Container-based Network Function Virtualization for Software-Defined Networks

Richard Cziva, Simon Jouet, Kyle J. S. White and Dimitrios P. Pezaros

University of Glasgow, United Kingdom

r.cziva.1@research.glasgow.ac.uk

Middleboxes

- Hardware-based network appliances to manipulate network traffic
  - Firewall
  - Load balancer
  - VPN
  - Intrusion Detection and Prevention Systems
  - WAN Accelerator
  - Web cache
- Enterprise networks rely on middleboxes
  - Middleboxes represent **45% of the network devices**
  - The advent of customer devices will further increase the number

Problems with middleboxes

- They incur significant capital investment
- They are cumbersome to maintain
- They can not be extended to run new functionality
- The proprietary software on which they run, limits innovation and creates vendor lock-in

Network Function Virtualization

- Decouples network functions from the hosting platform
  - Can reduce capital and operational expenditure
  - Improve resource efficiency
  - Introduce fault-tolerance and scalability
- Works well with Software-Defined Networking

Issues

- Current approach to NFV rely on static infrastructure
- Static routing: update of every switch routing table to redirect traffic
- Operator specific implementation for their own environment
- Poor reuse of software components
  - Deploy and configure once for a specific server
  - Operator specific deployment system(s)
- Inability to create/destroy network functions quickly
  - Inserting routing rules and deploying + configuring software is complex
  - Costly operation

State of art

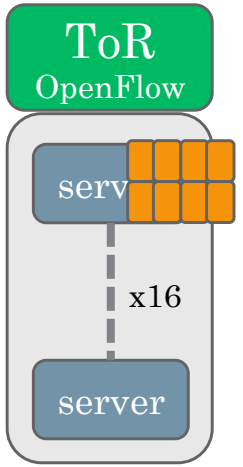
- OpenStack: early stage demos for NFV
- OPNFV: Linux foundation project, first release “Arno” this April
- Cloud4NFV: VM-based NFV orchestration for private clouds
- ClickOS: a custom, high-performance XEN-based VM
- “Stateless network functions”
- …

Glasgow Network Functions (GLANF)

- Main characteristics:
  1. Container-based
  2. Transparent
  3. Infrastructure independent
  4. Open innovation
- Two key contributions are
  1. Using **containers** for NFs
  2. End-to-end transparent traffic management (using **SDN**)

SDN and Containers

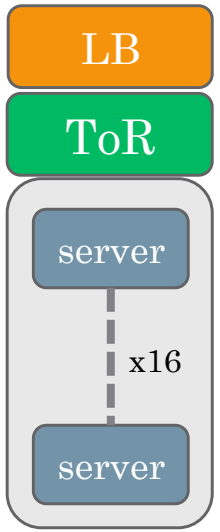
- Software Defined Networking:
  - Already deployed in DCs
  - Central control over the network
  - Can insert/remove new rules quickly from a single software
  - Open specifications / API
  - We used *OpenFlow* for our prototype
- Container Based Virtualization
  - Lightweight virtualization
  - Fast create/start/stop/delete
  - High performance
    - Small delay, High throughput, Low memory footprint
  - Reusable / Shareable
  - Traditional software environment
  - We used *Docker* containers for the prototype



Network Function Virtualization

- Middleboxes
  - Expensive ☑
  - Provisioned for peak demand ☒
  - Complex to configure and maintain ☑ ☒
  - Proprietary software → Vendor lock-in ☑

- Virtualize the network functions (NFV)
  - Decouple hardware function from hardware platform
  - Use commodity x86 hardware
    - Cheap, well known platform
  - Can deploy existing, vendor independent software
    - BRO, Snort, Iptables, OpenvSwitch …

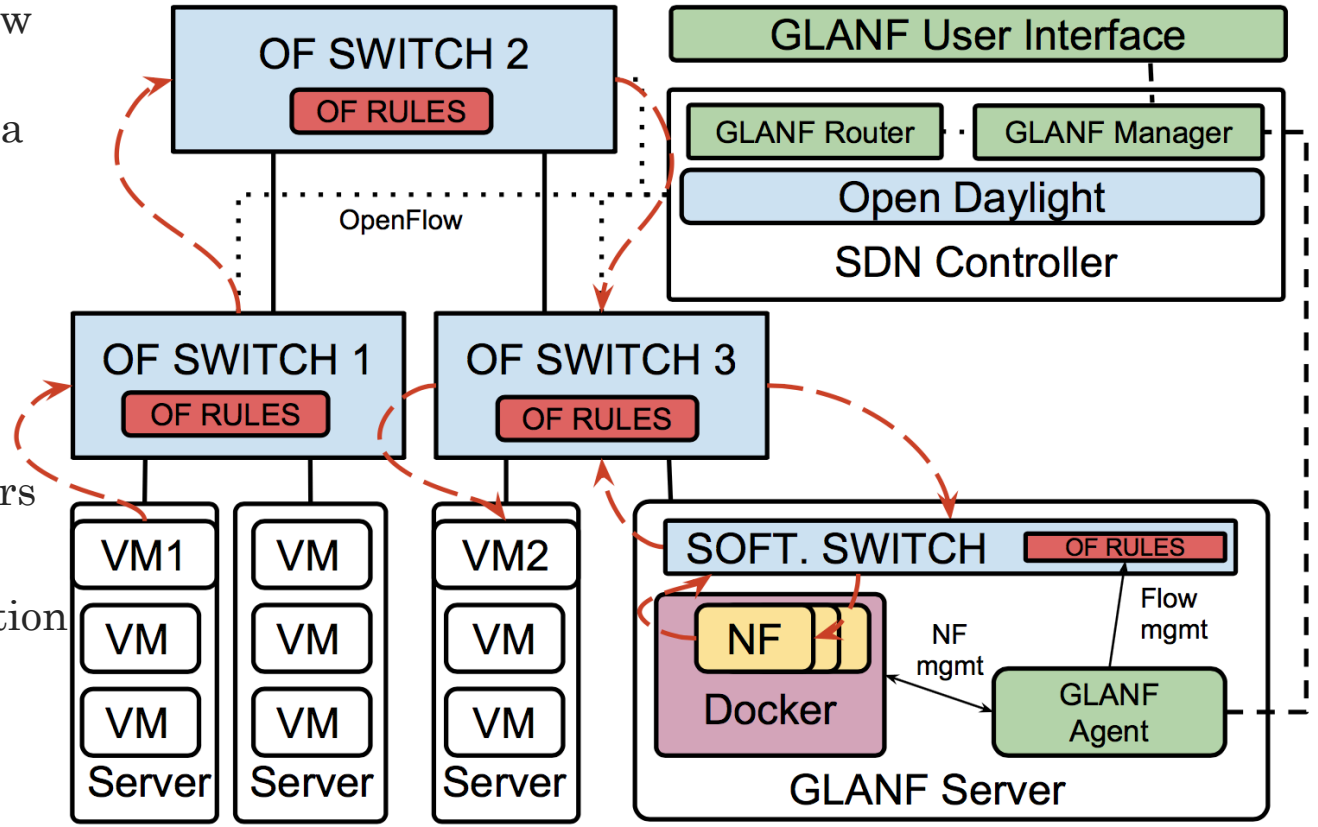


GLANF

- Framework for Network Function Virtualization
  - Infrastructure independent
    - Can be deployed in any OpenFlow environment
    - Doesn't require any specific hypervisor technology
  - Fast and Simple Deployment
    - Select a network function image
    - Select a host/subnet to enforce specific policy
    - Automatic
      - Selection of hosting server
      - Deployment of network function
      - Insertion of routing rules
  - Open innovation
    - Public/Private repository to share image
    - Open source implementation

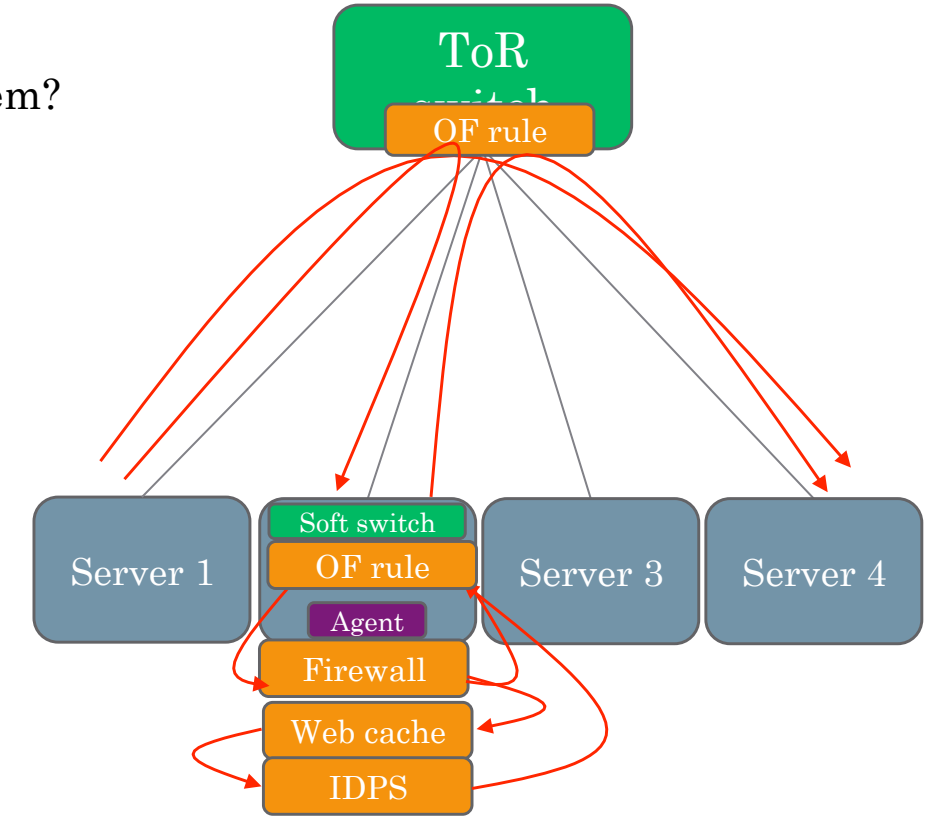
GLANF design

- Router
  - Hosted on the Open Daylight OpenFlow Controller
  - Creates and inserts the rules to apply a specific forwarding policy
- Manager
  - Provides a REST API to the system
- Agent
  - Daemon running on the GLANF servers
  - Manages containers and local routing
  - Provide host/container status information
- UI
  - Talks to the Manager
  - Adds/removes network functions

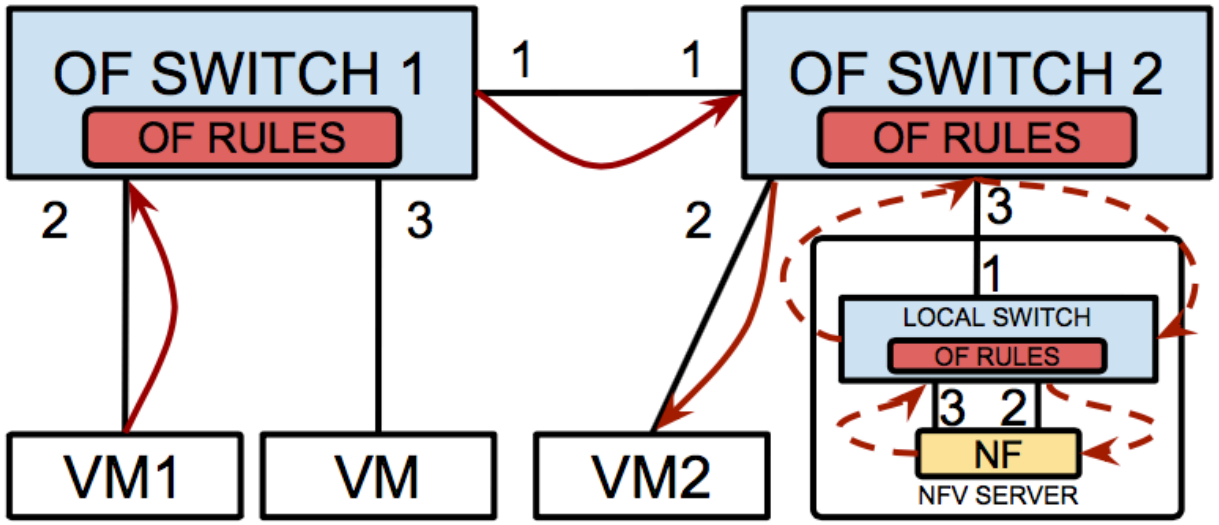


Traffic management step-by-step

1. Traffic from Server1 to Server4
2. Need a new Firewall placed between them?
   - Controller find a GLANF server
   - Pull the firewall image
   - Spawn an instance
3. Apply the policy
   - Reroute the traffic matching:
     - FROM server1
     - TO server2
4. Chaining Containers
   - Web Cache
   - IDPS

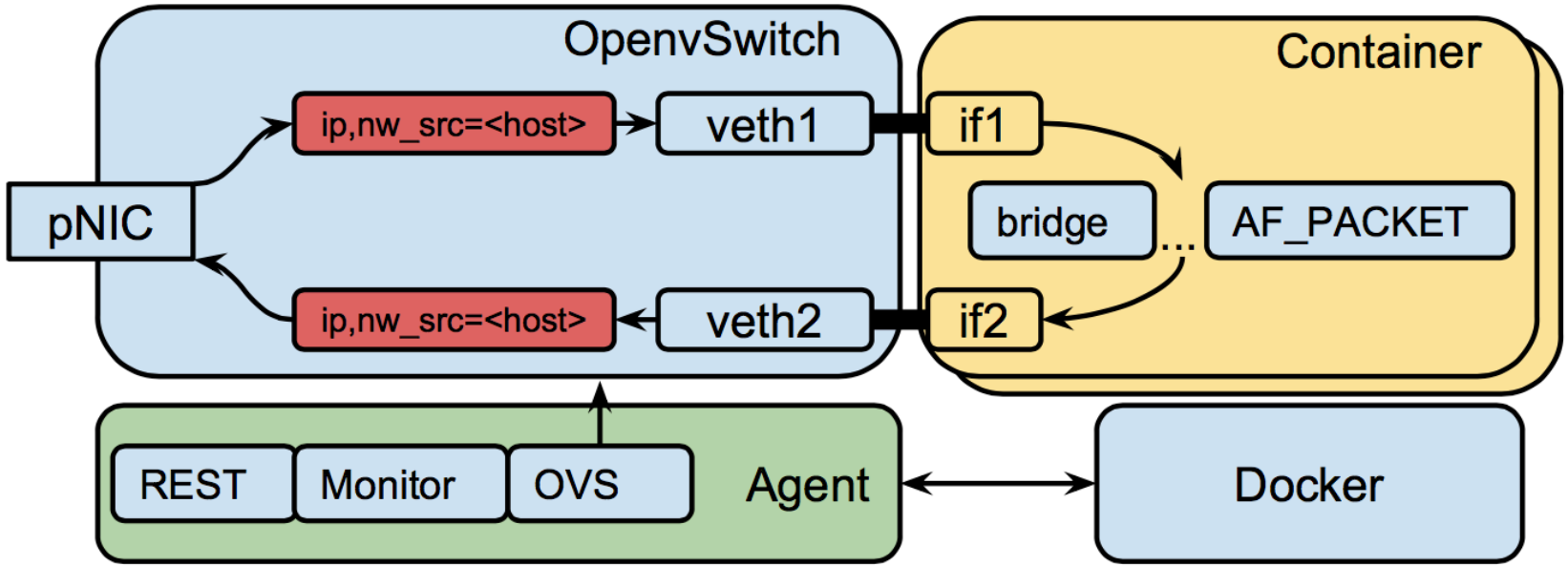


Transparent redirection



Switch	Match	Action
1	input_port: 2, src_ip: VM1	output_port: 1
2	input_port: 1, src_ip: VM1	output_port: 3
local	input_port: 1, src_ip: VM1	output_port: 2
local	input_port: 3, src_ip: VM1	output_port: 1

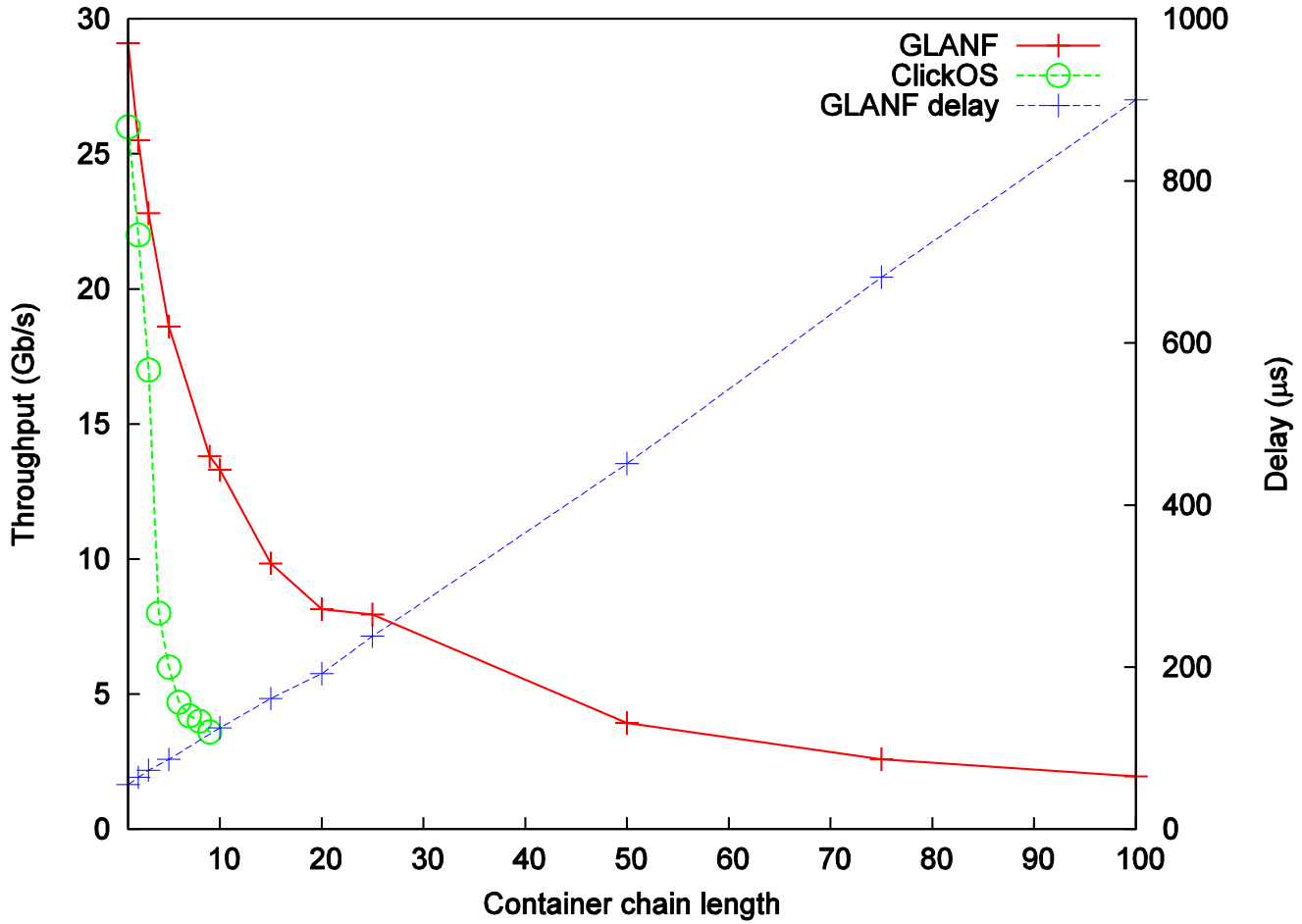
Inside the GLANF Server



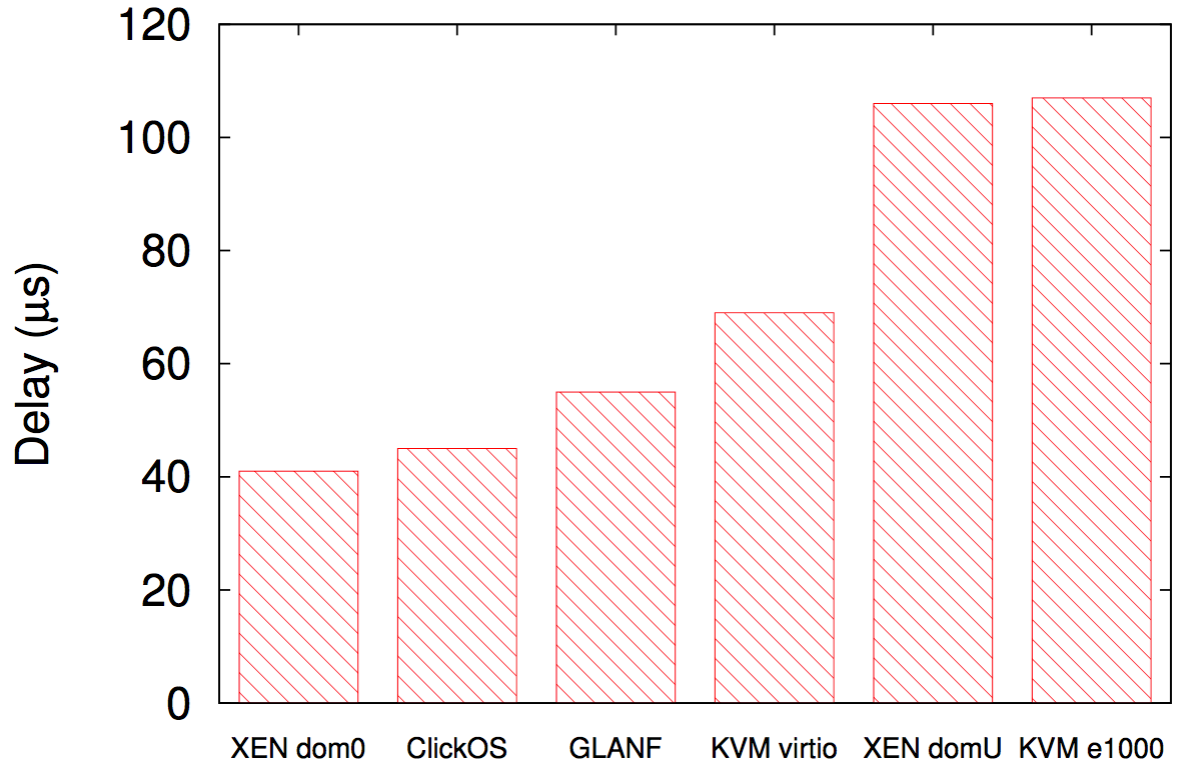
Performance evaluation

1. Throughput
2. Performance through NF chains
3. Idle ping delay
4. Start and stop time of the NFs

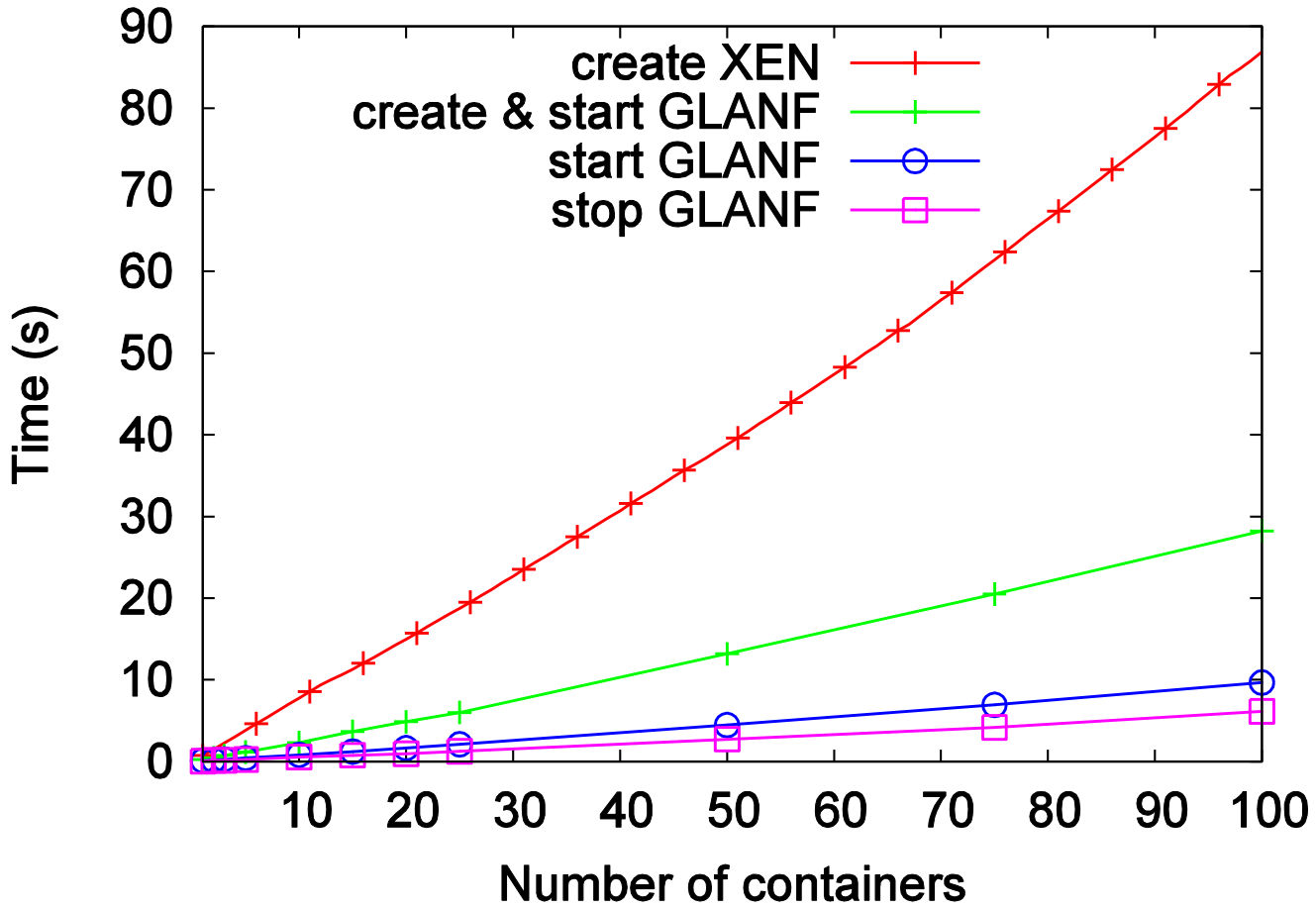
Throughput and delay



Idle ping delay



Start & stop of containers

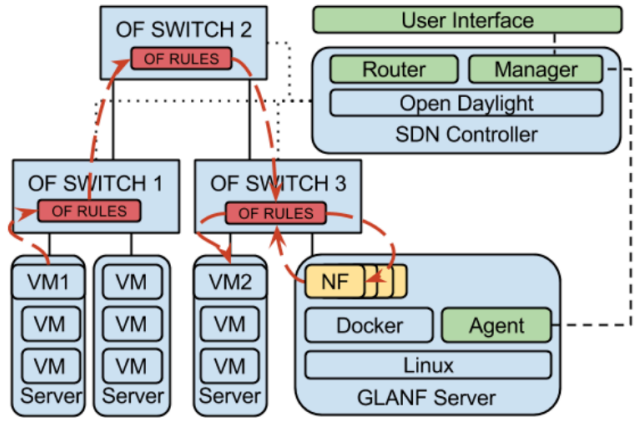


Home Network Functions Documentation

Glasgow Network Function

GLANF (Glasgow Network Function) is an open Network Function Virtualization (NFV) framework for OpenFlow-enabled infrastructures based on Docker containers.

Features



Container-based: Network functions are packaged in light-weight Docker containers to provide fast instantiation time, platform-independence, high throughput and low resource utilisation for the system.

Transparent: Hosts need not to change their traffic's destination to use network functions, as re-routing the traffic is handled entirely by the network without modifying packet headers.

Infrastructure independent: Traffic routing for NFs is handled separately from the DCs generic routing policies, allowing forwarding of traffic from any host to ephemeral NFs in

OpenFlow-enabled environments

http://glanf.dcs.gla.ac.uk

Thank you for your attention!

Container-Based Network Function Virtualization for Software-Defined Networks

Richard Cziva, Simon Jouet, Kyle J. S. White and Dimitrios P. Pezaros
School of Computing Science, University of Glasgow, Glasgow, G12 8QQ, Scotland
{r.cziva.1, s.jouet.1, k.white.3}@research.gla.ac.uk, dimitrios.pezaros@glasgow.ac.uk
http://glanf.dcs.gla.ac.uk

***Abstract*—Today's enterprise networks almost ubiquitously deploy middlebox services to improve in-network security and performance. Although virtualization of middleboxes attracts a significant attention, studies show that such implementations are still proprietary and deployed in a static manner at the boundaries of organisations, hindering open innovation.**

In this paper, we present an open framework to create, deploy and manage virtual Network Functions (NF)s in OpenFlow-enabled networks. We exploit container-based NFs to achieve low performance overhead, fast deployment and high reusability missing from today's NFV deployments. Through an SDN northbound API, NFs can be instantiated, traffic can be steered through the desired policy chain and applications can raise notifications. We demonstrate the systems operation through the development of exemplar NFs from common Operating System utility binaries, and we show that container-based NFV improves function instantiation time by up to 68% over existing hypervisor-based alternatives, and scales to one hundred chained NFs while incurring sub-millisecond latency.

I. Introduction

Enterprise networks rely on a wide spectrum of hardware-based network appliances or 'middleboxes' to transform, inspect, filter or otherwise manipulate network traffic on top of packet forwarding. In recent years, middleboxes have become

current NFV deployments by large ISPs and DC network operators suffer from the statically-configured underlying routing mechanisms in place, which do not support open interfaces and result in operator and environment-specific solutions in static or semi-static environments [5]. For example, deploying one or more network functions requires the update of all affected switches' routing tables to redirect traffic, therefore making it impractical to deploy infrastructure-wide NFs. Consequently, NFV systems exhibit poor component reuse, and are still unable to fulfill dynamic, temporal traffic workloads in an elastic manner [6] [7]. In such environments, there is no cross-layer information exchange between the routing layer and the network functions, which results in a limited view of the network to each functional entity. We argue that improvements in NFV can be achieved by synergistic management and optimisation of NFs and end-to-end routing between hosts and NFs.

At the same time, Software-Defined Networking (SDN) has emerged to logically centralise the network's control plane with OpenFlow as the main SDN protocol implementation [8] [9]. SDN is penetrating in highly dynamic environments such as Cloud Data Centers (DCs), mainly