

SDN-based SDK for DC Networks & Service Function Chaining Use Case

Irena Trajkovska

traj@zhaw.ch

Networking in DCs - Yet another abstraction layer?



Networking in DCs - actuals

- Reinvent networking for containers vs yet another abstraction layer to Neutron?
 - **Kuryr**: Docker libnetwork support to Neutron API
 - **OVN**: Augments OVS with logical switches and routers, security groups, and L2/L3/L4 ACLs, implemented on top of a tunnel-based (VXLAN, NVGRE, Geneve, STT, IPsec) overlay network
 - **Dragonflow** will support use cases of nested containers inside a VM without the need to introduce another layer of overlay abstraction (also integrate with Kuryr)
- Dockers native networking has seen light
- Single SDN solution for OpenStack VMs and Docker containers?

Sources:

<http://superuser.openstack.org/articles/project-kuryr-brings-container-networking-to-openstack-neutron>

<http://galsagie.github.io/sdn/openstack/ovs/2015/04/26/ovn-containers/>

<http://blog.docker.com/2015/11/docker-multi-host-networking-ga/>

http://superuser.openstack.org/articles/what-you-need-to-know-about-dragonflow-for-the-openstack-liberty-release?utm_campaign=OSN&utm_content=23742038&utm_medium=social&utm_source=twitter

Motivation and Objective

- **Problem**

- Datacenter cloud networking bases on protocols that were not initially designed for it
- Tunneling protocols generate inherent cascading and encapsulation especially in multi tenant systems
- Complexity leads to systems that are prone to errors, energy inefficient with increased difficulty to configure and maintain
- Vendor specific configuration requirements and heterogeneous architectures

- **Technology enablers – SDN**

- Reduce complexity by unifying and centralizing network configuration; easier programmability of networking systems

- **Goal:** Define competitive use cases and sophisticated solutions towards optimized SDN design for modern cloud architectures



Netfloc Strategy

- Network functions should be pluggable and chainable
- Reduce the complexity and protocol overhead in cloud networks
- Project a global view of the network for easier analysis
- Increase the traffic efficiency

Technical Innovation

- **NETwork FLOws for Clouds (Netfloc):** Datacenter network programming tools and libraries packed as SDK for network programmers
 - Based on OpenDaylight controller for SDN-enabled end-to-end management of OpenFlow enabled switches
 - Improves existing networking functions and leverage development of novel networking applications
- **Netfloc features**
 - Rest based Netfloc application as ODL module (ODL dynamically loads Netfloc applications at runtime)
 - APIs and programmable interfaces to retrieve information from the OpenDaylight datastore repository and OpenStack Neutron
 - Establishes per tenant specific network graph for end-to-end flow control between OpenStack VMs
 - Uses **connection based** abstractions as: **Network Graph, Network Path, Flow Pattern,..**
 - Completely SDN and OpenFlow based – enables to create abstractions of the underlying network and provide alternative libraries and flow patterns to the encapsulation protocols used in OpenStack environment

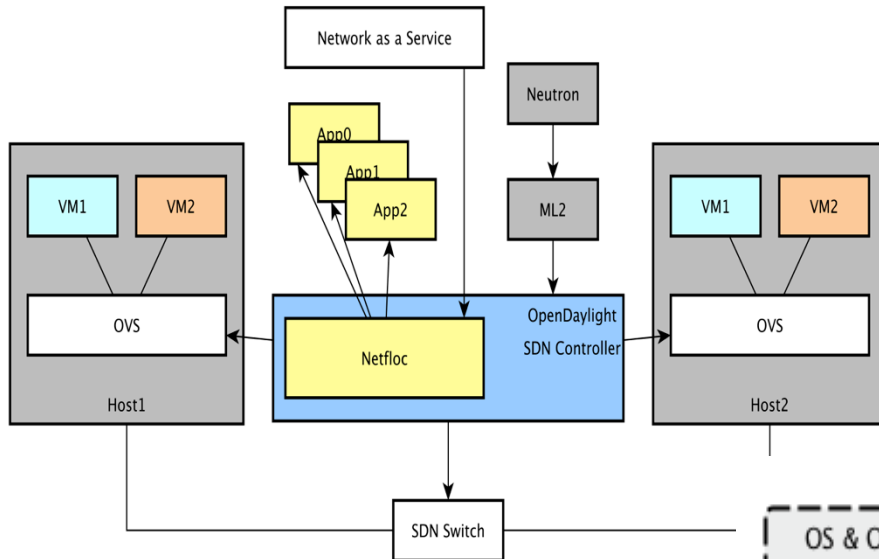
Application Driven Development

- *Isolated network slices and connectivity services*
 - Ensure tenant segregation using novel non-GRE/VxLAN tunneling mechanism for optimized packet header
- *Resilience*
 - Provide direct SDN data plane control & on-demand switch provision and configuration
- *Service Function Chaining*
 - Traffic steering mechanism within ODL
- *Monitoring, administration & QoS support*
 - Provide OpenFlow based datacenter monitoring and QoS network adaptation
-

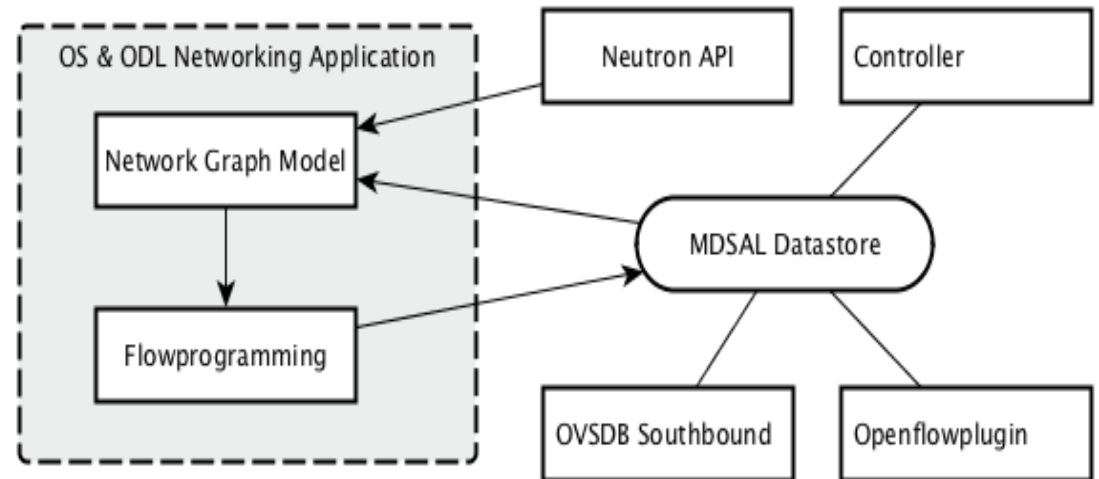


SDN Application within Netfloc

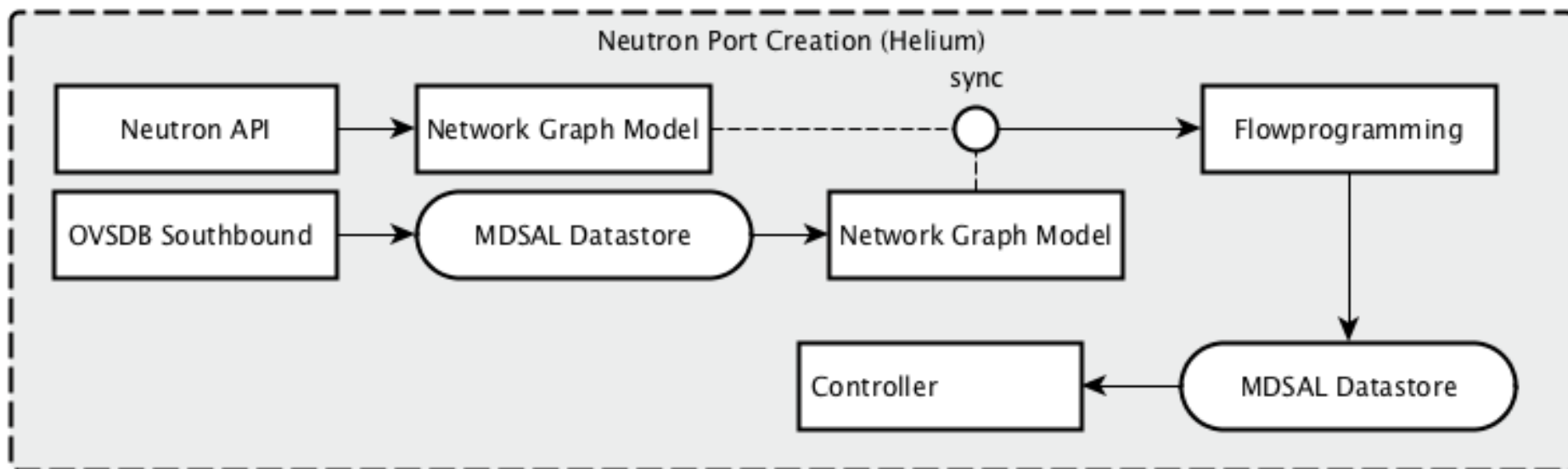
Technical Architecture



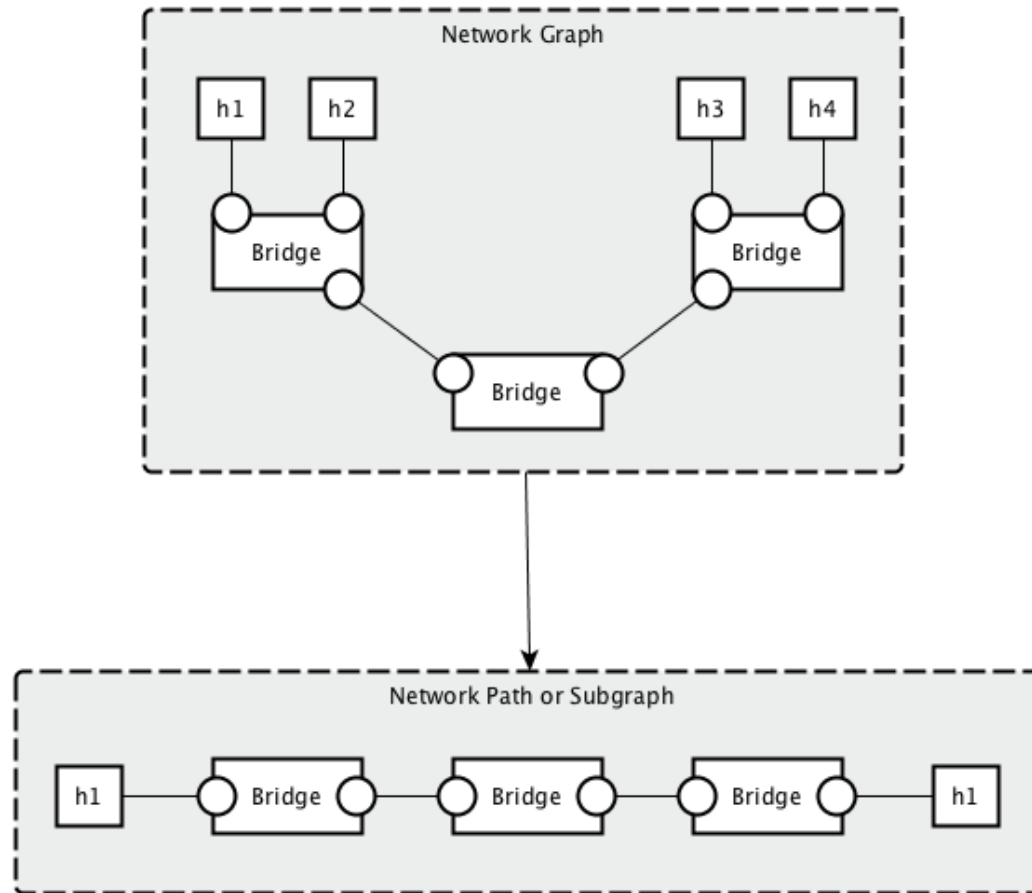
Netfloc application



Example: Neutron Port Creation



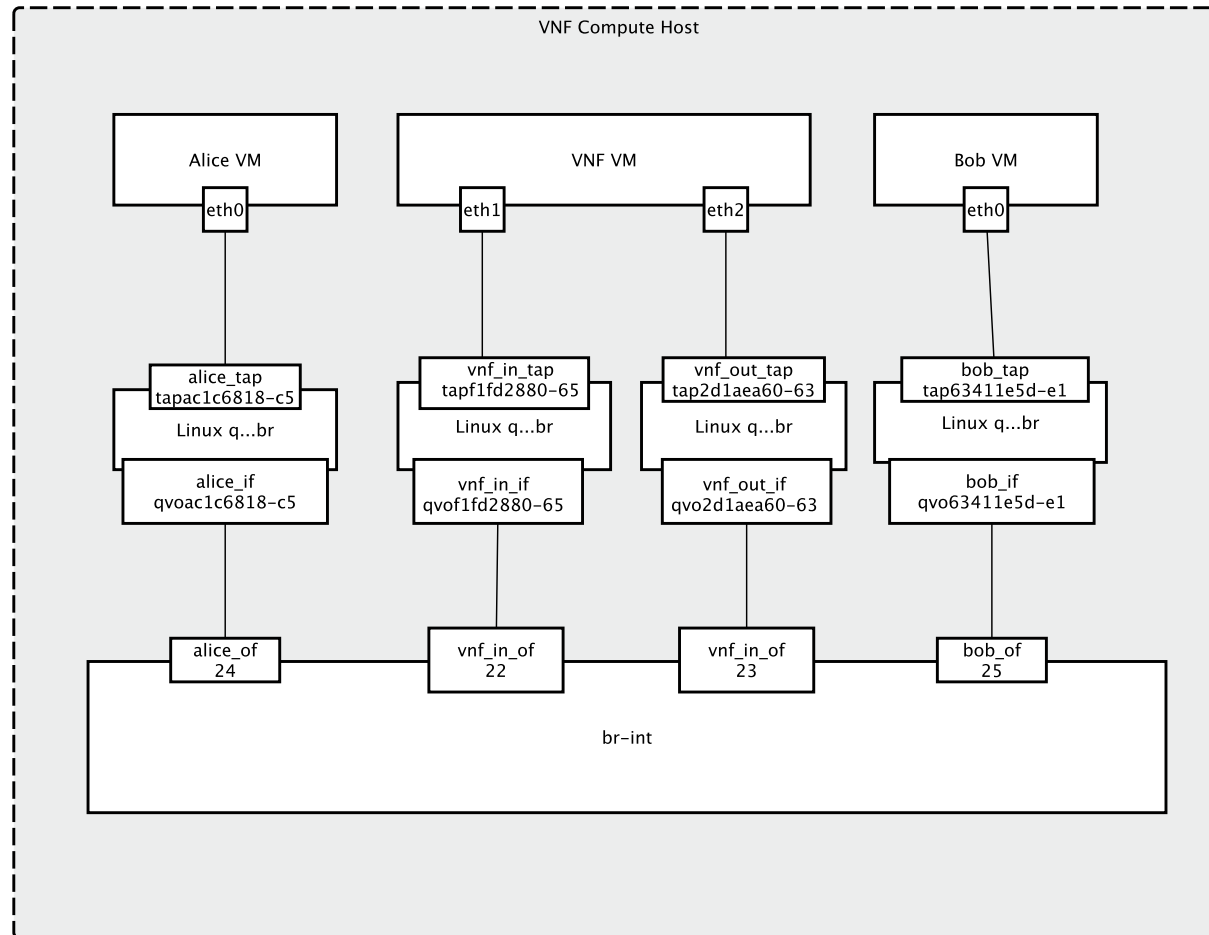
Network Graph Model



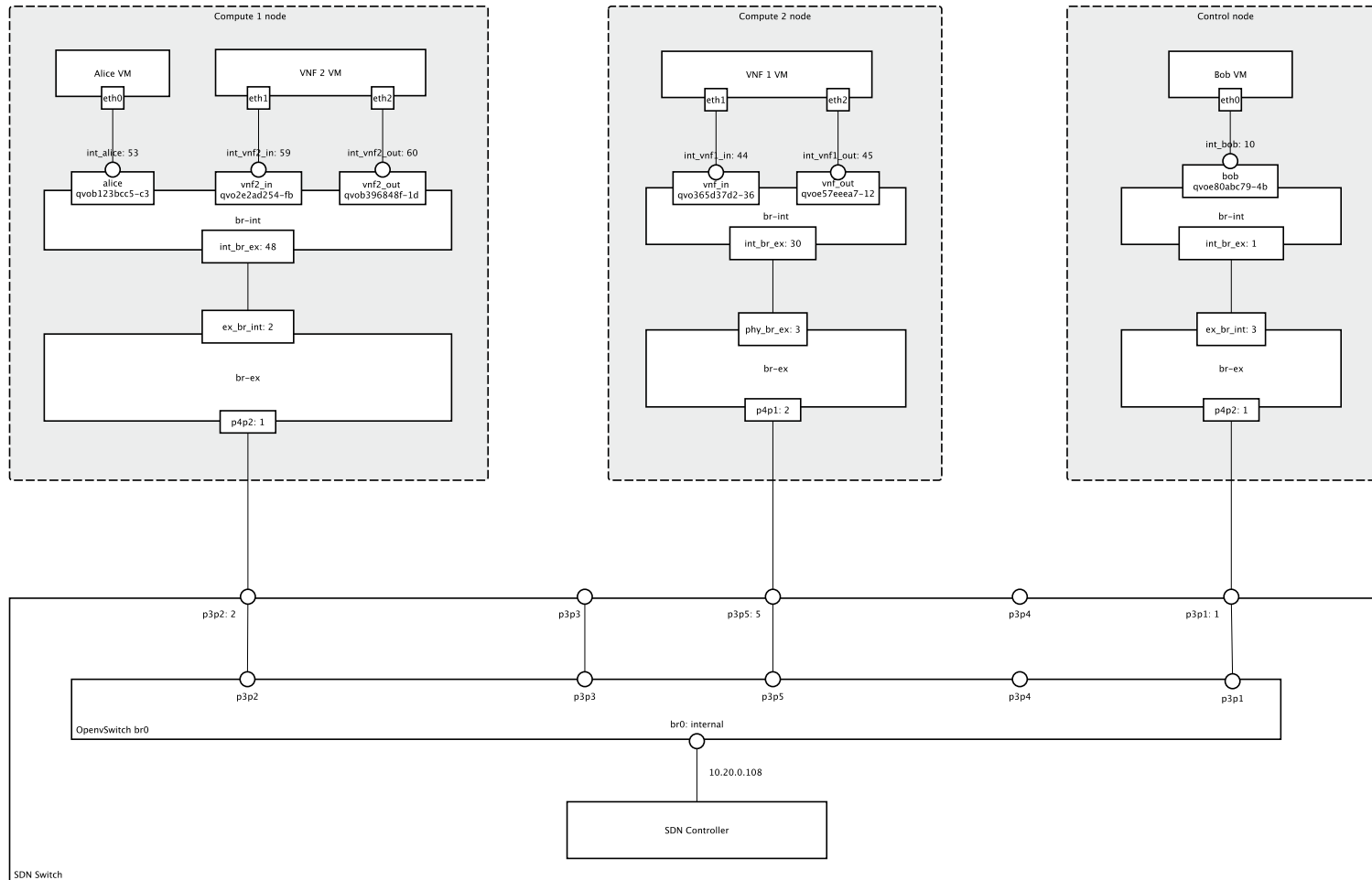
Some T-Nova VNFs

- vTU
 - Open-source, modular, customizable; contains encoding/decoding plug-ins
 - Convert video streams from one video format to another
 - Transcoding (mono-directional: video streaming; bi-directional: videoconferencing applications)
- vTC
 - Traffic analysis (pps), add tagging rules (change packet ToS)
 - Http request to apply rule IDs and ToS onto endpoint for the classifier
 - nDPI searches within the payload of the packet to find ex. “facebook” content
- vProxy
 - Bandwidth limitations, web caching, url filtering (parental control), system user anonymity
 - Dashboard which can be used by a user in order to configure Squid for his LAN in a high lever way

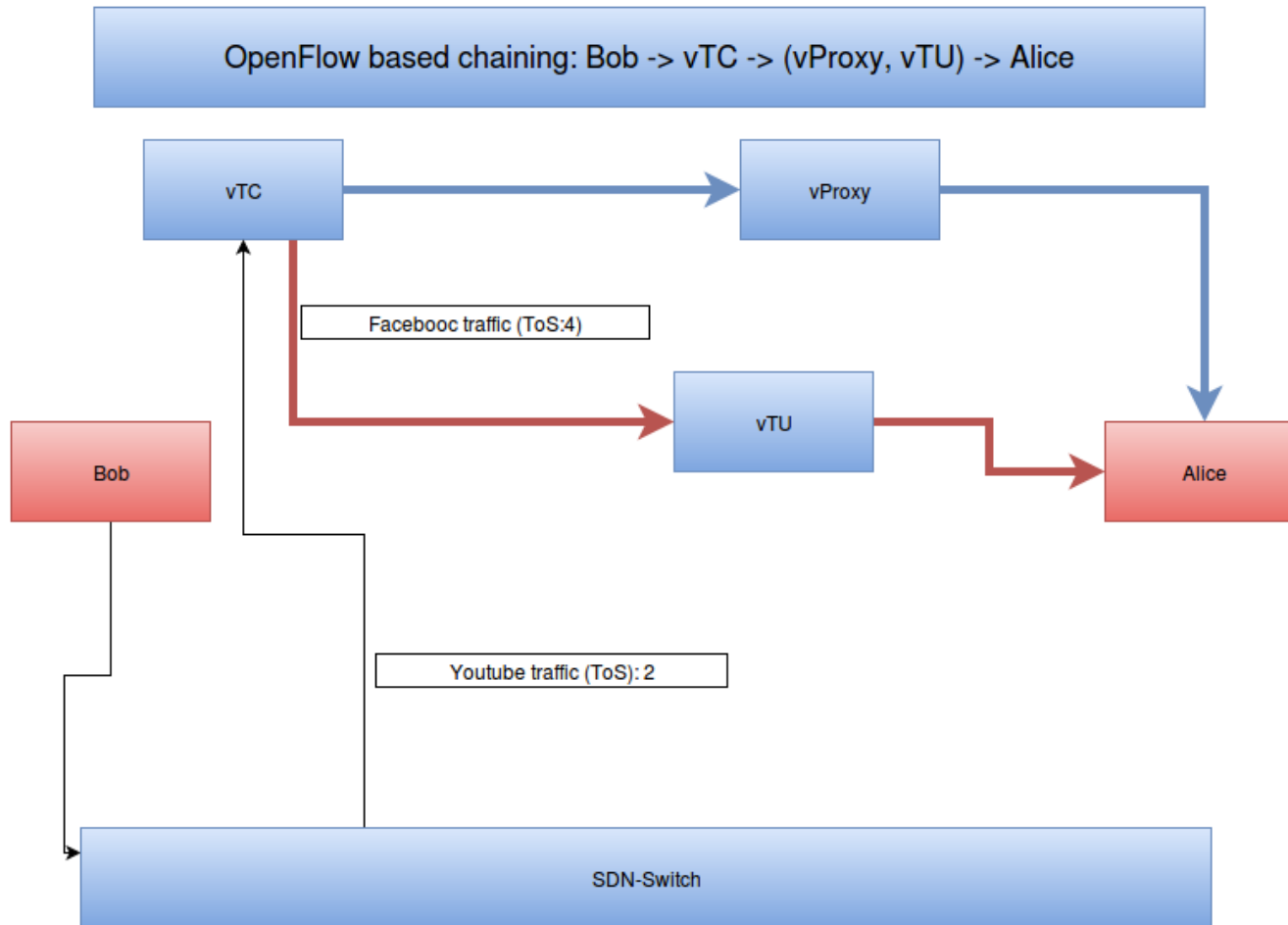
Single host chaining



Multi VNFs chaining (packet analysis)



Multi VNFs chaining (nDPI)



Netfloc SFC API Draft

- **CHAIN PATH**
 - A chain path is a logical subsequent connection between specific neutron ports.
 - *[port_A, sfc_1_in, sfc_1_out, sfc_2_in, sfc_2_out, port_B]*
 - Each of the listed items would be neutron ports.
 - The SDK caches all the created ports so they could be referenced by their ID.
- **CHAIN FLOW PATTERN**
 - A flow pattern for SFC chains would match on the specific structure of a chain path and can be applied onto one.
 - Needed in scenario where the chain is determined by a classifier, which is altering the packet structure in order to tag the traffic for different chains.

Measurements

- **I: GRE & non-GRE in SDN environment**
 - **Testbeds:** ICCLab and Demokritos
 - **Method:** (1) OpenStack GRE encapsulation & (2) Install flows on physical switch and OVSs
 - **Measurements:** latency, throughput, packet loss, jitter
 - **Setup:** Repeat measurements ex. 500 times for ICMP, TCP, UDP, and different MTU size
 - **Objective:** (1) Compare results from the two methods (GRE & non-GRE) in terms of performance achievements & (2) Compare results between two testbed setups for (1)
- **II: Service Function Chaining using T-Nova VNFs (vTC, vProxy)**
 - **Testbeds:** ICCLab and Demokritos
 - **Method:** (1) Install flows on physical switch and OVSs (2) Rewrite IP/mac addresses on OVSs
 - **Measurements:** latency, throughput, packet loss, jitter
 - **Setup:** Repeat measurements ex. 500 times for ICMP, TCP, UDP, and different MTU size
 - **Objective:** (1) Compare results in two different chaining methods: (1a) OpenFlow (1b) Mac rewriting in terms of network performance & (2) Compare results between two testbed setups for (1)

ICCLab SDN Impact and Community

- Projects:
 - **FIWARE** - A Core Platform of the Future Internet
 - **T-NOVA** - Network Functions-as-a-Service over virtualized Infrastructures
 - **SESAME** - Small cEIS coordinAtion for Multi-tenancy and Edge services
- Events: SDN biannual workshop: ICCLab and SWITCH co-organized
- Open Cloud Day
<http://www.ch-open.ch/fr/events/aktuelle-events/160615-open-cloud-day-2015/>



References

- **SDN Initiative info:**
 - <http://blog.zhaw.ch/icclab/category/research-approach/themes/software-defined-networking-for-clouds/>
- **Netfloc on github:** <http://icclab.github.io/netfloc/>
- **Netfloc architecture:** <http://netfloc.readthedocs.org/en/latest/architecture/>
- **SDN ICCLab blog posts:**
 - <http://blog.zhaw.ch/icclab/a-design-draft-for-tenant-isolation-without-tunneling-in-openstack/#more-8559>
 - <http://blog.zhaw.ch/icclab/provisioning-openstack-with-foreman-and-packstack/>
- **SDN meetup / workshop:**
 - Upcoming:
<http://www.meetup.com/SDN-Switzerland/events/225417314/>
 - Last: <http://blog.zhaw.ch/icclab/4th-sdn-workshop-in-bern/>

Demo

...

