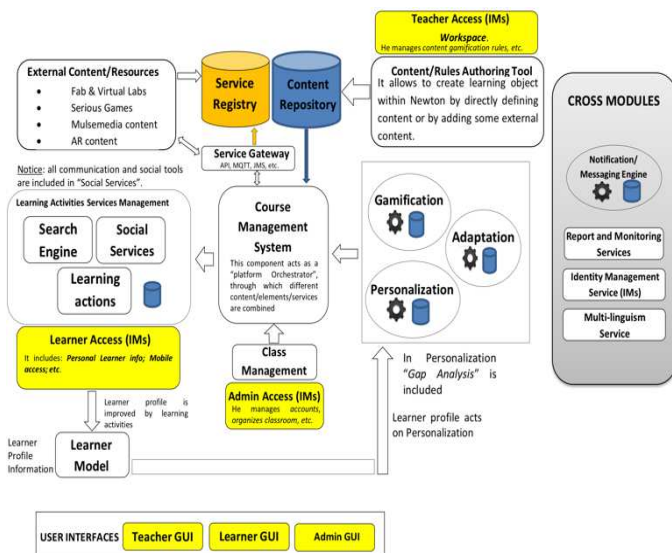


Figure 2. NEWTELP platform - functional view

III. PROPOSED INTEGRATION

The Virtual SDN and NFV laboratory is integrated into the NEWTELP platform as virtual lab as sketched in the Fig. 3.

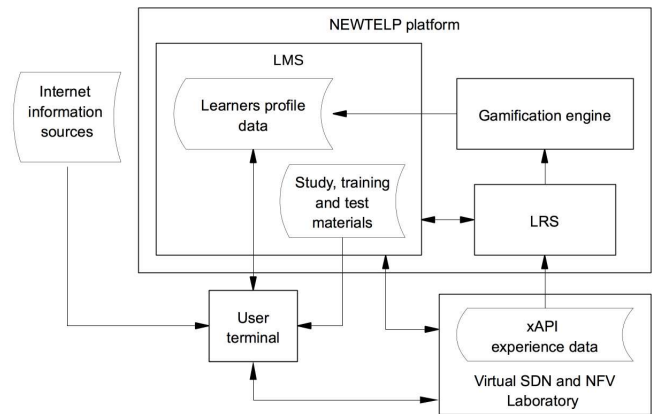


Figure 3. Functional view of the Virtual SDN and NFV laboratory integration into NEWTELP platform

Main part of the integration is performed using reporting engine. Engine requests and maintains necessary authorization information to be able to report the experience data to the NEWTELP platform. The reporting authorization information is obtained from Learning Management System (LMS) module using JSON based RESTfull web service over HTTPS (see Fig. 4 for details). For reporting it is used the Experience API (xAPI) standard¹, formerly known also as Tin Can. xAPI specification² is aimed for learning technology and makes possible to collect data about the wide range of experiences. The xAPI captures data in JSON format, overcomes some of the SCORM limitations, is community-driven and free to implement. The xAPI Reports are send to the Learning Recording Store (LRS) subsystems for the platform. From there, the relevant events are propagated into gamification module, where they are evaluated using configured rules. The resulting achievements and point are visible in the learners profile in the platform. Reported are the user experience data, which means important information about starting, completing, terminating the stages designed in the LAB scenarios.

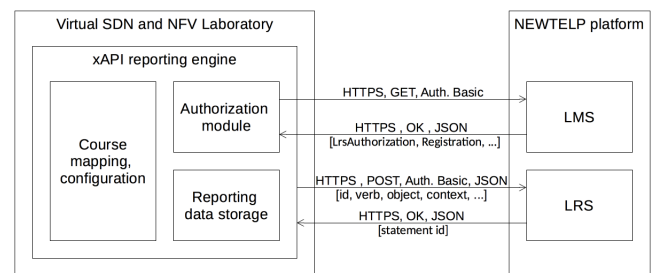


Figure 4. Technical view of Virtual SDN and NFV laboratory integration into NEWTELP platform

¹ www.xapi.com

² <https://github.com/adlnet/xAPI-Spec/blob/master/xAPI.md>

The reports are transmitted as xAPI statements using pre-defined verbs. Statements also contain time information, evaluation of the activity and mapping to particular course. Below is an example xAPI statement sent from Virtual SDN and NFV Laboratory to the LRS.

```
{
  "id": "6f6664d1-5811-460f-a4fe-93e3b1bb168f",
  "actor": {
    "objectType": "Agent",
    "mbox": "*****",
    "name": "*****"
  },
  "verb": {
    "id": "http://adlnet.gov/expapi/verbs/initialized",
    "display": {
      "en-US": "Initialized"
    }
  },
  "object": {
    "id": "https://www.newtelp.eu/VL/SDNNFV/course01",
    "objectType": "Activity",
    "definition": {
      "name": {
        "en-US": "SDN/NFV VR LAB"
      },
      "description": {
        "en-US": "SDN/NFV Virtual Lab"
      },
      "type": "http://adlnet.gov/expapi/activities/lesson"
    }
  },
  "context": {
    "registration": "384e1535-8bfa-4514-a224-ad4d94a04b55",
    "contextActivities": {
      "parent": [
        {
          "id":
            "https://www.newtelp.eu/VL/SDNNFV/course01",
          "objectType": "Activity"
        }
      ]
    }
  },
  "timestamp": "2018-02-13T10:35:44.696+02:00",
  "stored": "2018-02-13T10:48:53.220Z"
}
```

The statement is shown as stored in the LRS, already with the parameters added during receive process - the statement id (the first show parameter) and stored timestamp (the last parameter). In the “context” part, there are also visible the NEWTELP introduced “parent” mapping mechanism that helps to identify the activity in the tree of activities in the particular application/Virtual lab that uses xAPI to report his activities to the NEWTELP platform.

IV. IMPLEMENTATION

The integration with the platform is done mostly in the xAPI reporting engine as shown on Figure 4. The engine scans the activities logged in the Virtual SDN and NFV laboratory and updates his own reporting data storage. Reporting engine also maintains the authorization information (fetched from LRS) and course mapping information (schedule and user based). Based on the configuration it buffers the experience data or reports them immediately to the LRS using the xAPI. The buffering mechanism was introduced to have the opportunity to validate the data by trainer before reporting them to platform. The xAPI reporting engine is implemented in JAVA 10.

V. TESTING

Testing was carried out as part of pilot testing on two groups of students (up to 30 students in each group) which have completed SDN lessons.

For testing purposes, the Virtual SDN and NFV laboratory was extended with the ability to create SDN networks in Mininet³. Mininet is a simple and powerful network emulator enabling to create a network of virtual hosts, switches, controllers, and links on a single machine. Mininet allows running real kernel, switch and application code and therefore it is well suited for emulation of SDN networks. Individual tasks were provided in Mininet network simulator using a Mininet’s graphical user interface tool MiniEdit which simplifies the SDN network creation. The communication between the SDN network switches (emulated by Mininet) and SDN controller was captured and analyzed using a Wireshark. Wireshark is a widely-used network protocol analyzer with OpenFlow protocol support. All applications necessary for experimenting with SDN networks (Mininet, Miniedit and SDN controller) were pre-installed with the other necessary applications (Wireshark, web browser etc.) in a single virtualized environment (virtual machine with 2 CPU and 4 GB RAM running on Ubuntu 16.04 LTS) – see Fig. 5.

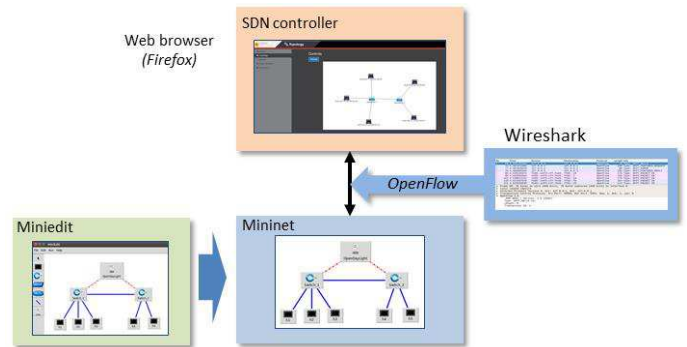


Figure 5. Applications installed in the VM used for testing

Each of the students worked in their own environment (virtual machine) and their activities were recorded locally and on the central logging server (see Fig. 6).

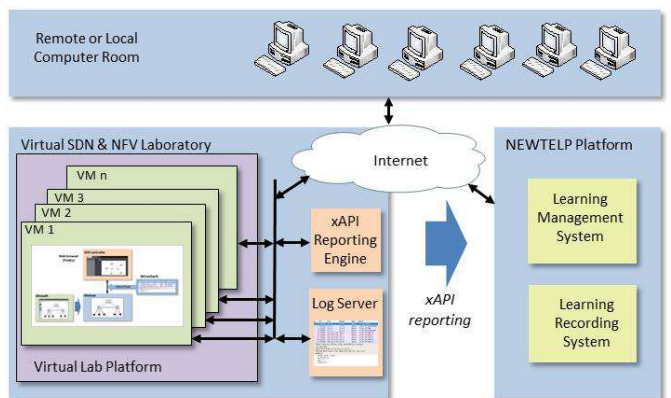


Figure 6. Schematic view of integration testing

³ www.mininet.org

Testing was carried out as part of pilot testing in the “Network virtualization – SDN/NFV” course that was carried in two vocational schools in Slovakia. Except for Virtual SDN and NFV laboratory experiments, the course contained also the main study material, entry test, worksheets, auto-tests and final test. The student’s activities were logged locally on the particular virtual server and on the Virtual SDN and NFV laboratory central log server. Processed activities were passed to the reporting data storage of the xAPI reporting engine and passed to the NEWTELP platform. The main scenario that performed the students during the pilot testing had the following steps:

1. Startup of necessary applications (Mininet - SDN network simulator, Floodlight - SDN controller, Wireshark – packet analyzer, Firefox – user interface).
2. Creation of SDN network (design, configuration, deployment, operation).
3. Execution and Fulfillment of pre-designed tasks.

The pre-defined tasks were:

1. Analysis of the initial handshaking communication between controller and switches.
2. Analysis of the communication between controller and switch and hosts and switch during communication between hosts connected to the same switch.
3. Analysis of the communication between controller and switches and hosts and switches during communication between hosts connected to different switch.
4. As previous one, but more complex data communication scenario (web page retrieval) was performed.

The verification of the proper reporting using xAPI was realized manually in two ways:

- Based on the comparison of log files on particular virtual servers, central log and messages sent to NEWTELP platform. Verification included: login times, application - correctness of startup. Creation of scheme in Mininet (automatically the existence, manually the quality/completeness), generated xAPI statements (initiated, completed, terminated).
- According to worksheets elaborated by students during performance of the main scenario in the Virtual SDN and NFV laboratory.

The verification revealed that the most crucial part is the complexity and reliability of the evaluation mechanism implemented in the reporting engine. Automated time-based evaluation of activities and output dependencies gave satisfactory results. Quality check of the Mininet scheme by the trainer remains as necessary manual step.

VI. CONCLUSIONS

In this contribution we described the integration of the Virtual SDN and NFV laboratory into the NEWTELP platform within the Horizon 2020 NEWTON project. Testing was carried out as part of pilot testing in course carried in more schools. For the further work we identified as important to keep enhancing the quality of the automatic evaluation mechanisms in the reporting engine. The solution can be furthermore optimized to report more low-level activities into the platform and implement more the high-level rules in the gamification engine.

ACKNOWLEDGMENT

Research described in the paper was financially supported by the H2020 project NEWTON, No. 688503 and VEGA project INOMET, No. 1/0800/16 and VEGA 1/0836/16.

REFERENCES

- [1] Benzekki Kamal et al. Software - defined networking (SDN): a survey., Security and Communication Networks 9, no. 18 (2016): 5803-5833.
- [2] Chiosi, M., Wright, S., Clarke, D., Willis, P., et al.: Network Functions Virtualisation: Network Operator Perspectives on Industry Progress, http://portal.etsi.org/NFV/NFV_White_Paper2.pdf
- [3] ETSI GS NFV-SWA 001 V1.1.1, Network Functions Virtualisation: Virtual Network Functions Architecture. ETSI, December 2014
- [4] ETSI GS NFV-INF 001 V1.1.1, Network Functions Virtualisation: Infrastructure Overview. ETSI, December 2014
- [5] ETSI GS NFV-MAN 001 V1.1.1, Network Functions Virtualisation: Management and Orchestration. ETSI, December 2014
- [6] J. Londák, M. Medvecký, and P. Podhradský, “Virtual SDN and NFV laboratory - architecture and implementation”, in Proceedings ELMAR-2017: 59th International symposium. Zadar, Croatia. 18-20 September, 2017. Zagreb : University of Zagreb, 2017, pp. 197-200. ISSN 1334-2630. ISBN 978-953-184-230-3.
- [7] Horizon 2020 NEWTON Project (www.newtonproject.eu), available online