

A Flexible Multi-PoP Infrastructure Emulator for Carrier-grade MANO Systems

Manuel Peuster*, Sevil Dräxler*, Hadi Razzaghi Kouchaksaraei*,
Steven v. Rossem†, Wouter Tavernier† and Holger Karl*

*Paderborn University: {manuel.peuster, sevil.draexler, hadi.razzaghi, holger.karl}@upb.de

†Ghent University - imec: {steven.vanrossem, wouter.tavernier}@intec.ugent.be

Abstract—Developing a virtualized network service does not only involve the implementation and configuration of the network functions it is composed of but also its integration and test with management solutions that will control the service in its production environment. These integration tasks require testbeds that offer the needed network function virtualization infrastructure (NFVI), like OpenStack, introducing a lot of management and maintenance overheads. Such testbed setups become even more complicated when the multi point-of-presence (PoP) case, with multiple infrastructure installations, is considered.

In this demo, we showcase an emulation platform that executes containerized network services in user-defined multi-PoP topologies. The platform does not only allow network service developers to locally test their services but also to connect real-world management and orchestration solutions to the emulated PoPs. During our interactive demonstration we focus on the integration between the emulated infrastructure and state-of-the-art orchestration solutions like SONATA or OSM.

I. INTRODUCTION

The recent rise of the network function virtualization (NFV) concept introduced a new level of flexibility into the networking domain by allowing fully automated network service deployments and runtime reconfigurations. These automated processes are triggered and controlled by management and orchestration (MANO) systems which are a key entity in such setups [1] and have recently been the focus of many research and open-source projects [2]–[4]. Such MANO systems do not only control the resources, e.g., the provisioning of virtual machines in which virtual network functions (VNFs) are executed, but also the configuration and setup of the actual VNF software stacks and their interconnections across multiple data centers, typically called points of presence (PoP). This introduces a dependence between the MANO system and the managed services, which has to be tested by network service developers to verify that their services play well with all candidate MANO systems. Examples include tests of configuration mechanisms as well as the validation of scaling and placement decisions and their effects on the network service. However, these validation tasks become very complicated and resource consuming, even if they could be automated in a DevOps pipeline [5], because testbed installations for each PoP and its attached virtualization infrastructure are required.

In this demo, we will show how a network service developer can utilize our multi-PoP infrastructure emulation platform, called *MeDICINE* [6], to quickly setup NFV test environments for arbitrary network services. After motivating this work

by outlining the challenges of NFV development and test processes in Section II, we give more details about our NFV emulation platform and the used abstractions to integrate with existing MANO systems in Section III. Finally, we describe the planned demonstration in Section IV.

II. NFV SERVICE DEVELOPMENT & TESTING

The development of NFV services involves a range of development, testing and integration tasks. Services consist of a number of network functions, each typically developed and tested in isolated environments (unit testing). After this, network functions are combined and potentially executed in an ordered manner (service function chaining) to create the service. This service needs again to be tested which requires an adequate testing environment that offers reproducibility as well as debugging and monitoring functionalities. Finally, the integration between service, the involved functions, and the MANO system of the target platform needs to be validated. Thus, the canonical NFV development and testing process follows a process similar to the following list of steps:

- 1) Development of individual network functions.
- 2) Test deployment of individual network functions on controlled infrastructure. This includes the deployment in different infrastructure setups (e.g., single-PoP vs. multi-PoP) and the use of different MANO systems.
- 3) Development of the composed network service, including service chaining and interactions among individual functions of the service.
- 4) Test deployment of the complete network service on controlled infrastructure. Again including different infrastructure setups and the use of different MANO systems.

Deploying these single- or multi-PoP infrastructure environments to be controlled by a range of MANO solutions in an on-demand setting can be very costly in terms of required hardware and operational as well as set-up costs. A quick and low-cost network function virtualization infrastructure (NFVI) and MANO system testbed emulation mechanism, as described in step 1 to 4, is therefore extremely desirable. In previous work [6], we presented *MeDICINE* as highly configurable NFVI emulator. In this demo we showcase its recent extensions to support multiple MANO frameworks allowing interoperability tests between developed network services and MANO systems. This is not possible with existing test and troubleshooting tools [7] and can so far only be done with

platforms that are limited to execute simplified network functions [8] or with expensive, full-featured NFVI deployments.

III. BUILDING A MULTI-POP NFVI EMULATOR

Our goal is to design and implement a system that allows a user to define and setup arbitrary, complex multi-PoP environments. Each PoP emulates a small data center with its independent NFVI and the attached virtual infrastructure manager (VIM). These PoPs can then be used and controlled by real-world, carrier-grade MANO systems, like the SONATA service platform [2] or OSM [3].

A. Background: *MeDICINE* platform

As described in [6], *MeDICINE* is a Mininet-based, open-source emulation platform that allows the execution of arbitrary application containers inside emulated PoPs connected to user-defined topologies on a single physical or virtual machine. These topologies are described with Python scripts and provide the full set of Mininet features, like user-defined delays or loss rates between the PoPs. Using *MeDICINE* in an NFV context allows us to run arbitrary network functions inside emulated multi-PoP topologies as long as the used functions can be packaged as containers (e.g. Docker).

B. Integration concept

We identified two different approaches to enable the easy integration between our emulated environment and real-world MANO systems. The first approach is to implement customized driver plugins that support our emulation platform for each MANO system. The second approach is to add already standardized interfaces to our emulation platform which can then directly be used by off-the-shelf MANO installations. Since we want to support as much MANO solutions as possible and keep the development overhead low, we decided to go for the second approach. More specifically, we realized it by adding OpenStack-like interfaces to our platform, because practically every MANO system comes with driver plugins for the OpenStack APIs.

As illustrated in Fig. 1, our platform automatically starts OpenStack-like control interfaces for each of the emulated PoPs which allow MANO systems to start, stop and manage VNFs. This approach has a good level of abstraction and we have been able to implement the needed set of API endpoints required by the SONATA and OSM orchestrators in less than 3.8k lines of Python code. Specifically, our system provides the core functionalities of OpenStack’s *Nova*, *Heat*, *Keystone*, *Glance*, and *Neutron* APIs¹. Even though not all of these APIs are directly required to manage VNFs, all of them are needed to let the MANO systems believe that each emulated PoP in our platform is a real OpenStack deployment.

From the perspective of the MANO systems, this setup looks like a real-world multi-VIM deployment, i.e., the MANO system’s southbound interfaces can connect to the OpenStack-like VIM interfaces of each emulated PoP.

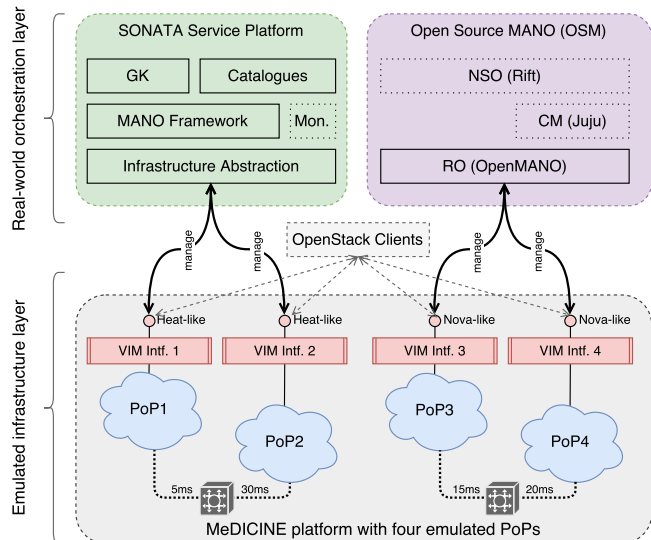


Fig. 1: Demonstration architecture in which two state-of-the-art MANO systems (SONATA [2] and OSM [3]) control multiple PoPs of the virtual infrastructure layer that is emulated by our *MeDICINE* platform [6].

IV. DEMONSTRATION

The objective of the planned demo is two-fold. First, we explain how a user can use the *MeDICINE* platform to specify and emulate arbitrary multi-PoP topologies and run containerized applications, like VNFs, in these PoPs. Second, we showcase how a user can connect a MANO system of his choice to the emulated infrastructure to build a full-featured, local, multi-PoP NFVI testbed. This is demonstrated by connecting two reference MANO systems, namely the SONATA service platform and OSM, to our platform. Finally, we will demonstrate how a network service, consisting of multiple chained VNFs, is deployed on the emulated infrastructure using the MANO systems mentioned above.

A. Used network service

For the demonstration, we designed an example network service that acts as a content delivery network (CDN) that will be showcased in a video streaming context. The service consists of three containerized VNFs. The first VNF of the service is a caching proxy solution based on Squid², the second VNF is a layer 4 forwarding element based on Socat³ and the third VNF is a web service based on Apache⁴ that acts as content provider. Figure 2 shows the service and how it is accessed by users during the demo to proof that the deployed service instance is actually running on our emulation platform.

We implemented three different versions of the service using the HEAT template language, SONATA description format, as well as the OSM descriptor language. Using this, we

²<http://www.squid-cache.org>

³<http://www.dest-unreach.org/socat/>

⁴<http://httpd.apache.org>

¹<https://developer.openstack.org/api-guide/quick-start/>

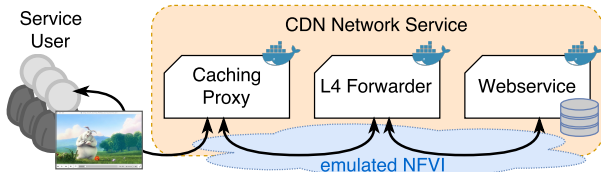


Fig. 2: A CDN-like demo network service consisting of three interconnected VNFs.

demonstrate how we can deploy the same service on our platform using different MANO systems. We also have some alternative VNF implementations available which we can use to demonstrate descriptor modifications and re-deployments during our demo.

B. Demonstration steps

The demonstration includes the following steps:

- 1) Specification and setup of the emulated multi-PoP topology.
- 2) Deployment and management of the demo service on our platform using the stock *OpenStack command line clients*.
- 3) Registration of emulated PoPs to real-world NFV MANO systems, namely SONATA and OSM.
- 4) On-boarding, deployment, and management of the example service on the emulated infrastructure using the connected MANO systems.
- 5) Live interaction with the running service to validate that it is properly deployed and functional, e.g., by streaming content through the service and monitoring performance metrics, like packet counters of the running VNF.

C. Visualization and monitoring

The outcome of each demonstration step will be visible on various graphical interfaces and dashboards as shown in Fig. 3. This is done with the help of our monitoring framework that is available for the *MeDICINE* platform. The monitoring framework is able to gather, export, and visualize arbitrary, user-configured metrics and offers a user-friendly way to validate or debug the service’s functionality and resource usage. It allows the service developer to describe custom sets of monitoring metrics independently of the service’s descriptor language and can monitor any kind of service, e.g., by internally querying the traffic counters of each virtual interface connected to one of the emulated PoPs.

D. Requirements

The demo can be executed either locally on a single laptop running the entire emulation platform or remotely on a server. It requires a power outlet for a laptop and one or two large screens to show the dashboards and monitoring data. A stable Internet connection is only required if the remote fallback demonstration is used. A wall or stand to mount a poster will help to describe the demo to the audience. There is a

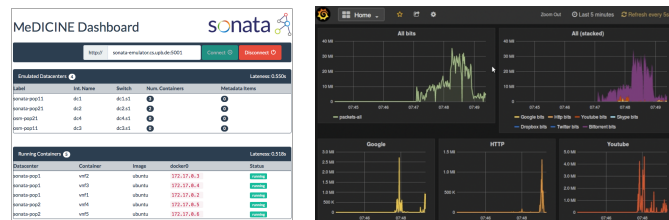


Fig. 3: Demonstration dashboards showing VNF instances running inside the emulated PoPs and the corresponding monitoring information, like CPU, memory and network performance.

YouTube video available that shows some parts, i.e. the OSM integration, of the planned demo⁵.

V. CONCLUSION

Our demonstrated system provides a novel emulation environment which lowers the barrier of testing network services by not requiring full-featured NFVI testbed installations (e.g., OpenStack clusters). It is in particular useful for integration tests between user-developed network services and real-world MANO solutions which can be connected through OpenStack-like VIM interfaces. In addition to this, researchers can use the system to conduct experiments with novel MANO approaches using many emulated PoPs. The demonstrated software is open source and available on GitHub⁶.

ACKNOWLEDGMENTS

We would like to thank our project group students who implemented some of the used interfaces. This work has been partially supported by the SONATA project, funded by the European Commission under Grant number 671517 through the Horizon 2020 and 5G-PPP programs (www.sonata-nfv.eu) and the German Research Foundation (DFG) within the Collaborative Research Centre “On-The-Fly Computing” (SFB 901).

REFERENCES

- [1] ETSI, “NFV Management and Orchestration; Report on Architectural Options,” Website, 2016, online at http://www.etsi.org/deliver/etsi_gs/NFV-IFA/001_099/009/01.01.01_60/gs_NFV-IFA009v010101p.pdf.
- [2] SONATA consortium, “D4.1: Orchestrator prototype,” Website, 2016, online at <http://www.sonata-nfv.eu/content/d41-orchestrator-prototype>.
- [3] ETSI OSM, “Open Source MANO: Open Source NFV Management and Orchestration (MANO) software stack aligned with ETSI NFV,” Website, 2016, online at <https://osm.etsi.org>.
- [4] Fraunhofer Fokus, “OpenBaton,” Website, 2015, online at <http://openbaton.github.io/>.
- [5] H. Karl, S. Dräxler, M. Peuster, A. Galis, M. Bredel, A. Ramos, J. Martrat, M. S. Siddiqui, S. van Rossem, W. Tavernier *et al.*, “DevOps for network function virtualisation: an architectural approach,” *Transactions on Emerging Telecommunications Technologies*, vol. 27, no. 9, pp. 1206–1215, 2016.
- [6] M. Peuster, H. Karl, and S. van Rossem, “MeDICINE: Rapid Prototyping of Production-Ready Network Services in Multi-PoP Environments,” in *Network Function Virtualization and Software Defined Network (IEEE NFV-SDN)*. IEEE, 2016.
- [7] T. Lévai, I. Pelle, F. Németh, and A. Gulyás, “EPOXIDE: a modular prototype for SDN troubleshooting,” in *ACM SIGCOMM Computer Communication Review*, vol. 45, no. 4. ACM, 2015, pp. 359–360.
- [8] S. Clayman, L. Mamatas, and A. Galis, “Experimenting with control operations in software-defined infrastructures,” in *NetSoft Conference and Workshops (NetSoft)*. IEEE, 2016, pp. 390–396.

⁵YouTube Video: <https://youtu.be/8X2lpAbeLvM>

⁶GitHub: <https://github.com/sonata-nfv/son-emu>