ROMA TRE
UNIVERSITÀ DEGLI STUDI

NOMS

Kathará

A Container-Based Framework for Implementing Network Function Virtualization and Software Defined Networks

G. BONOFIGLIO, V. IOVINELLA, G. LOSPOTO, G. DI BATTISTA

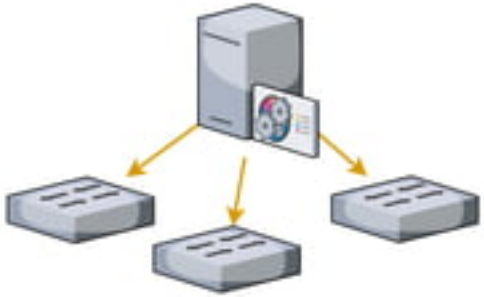
Introduction

- Developed by the Computer Networks research group of the Engineering Department at Roma Tre University

- http://kathara.org/
- http://www.dia.uniroma3.it/~compunet/www/view/group.php?id=compunet

Context

Introducing software at different levels:



SDN

NFV

Programmable
Data-plane

Network Function Virtualization (NFV)

Traditional NFs are physically bound to middleboxes:

- High costs for updates, repair and substitution
- Cannot keep up with real-time traffic and increasing demand

Virtual NF:

- Decoupling NFs from the specific-purpose hardware
- Software VNFs on general-purpose hardware

Network Function Virtualization (NFV)

Pro:
more flexibility, lower costs

Con:
lower performance wrt the specific-purpose hardware

Data-plane programmability

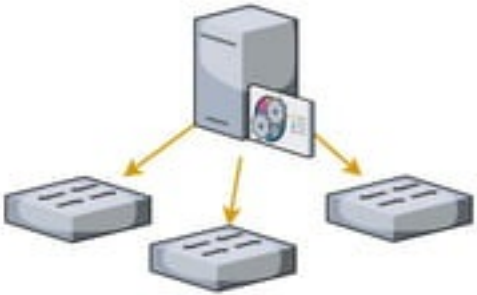
- It is possible to implement several NFs mostly by altering the forwarding plane
- This opens the possibility to implement flexible NFs on high performance hardware

P4 language

- Open-source project by P4 Consortium
- Leader on the market for programming protocol independent packet processing
- Can be compiled and executed on specific network equipment

- Barefoot Tofino (a P4 target) can process packets up to 6,5 Tb/s.

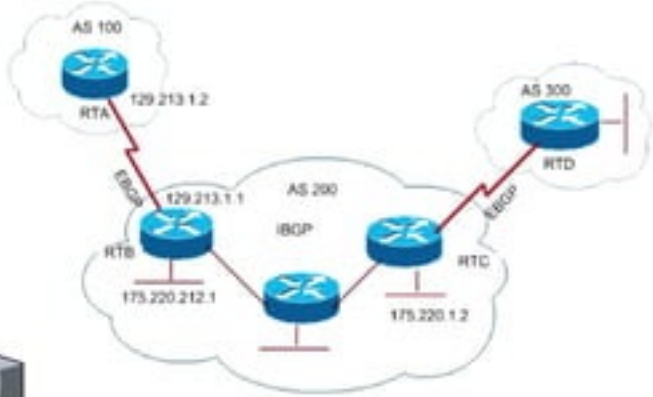
A complex environment



SDN

NFV

Programmable Switches

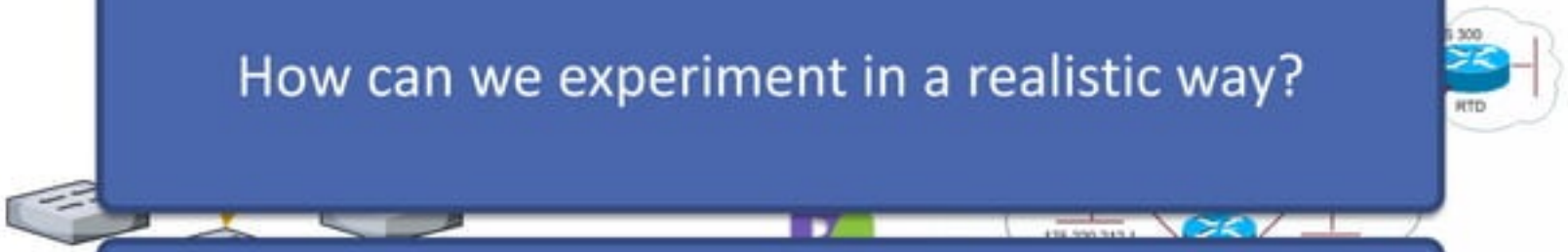


Traditional routing protocols

A complex environment

How can we experiment in a realistic way?

Can we gain benefits from virtualizing part of the network for production?



State of the art

Netkit

M. Pizzonia et al,
"Netkit: easy emulation of complex networks on inexpensive hardware", 2008.
H. Mostafaei at al,
"Sdnetkit: A testbed for experimenting sdn in multi-domain networks", 2017.

- *"The poor man's system to experiment computer networking"*
- Developed by **Compunet Lab Roma Tre**
- Supports SDN and traditional routing protocols
- Based on VM

- Kathará supports P4 and can be extended
- Based on containers

Netkit & SDNetkit

ROMA TRE
UNIVERSITÀ DEGLI STUDI

Docker Compose

- Comes natively with Docker
- Focuses on services
- Limited interaction with networking capabilities

Moreover, Docker itself is not made to implement fully fledged networks.

Developing Kathará, we faced and solved several issues to configure the networking properly.

ClickOS

J. Martins et al, "Clickos and the art of network function virtualization", 2014.

- Focuses only on virtual middleboxes (NFV)
- Based on VM

- Kathará offers advanced routing functionalities
- Based on containers

ClickOS

GNF

R. Cziva et al, "Container-based network function virtualization for software-defined networks", 2015.

- Deployment of pre-built VNFs in SDN networks
- Based on containers

- Kathará is agnostic with respect to the underlying network architecture
- Offers the possibility to implement any custom VNF through data-plane programmable nodes

GNF

Kathará

Two main objectives

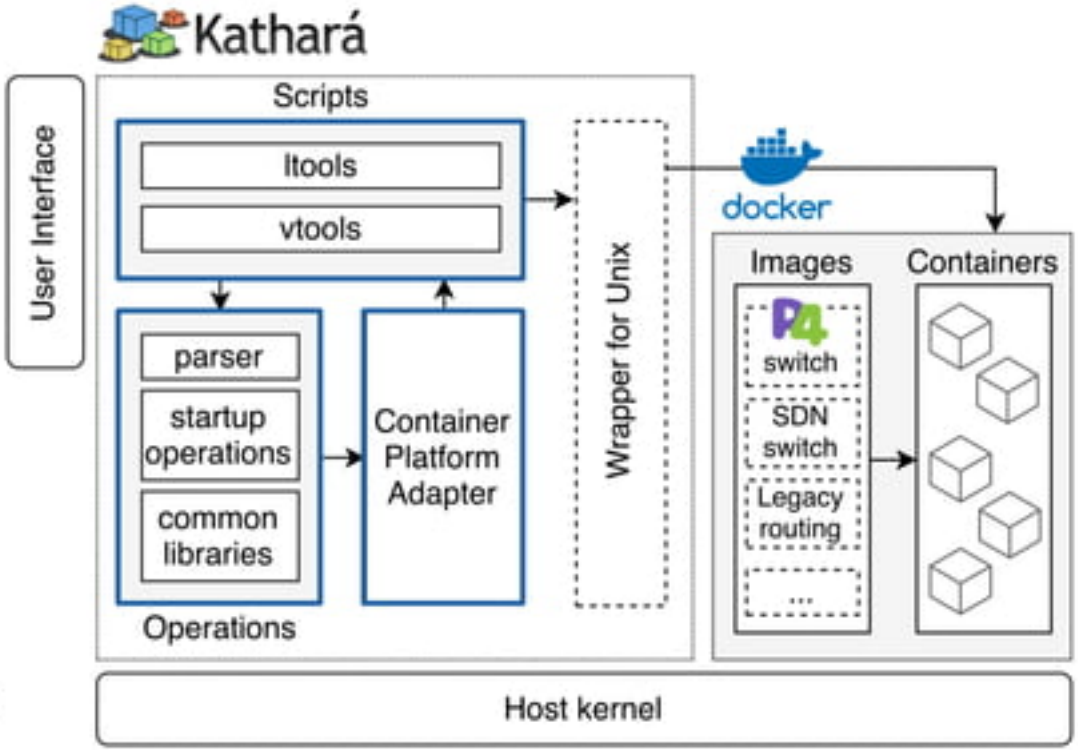
- Verify the possibility of implementing NFV through the P4 language
- Interact with SDN, NFV and P4 together with standard protocols to test network solutions, with very close approximation to real world scenarios

Kathará (Καθαρά)

- Framework based on Docker containers to create and manage virtual networks
- It comes with ready-to-pull images to implement SDN, data-programmable switches, standard routing protocols, DNS, web servers and more
- Can be easily extended through custom images
- Offers a simple command-line UI, inherited from Netkit
- Offers a very simple GUI

Kathará Architecture

- 3 main modules:
  - Scripts
  - Operations
  - Container Platform Adapter
- Pre-built images to implement what we need



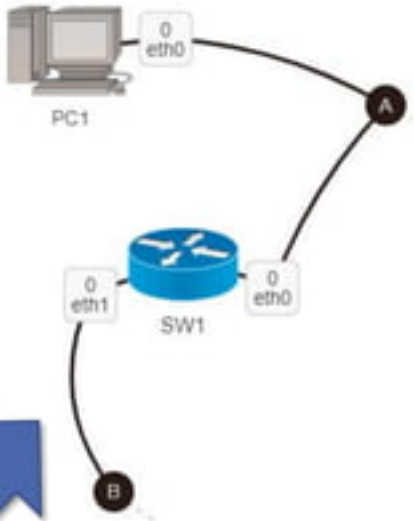
Kathará Images

Multi-platform technologies:

- **Open vSwitch**
  Software implementation of Open Flow enabled switch
- **Behavioral Model**
  Software implementation of P4 target switch
- **Quagga**
  Standard routing protocols suite (OSPF, BGP, RIP, etc)

vtools

- Commands for managing single network nodes

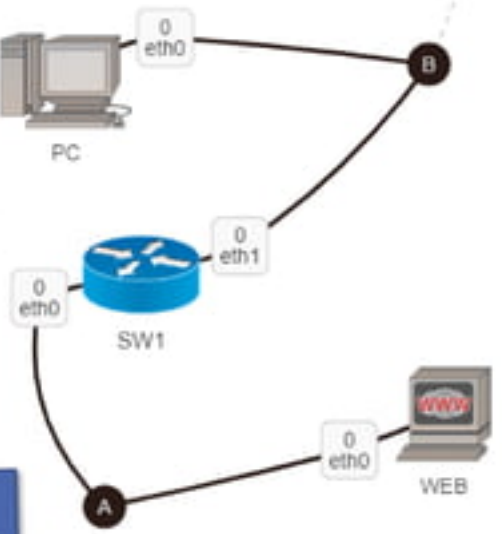


Example of vstart:

```
test@kathara:~$ vstart --eth 0:A PC1
test@kathara:~$ vstart --eth 0:A --eth 1:B --image=OVS SW1
```

ltools

- Commands for managing «*labs*»
- Based on configuration files for topology and startup ops

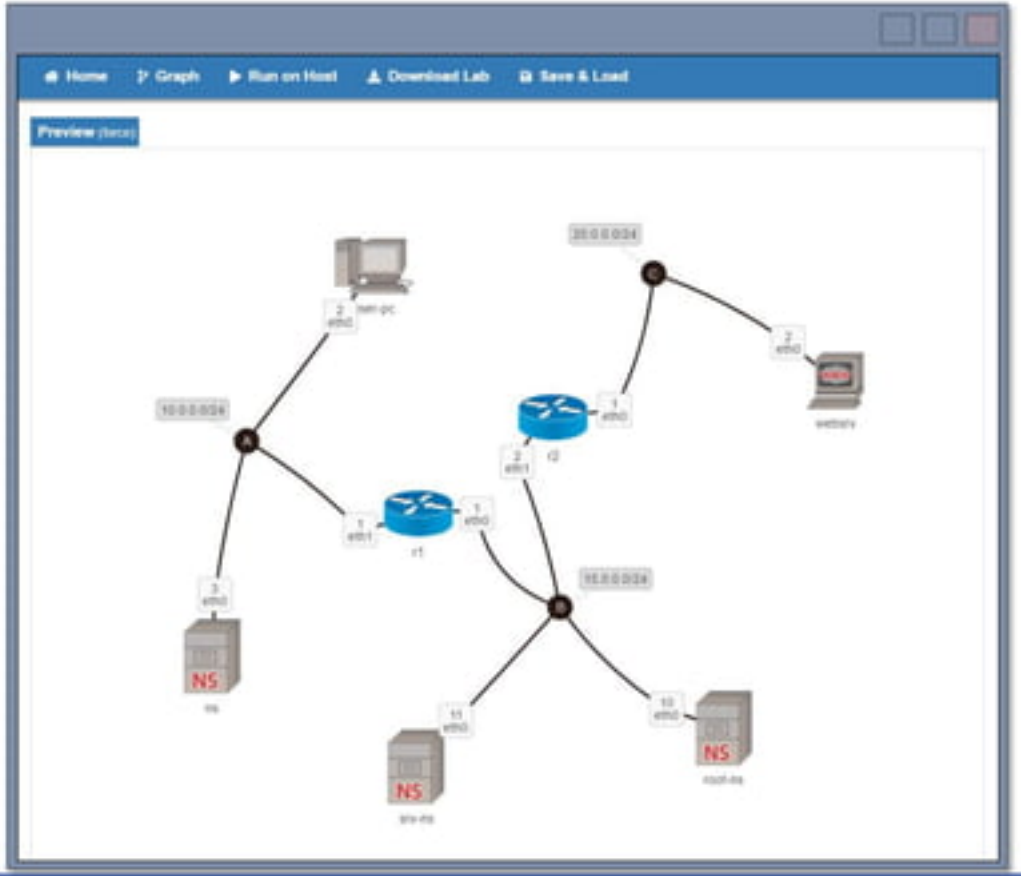


Example of lab.conf:

```
web[0]=A
sw1[0]=A
sw1[1]=B
sw1[image]=P4
pc[0]=B
```

GUI

- Can automatize basic and common operations
- Can show a preview of the network
- Can export a configuration file to be loaded later
- Works from the web or as a stand-alone executable



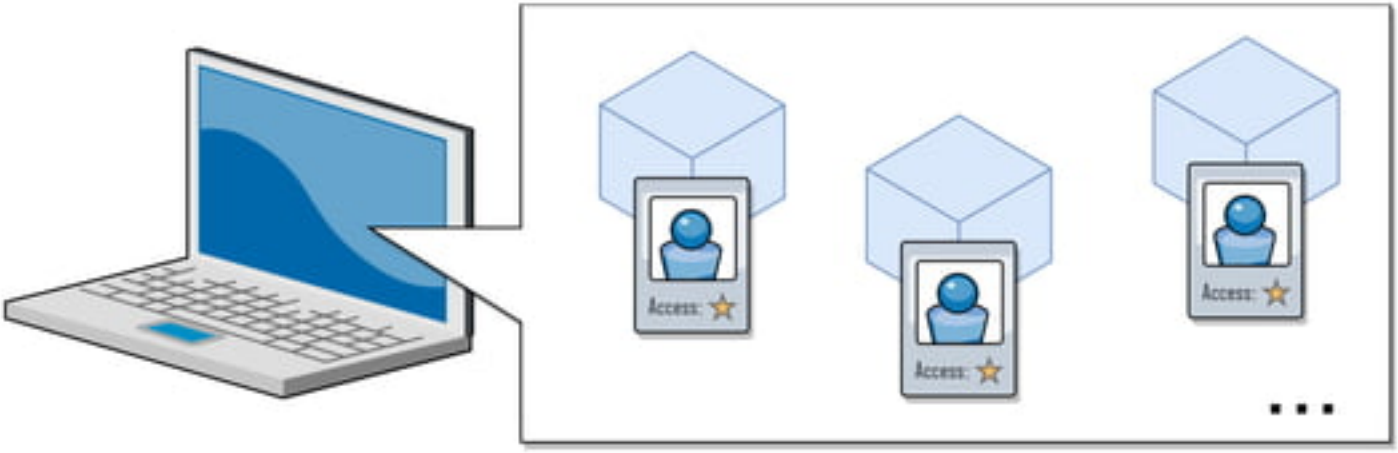
Security in Kathará

A different target from Docker

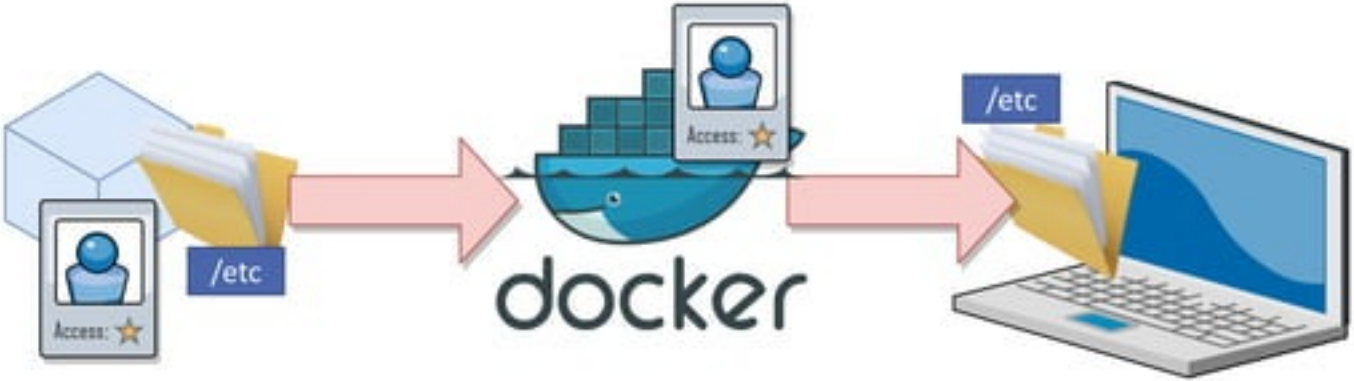
Problem:
Docker is a tool directed to system administrators, but Kathará is not.

Why is that and how can security issues be resolved?

Vulnerability



Vulnerability



Possible solutions

- Let only administrators to use Kathará
- Configure SUDO to accept only some command patterns

- **Create a wrapper**

Wrapper features

- Safe software executed with admin rights
- Middle layer between Kathará and Docker
- Only allows safe commands to be executed by Docker

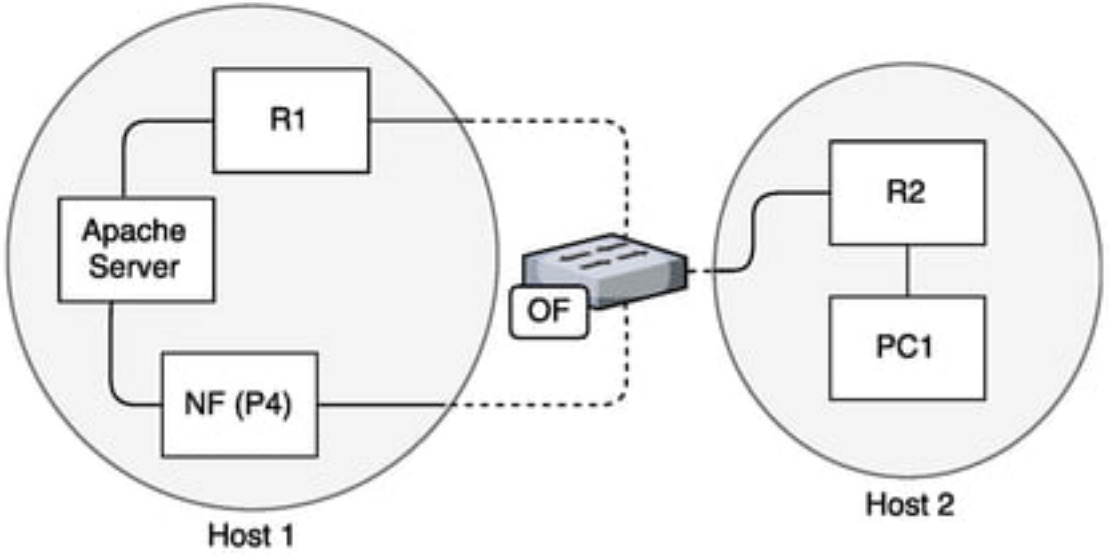
Use Cases

Use Case 1: Node transfer

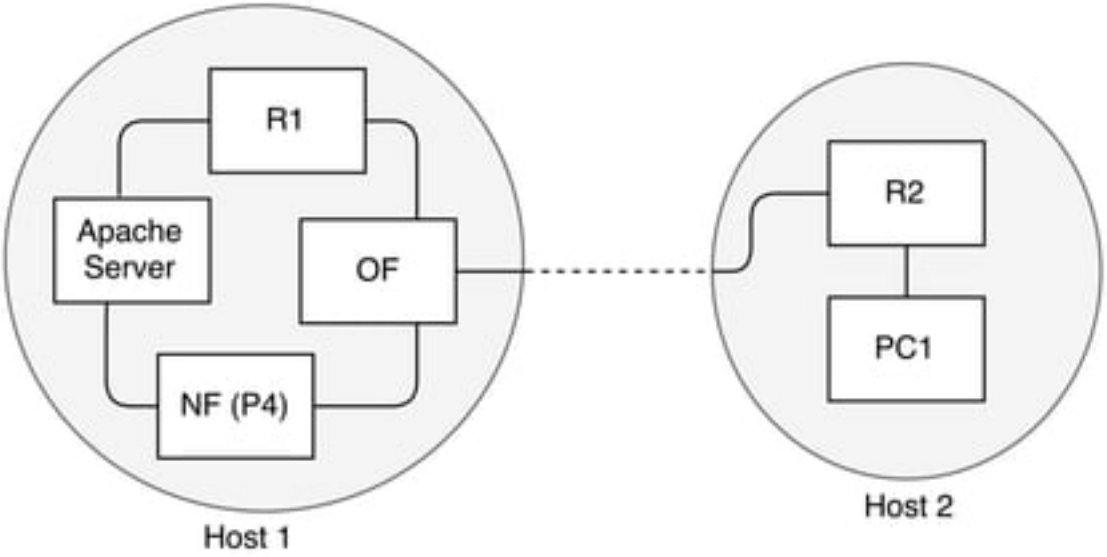
- Made possible by the *very close approximation* between software and hardware solutions
- Thanks to the container technology and the usage of multi-platform implemetations (OpenFlow, P4, BGP,…)

- The virtual nodes created with Kathará are thus operationally identical to physical network nodes

Use Case 1: Node transfer



Use Case 1: Node transfer

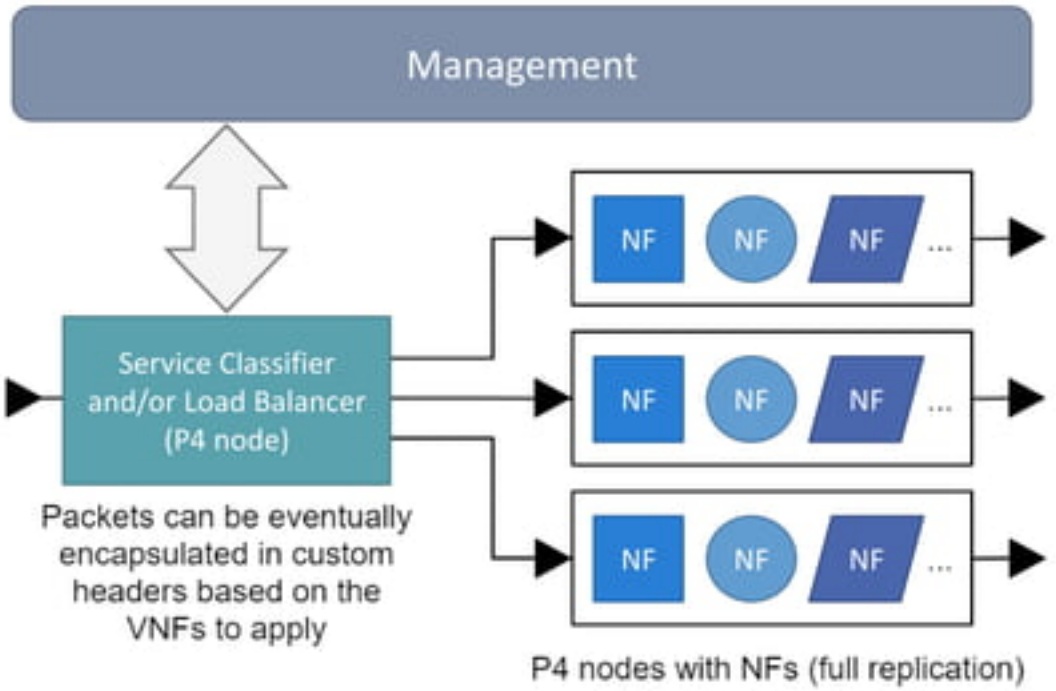


Use Case 2: NFV through P4

Goals:

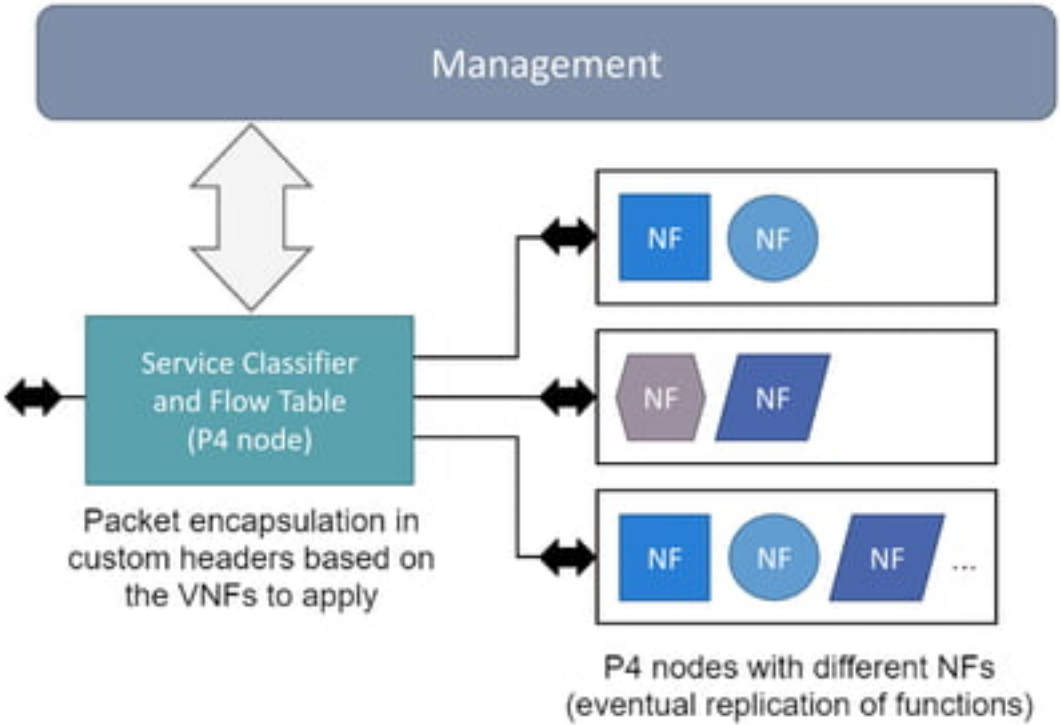
- **Pros from the NFV architecture: flexibilty, scalability, decoupling from harware**
- Gain in performance (wire speed)
- Simple programmability through a specific language
- Compliance to SFC (RFC-7665)

Architecture 1



Use Case 2: NFV through P4

Architecture 2



Evaluation

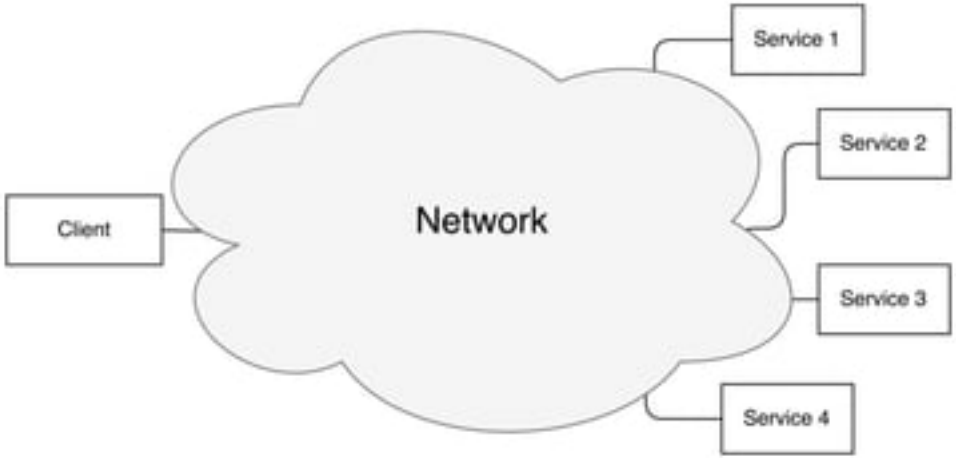
Evaluation goals

- To prove the effectiveness of Kathará in a production environment with respect to VMs
- To prove that Kathará can manage an increasing number of network nodes, even on low performance hardware

Evaluation testbed

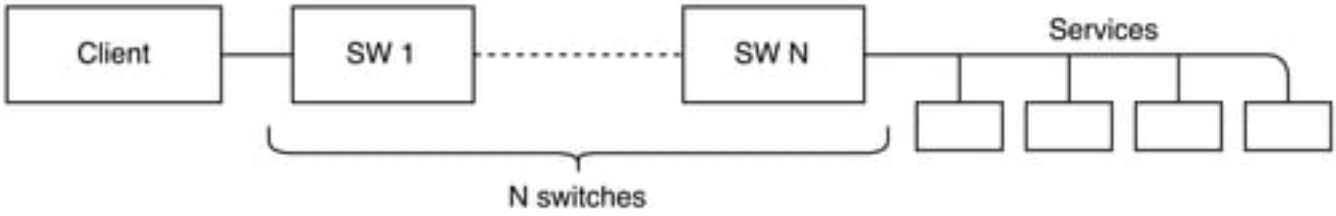
VM equipped with Ubuntu, 3 GBytes of RAM, 4 cores of 2.21 GHz each

A realistic web service based on Apache, PHP and MySQL

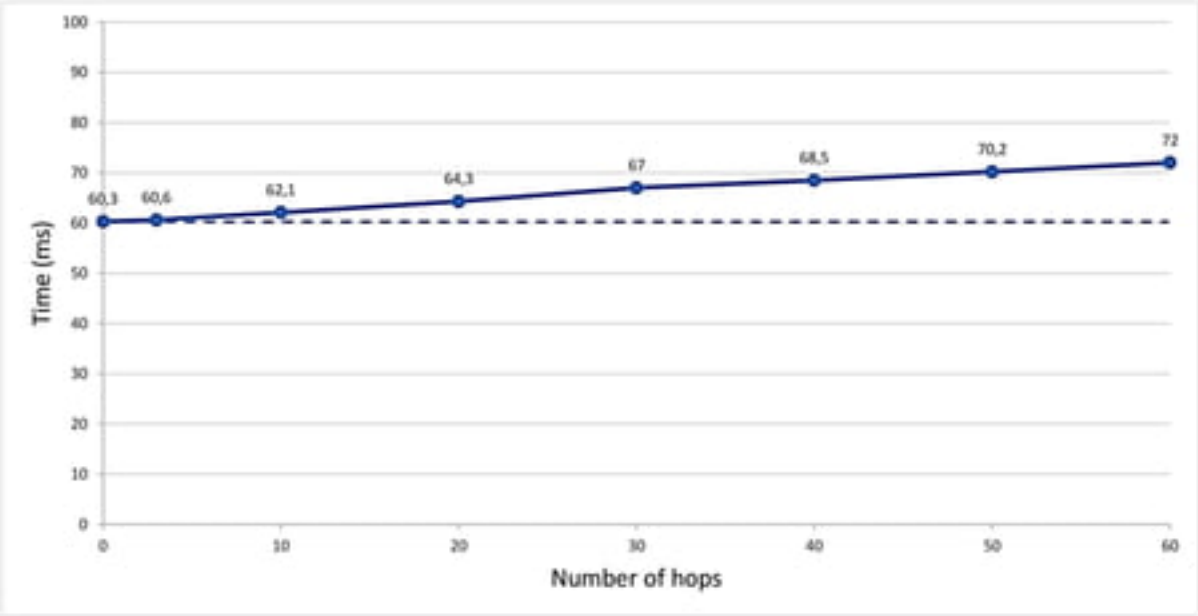


Evaluation testbed

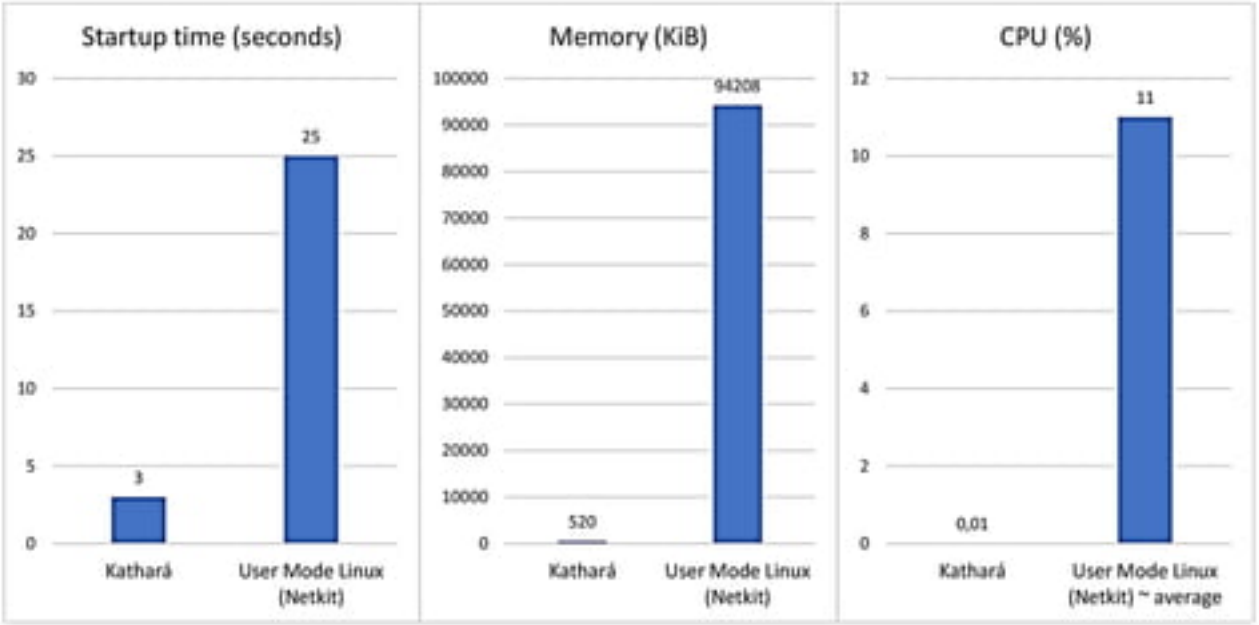
VM equipped with Ubuntu, 3 GBytes of RAM, 4 cores of 2.21 GHz each
A realistic web service based on Apache, PHP and MySQL



Evaluation results



Evaluation results



Evaluation results

- Under 40 network nodes running at the same time using UML (Virtual Machines)
- Over 300 network nodes running at the same time using Containers

(on a VM equipped with Ubuntu, 3 GBytes of RAM, 4 cores of 2.21 GHz each)

Conclusions and Future Works

Compatibility

- Kathará is fully compatible with any major OS

  - Linux
  - Windows
  - MacOS

Take away

- Kathará can implement any kind of network topology, enabling the usage of SDN, NFV and standard protocols together
- The application of standard and multi-platform technologies allows to transfer nodes from virtual to physical devices
- It offers higher performance with respect to VMs by several orders of magnitude

Future

- Kathará Inception
  - ❖ Kathará inside Kathará inside Kathará …
- Interaction with orchestrators for automatic cloud deployment
- New included images to implement new protocols

Thanks for your attention