

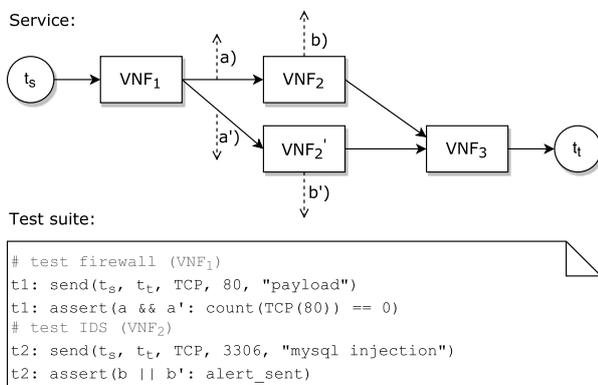
BACHELOR/MASTER THESIS

Towards DevOps for NFV: Automated Testing of Network Service Chains

Background

Prototyping and testing single virtualized network functions (VNFs), like firewalls or intrusion detection systems (IDS), as well as complex network service chains which consist of multiple VNFs is still a challenging task with a lot of manual steps. For this purpose, our group develops an NFV emulation platform to support rapid prototyping of single VNFs as well as of entire network services [2, 3]. This platform allows network service developers to execute their services locally before they are moved to production (e.g. before they are deployed in a public cloud environment). However, our platform and the corresponding toolchain do not yet support automated testing of network services nor do they provide solutions to define such tests.

The idea is to design a unit test-like framework that is able to test network services and network functions by sending test traffic through them and observe their outputs, e.g., check if a firewall drops the right flows or if a IDS fires the correct alerts. To automate this, a programming model for such *network function tests* as well as a execution framework is needed. Similar systems for software defined networking (SDN) use cases are already available but they lack support for generic VNFs and test automation [1]. The following figure shows the general idea and how a test programming model could look like. Each test case specifies which kind of traffic is sent through the service and checks the resulting behavior at different points in the service chain.



Thesis Goals

The goal of this thesis is to design and specify a programming model to implement test cases for network service chains and single network functions. The model needs to be flexible and generic so that network service developers can specify tests independently from the programming languages in which the VNFs are implemented or the network protocols they support. After this, a framework and execution environment for the designed test specifications have to be implemented and their usefulness has to be validated with a set of use cases based on example services.

Milestones

- Search for existing literature and testing approaches
- Design programming model for network test cases
- Implement framework to support test development
- Build execution environment (e.g. based on [3])
- Validate your solution with a set of examples

Required knowledge (or willing to learn)

- Linux and Linux networking
- Networking protocols
- Good programming skills (especially Python)

References

- [1] I. Pelle, T. Lévai, F. Németh, and A. Gulyás. One tool to rule them all: A modular troubleshooting framework for sdn (and other) networks. In *Proceedings of the 1st ACM SIGCOMM Symposium on Software Defined Networking Research*, page 24. ACM, 2015.
- [2] M. Peuster. Containernet on GitHub. Online at <https://github.com/containernet/containernet>.
- [3] M. Peuster, H. Karl, and S. van Rossem. Medicine: Rapid prototyping of production-ready network services in multi-pop environments. *arXiv preprint arXiv:1606.05995*, 2016.