Network Function Virtualization in GTS

Richard Cziva
University of Glasgow
NORDUnet

University of Glasgow

NETLAB
NETWORKED SYSTEMS RESEARCH LABORATORY

Network Function Virtualization

- Moving network services from hardware to software
  - Firewalls
  - Caches
  - Rate limiters
  - ..
- Trending topic between operators
  - and recently in research too

Why do we need NFV?

- We are facing increase in:
  - Number of connected clients
  - Amount of aggregate traffic
  - Needs for specialized services

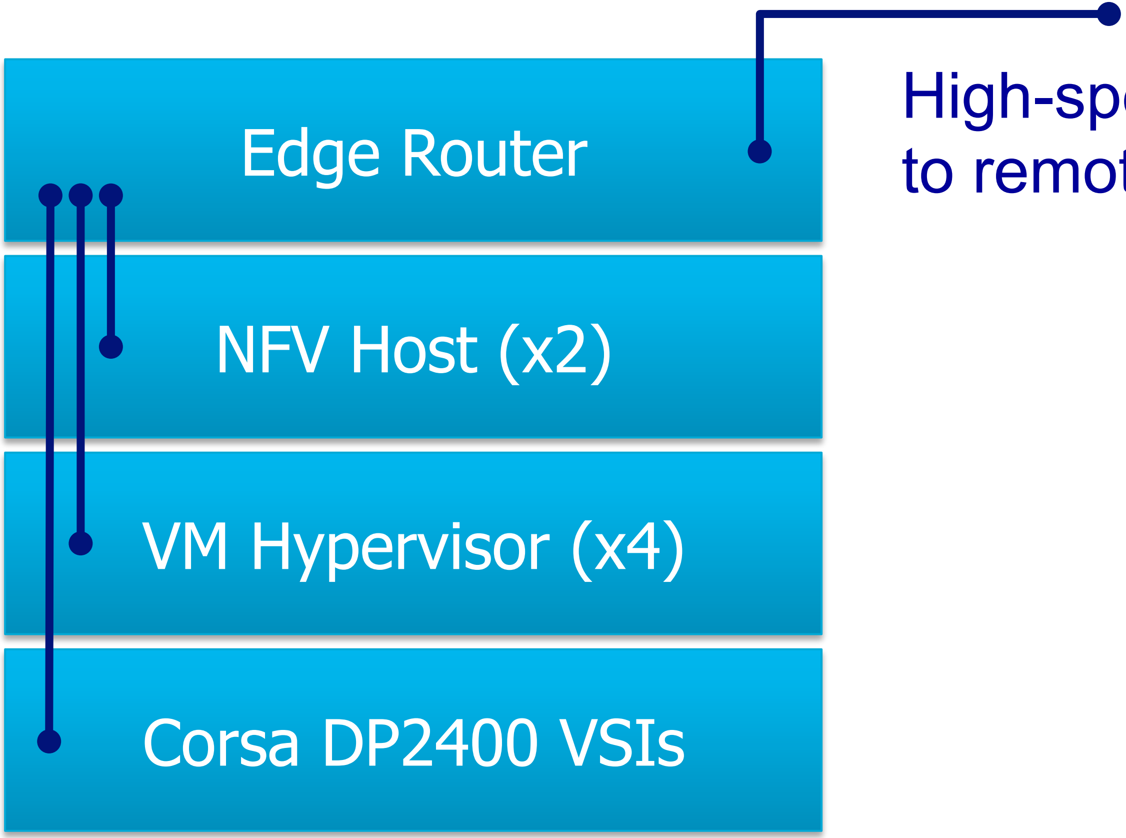
Traditional middlebox and network management can't cope with this in an efficient manner

- Virtualization has many benefits
  - CAPEX/OPEX savings
  - Increased automation, no human intervention
  - Brings agility and fault tolerance to the infrastructure

NFV in GTS

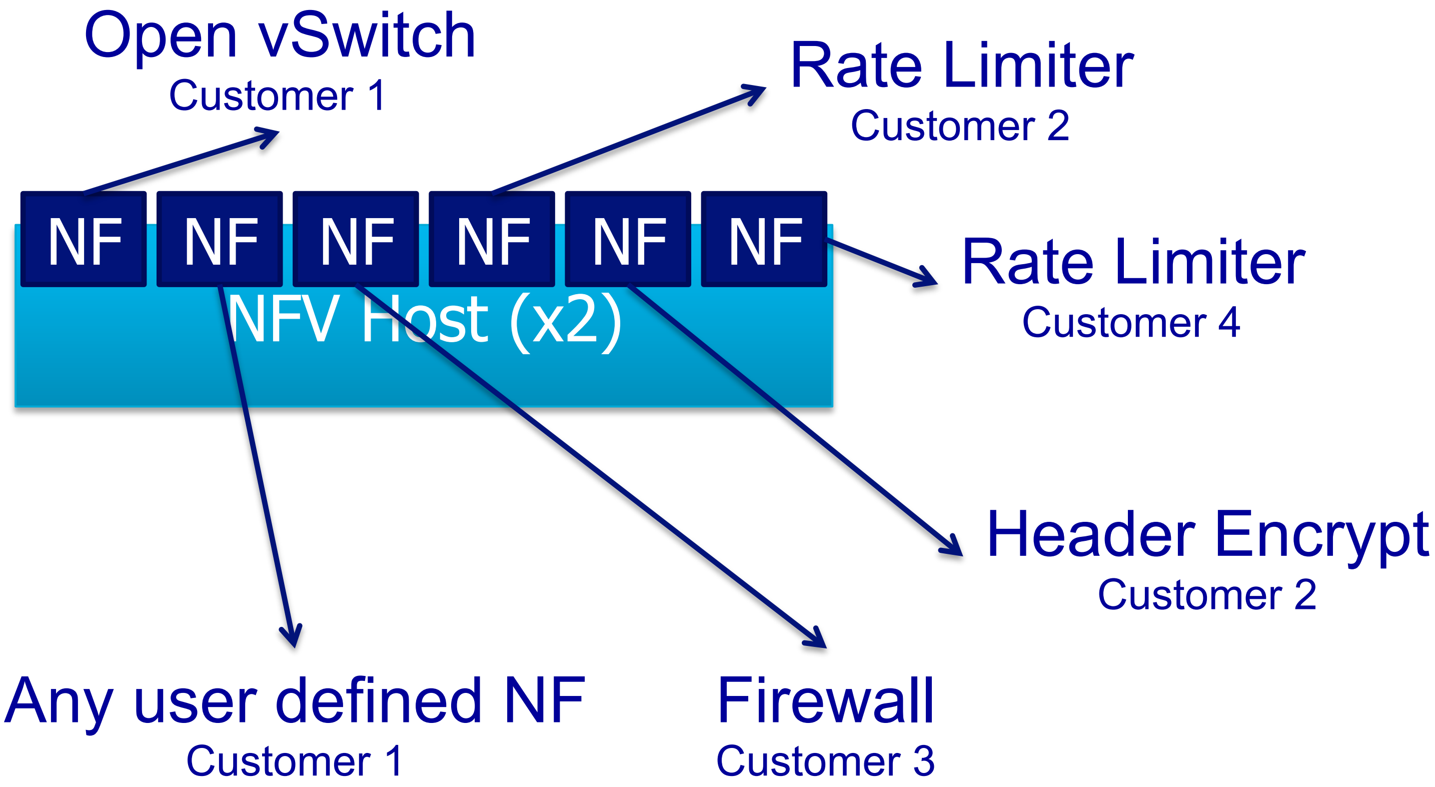
- GTS is an experimental network testbed service
  - We could use NFV “internally”
    - For monitoring purposes
    - To enforce Quality of Service
    - To implement virtualized resources (e.g., Open vSwitch)
  - Also, we can allow users to use and experiment with NFV in their network testbeds
- During the last 10 months, we took the first early steps

NFV in GTS Architecture



High-speed uplink
to remote locations

NFV in GTS Architecture

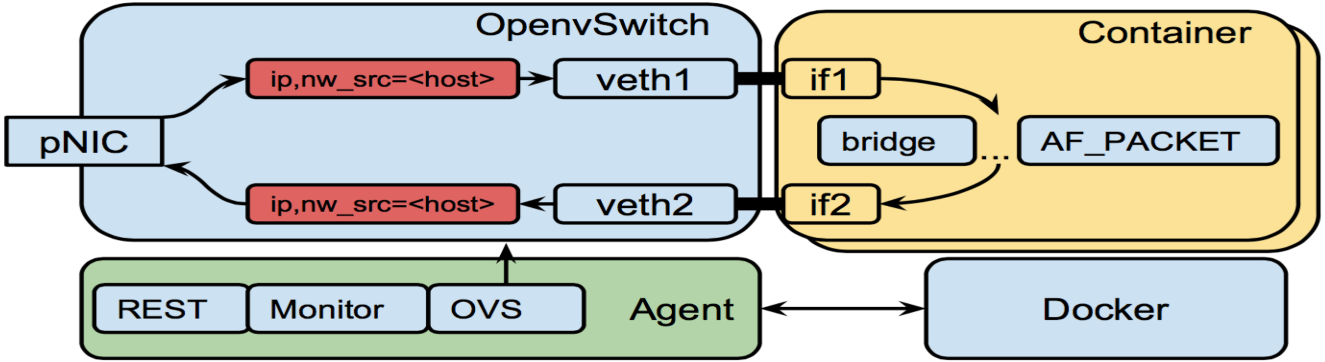


Containers for NFV

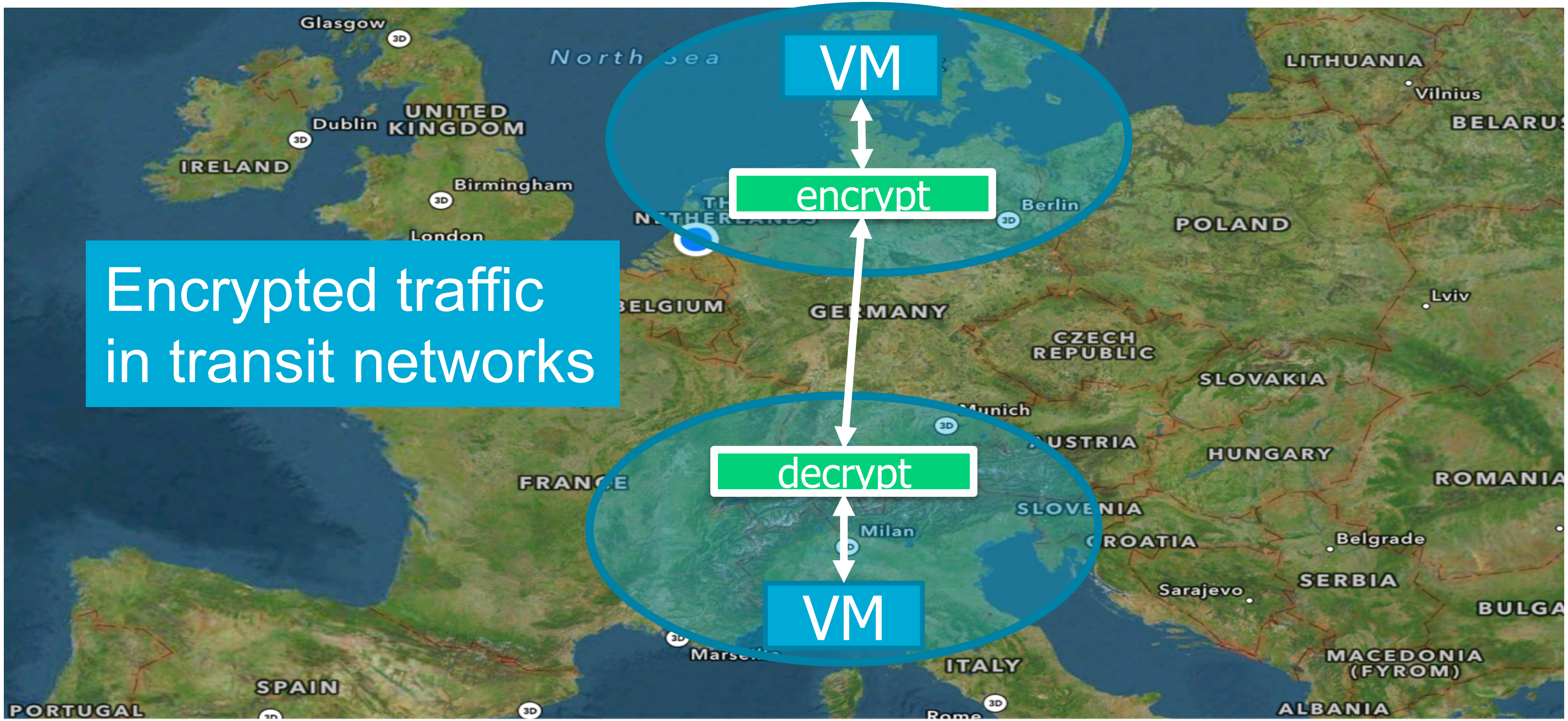
- Lightweight ”virtualization”
  - Very high NF to host density
  - Good chaining properties
- Fast create/start/stop/delete
- Small latency, high throughput, low memory usage
- Containers are easily reusable / shareable
- Traditional software environment
- Microservices architecture

What does the container do?

- Inside the NFV Host



NFV – User perspective



Encrypted traffic in transit networks

NFV DSL - hosts

```
host {
        id = "h1"
        port { id = "eth1" }
        location = ”MIA”
}
host {
        id = "h2”
        port { id = "eth1" }
        location = “GVA”
}
```

VM

VM

NFV DSL - NFs

```
nf {
        id = ”nf1”
        template = “gvs/encrypt”
        port { id = ”src” }
        port { id = ”dst” }
        location = ”MIA”
}
nf {
        id = ”nf2”
        template = “gvs/decrypt”
        port { id = ”src” }
        port { id = ”dst” }
        location = ”GVA”
}
```

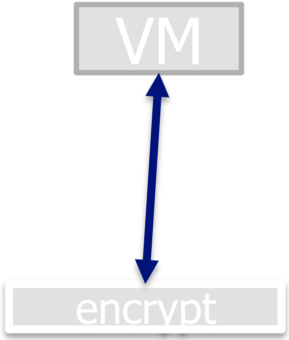
encrypt

decrypt

NFV DSL – VCs and links

```
link {
        id = ”l1”
        port { id = ”src” }
        port { id = ”dst” }
}
…

adjacency h1.eth1, l1.src
adjacency l1.dst, nf1.src
…
```

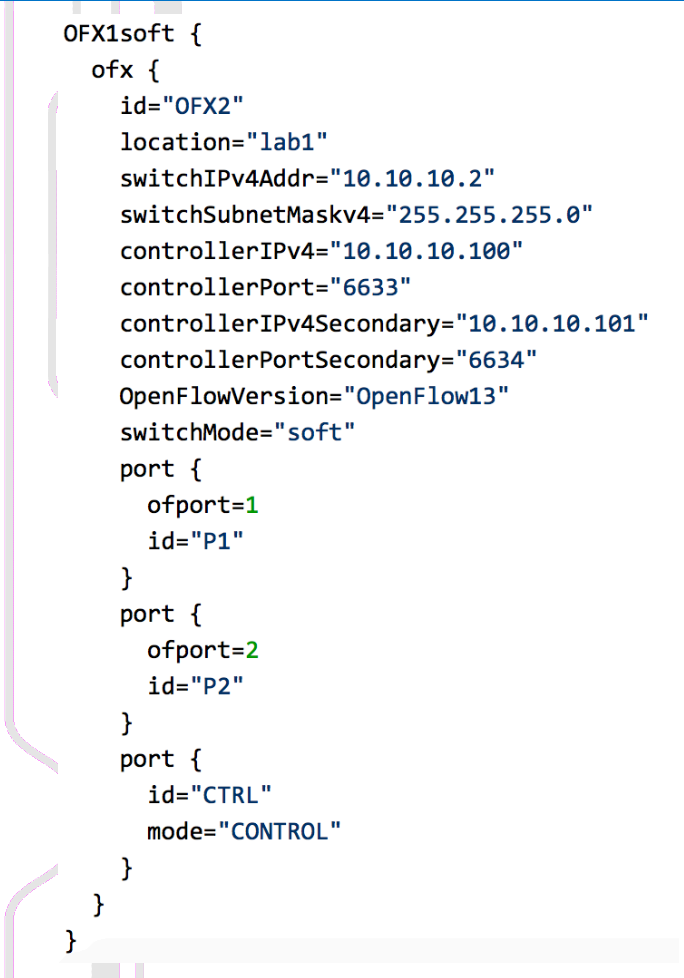


Available container NFs

- Software switch
- Firewall
- HTTP proxy
- Network measurement functions
- Introducing delay
- Rate limiter
- DNS load balancer
- SNORT

More information / publications at:
http://netlab.dcs.gla.ac.uk/

OFX OpenFlow Switch resource



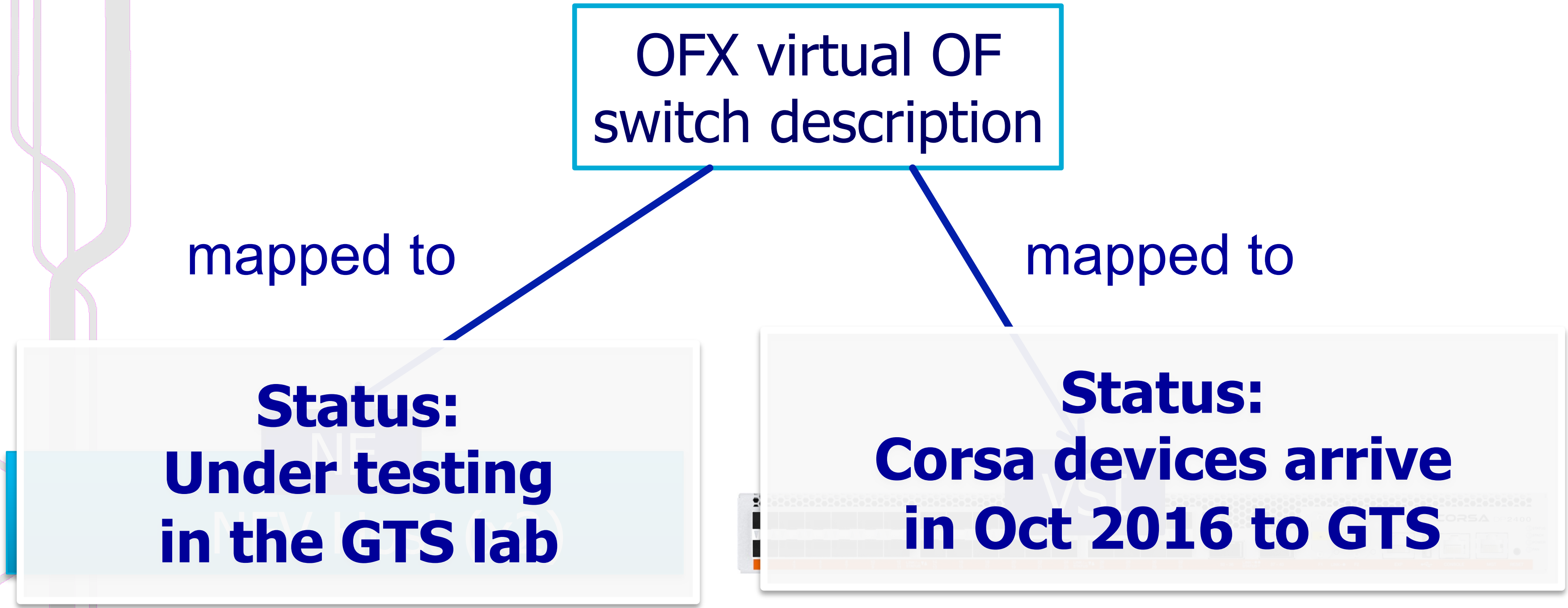
```
OFX1soft {
  ofx {
    id="OFX2"
    location="lab1"
    switchIPv4Addr="10.10.10.2"
    switchSubnetMaskv4="255.255.255.0"
    controllerIPv4="10.10.10.100"
    controllerPort="6633"
    controllerIPv4Secondary="10.10.10.101"
    controllerPortSecondary="6634"
    OpenFlowVersion="OpenFlow13"
    switchMode="soft"
    port {
      ofport=1
      id="P1"
    }
    port {
      ofport=2
      id="P2"
    }
    port {
      id="CTRL"
      mode="CONTROL"
    }
  }
}
```

- An OpenFlow resource (called OFX) can be described with a DSL
- User defines:
  - OpenFlow port (physical) ids
  - OpenFlow protocol version
  - Controllers (primary and secondary)
  - IP address of the switch
  - Switch DPID
  - Location of the resource
  - …

NORDUnet
Nordic Gateway for Research & Education

Mapping OFX to VSI / NF

- Due to the GVM used at GTS, a virtual description can be mapped to different physical (or virtual) resources



OFX virtual OF switch description

mapped to

mapped to

Status: Under testing in the GTS lab

Status: Corsa devices arrive in Oct 2016 to GTS

LIVE DEMO

- Live demo will show
  - An OFX (OpenFlow switch) mapped to a NF container
  - A rate limiter NF

Thank you

Thank you for your attention!