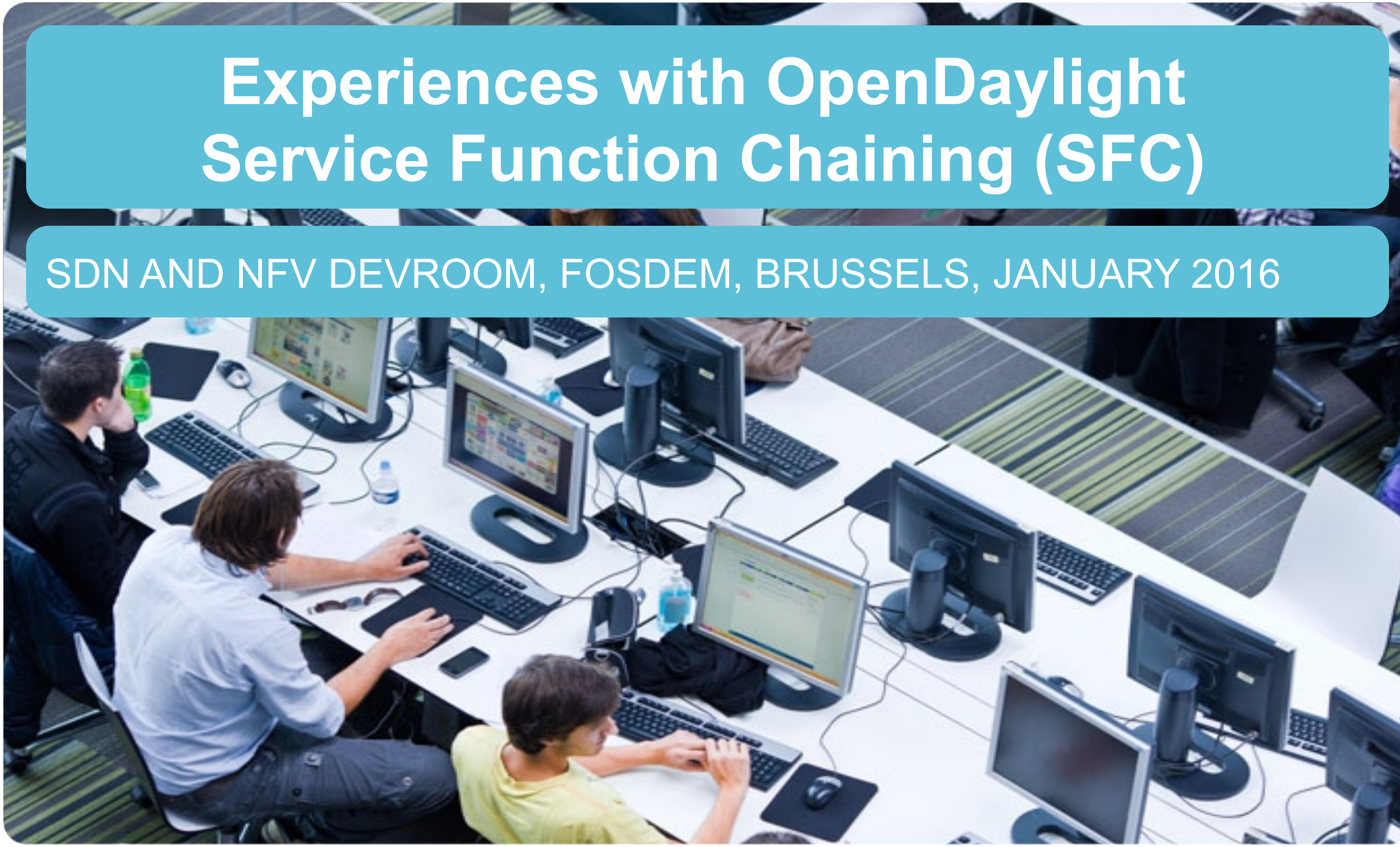
Experiences with OpenDaylight Service Function Chaining (SFC)

SDN AND NFV DEVROOM, FOSDEM, BRUSSELS, JANUARY 2016



Ronald van der Pol <Ronald.vanderPol@SURFnet.nl>

SURF NET

Overview

- About SURFnet
- Service Function Chaining (SFC)
- OpenDaylight SFC Proof of Concept
- What we learned

SURFnet Dutch National Research Network

Nationwide dark fiber infrastructure

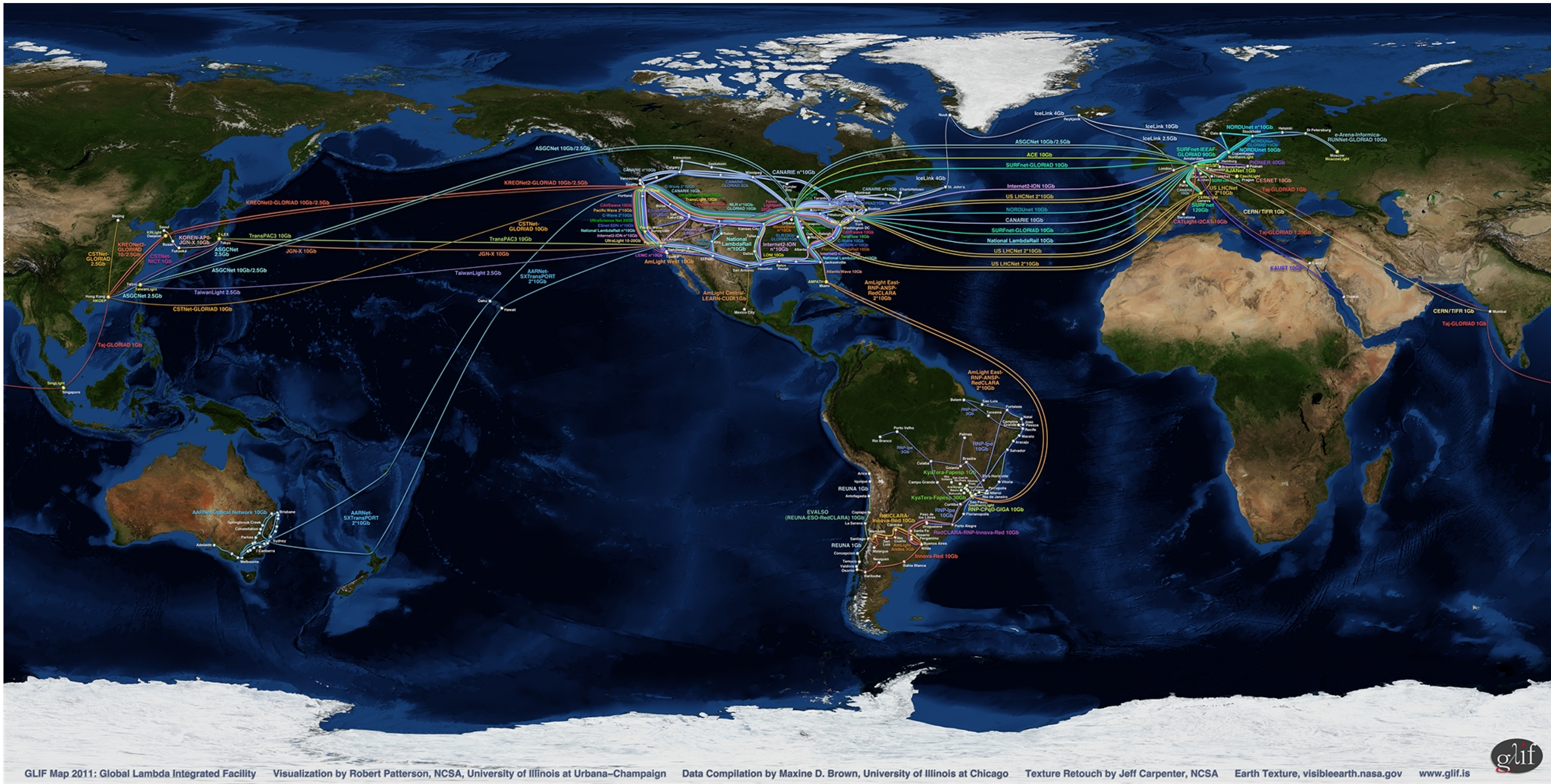
DWDM & Carrier Ethernet

Around 165 connected institutions (universities, university medical centres, research institutes)

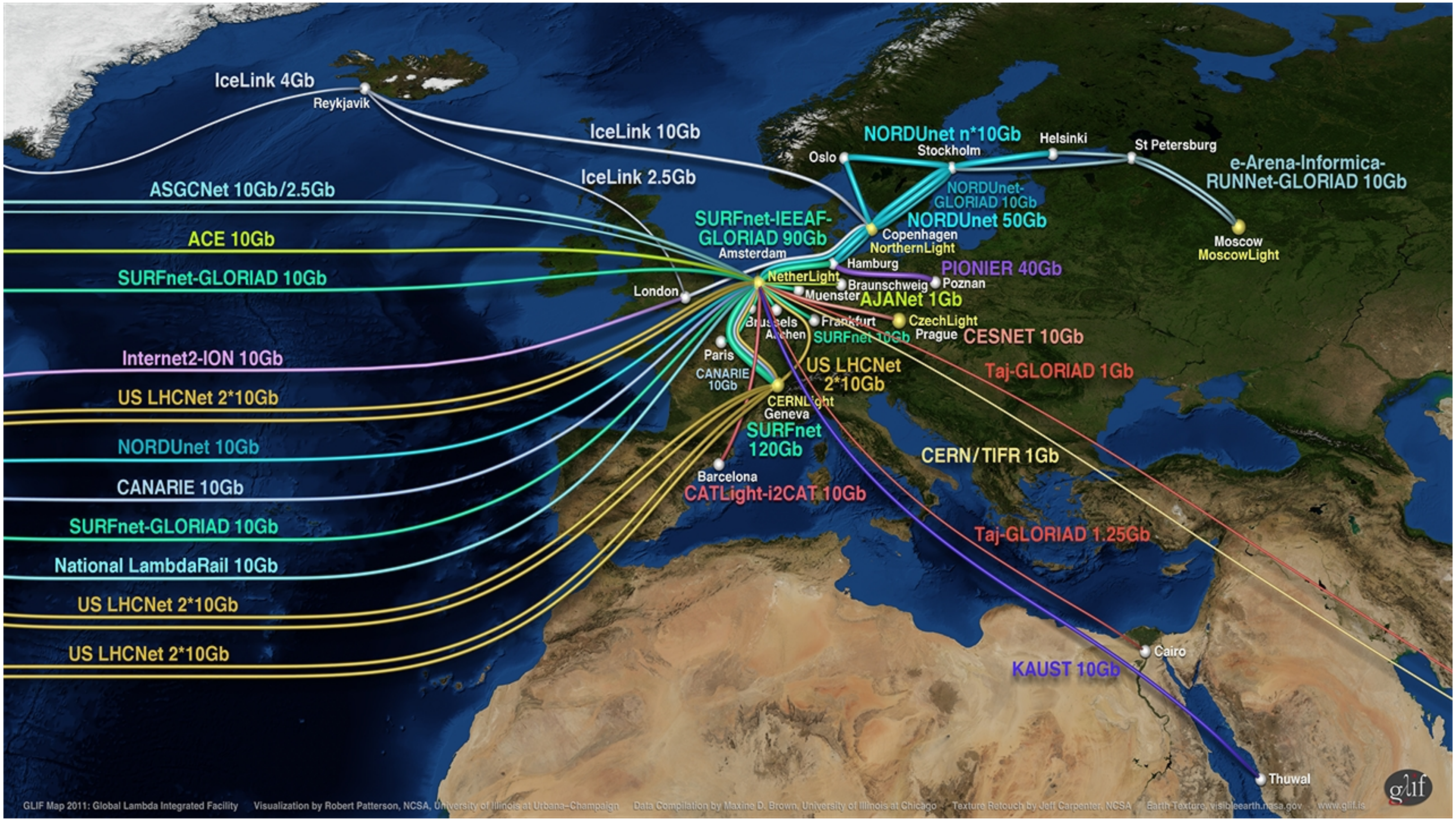
IPv4/IPv6 unicast/multicast + (dynamic) high speed P2P circuits

Federated ID, collaboration, security, wireless services & innovation

GLIF Worldwide Research Infrastructure



GLIF European Part



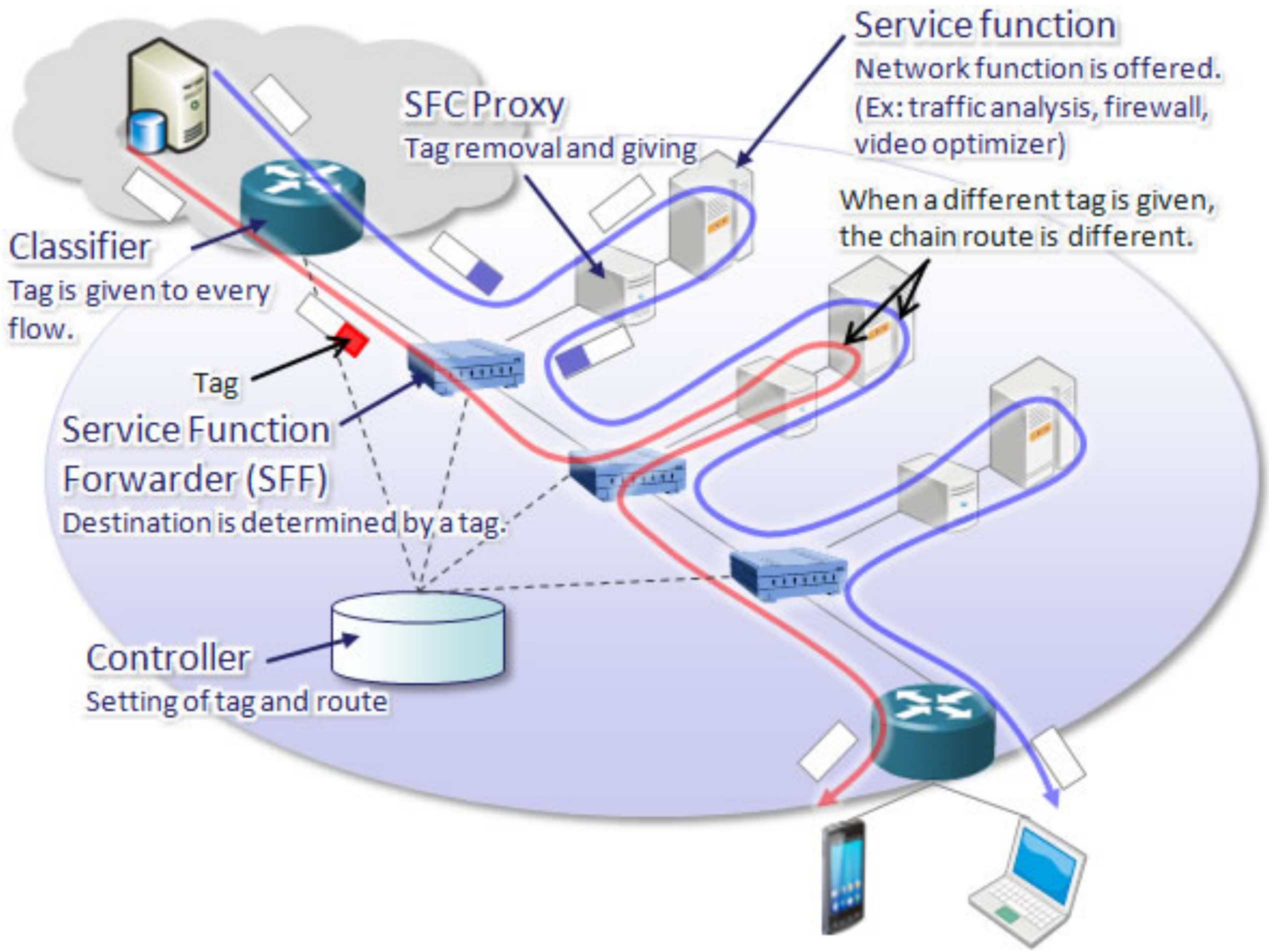
Service Function Chaining (SFC)

SFC is an architecture to steer network traffic through one or more virtual network functions.

SFC components:

- **The *Service Function* (SF) is the NFV software application.**
- **The *Service Function Chain* (SFC) defines an ordered set of Service Function (SF) types. Defines what type of SF, not which specific instance of that SF.**
- **The *Service Function Forwarder* (SFF) is responsible for forwarding network traffic to and from Service Functions.**
- **The *Service Function Path* (SFP) is a level of indirection between SFC and RSP.**
- **A *Rendered Service Path* (RSP) defines the specific SF and SFF instances defined in a SFC.**

Example



Picture by NTT

OpenDaylight SFC Proof of Concept

We wanted to learn more about NFV and SFC:

- **How does it work?**
- **What can it do?**
- **What is the maturity of the Open Source implementations?**
- **Can it be useful in the SURFnet network?**

We decided to get hands-on experience with it and build a proof of concept to be shown at SuperComputing 2015 in Austin, TX, USA.

We chose the OpenDaylight SFC implementation as a base for our PoC.

We chose to show 4K streaming video between the Netherlands and the USA because video is always a good way to show high performance networking.

High Level Concept of the PoC

We decided to steer the 4K streaming video through Service Functions that did video transcoding.

Our transcoding SFs were:

- **Add logo in the top left corner**
- **Add text in the top right corner**
- **Mirror the image**
- **Put the image upside down**
- **Convert from colour to greyscale**

All done live on a 3 Gbit/s uncompressed 4K video stream.

SFFs were 40 Gbit/s hardware OpenFlow switches. High speed (we need to be ready for 40/100 Gbit/s in the academic/scientific envirionment)

SFs were placed in clouds in Europe, so the 3 Gbit/s traffic was sent between the USA and Europe using the dedicated research network infrastructure.

4K Sender & Receiver

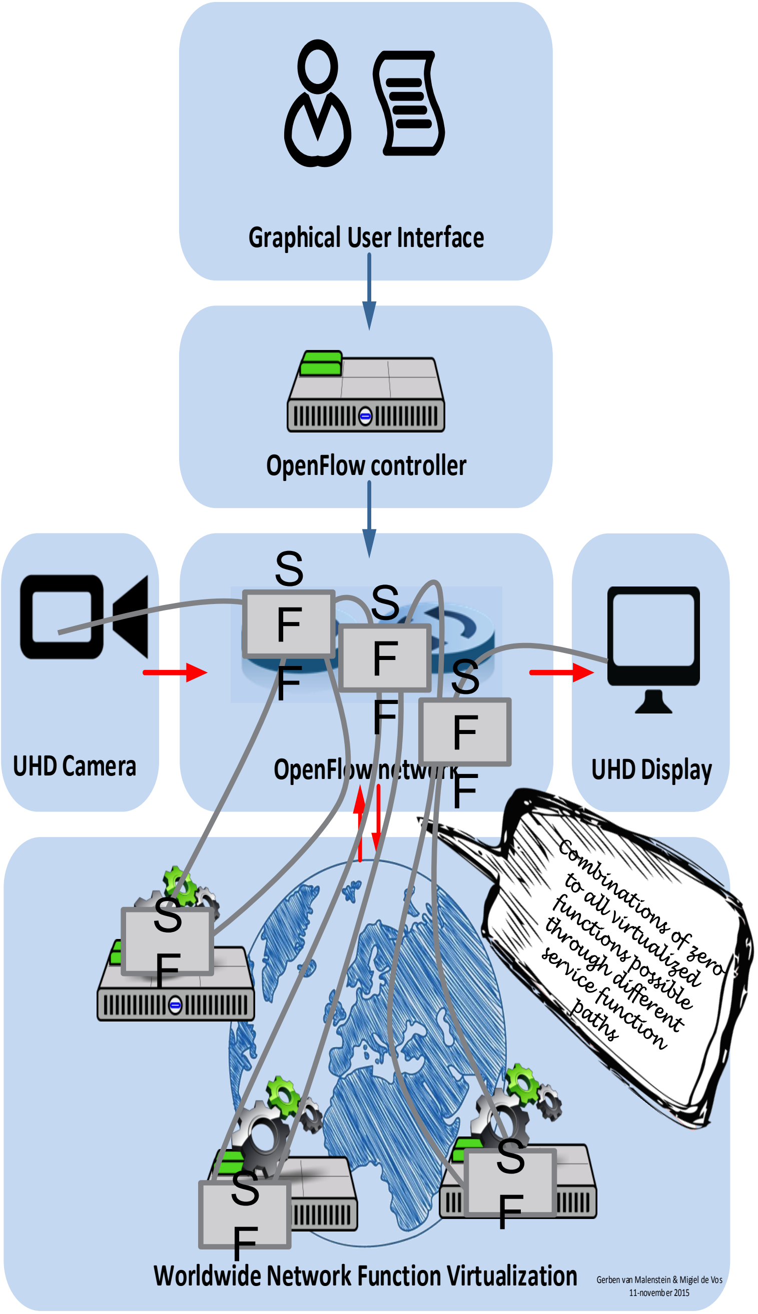
Sender (SC15-master)
10GE NIC
Design DeckLink 4K Extreme
JVC GY-HMQ10 camera
Ultragrid sender (UHD uncompressed)

Receiver (SC15-slave)
10GE NIC
GeForce GTX 970
Panasonic TX 55CX700E screen
Ultragrid receiver

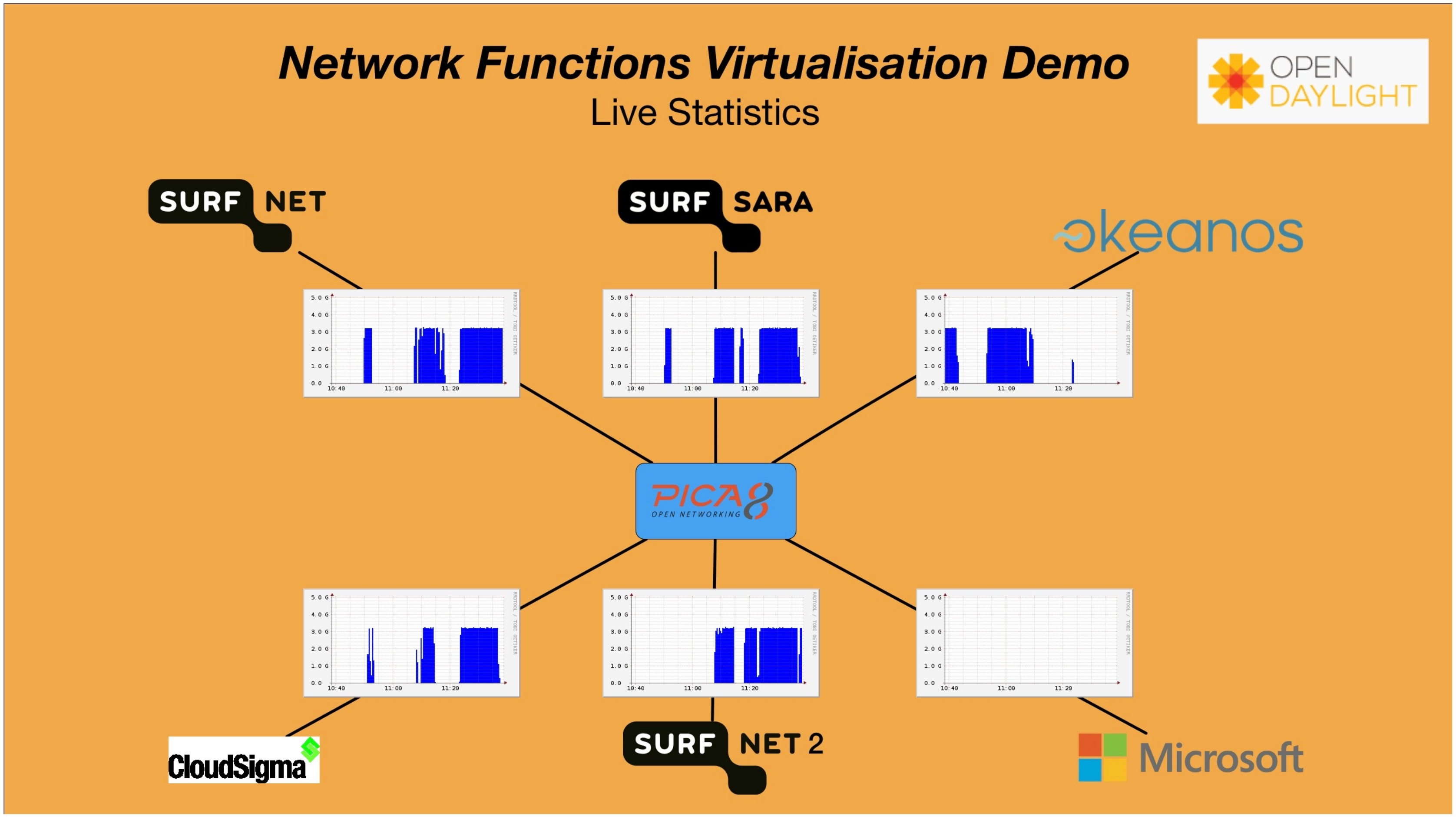
Pica8 P5101 40G OpenFlow Switches

Clouds Used

- SURFnet OpenStack testbed @ Amsterdam
- SURFsara HPC cloud @ Amsterdam
- Okeanos @ Greece
- Cloud Sigma @ Switserland
- Microsoft Azure @ Amsterdam

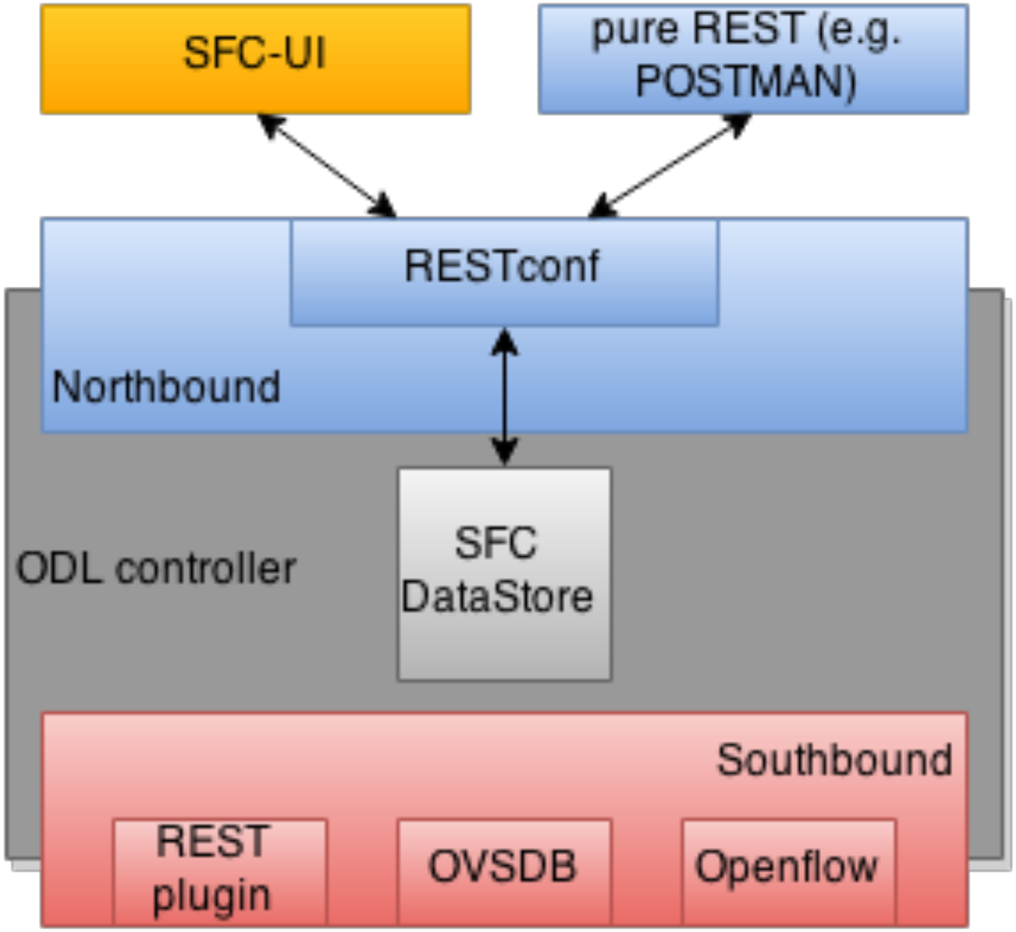

Graphical User Interface
OpenFlow controller
UHD Camera
OpenFlow network
UHD Display
Combinations of zero to all virtualized functions possible through different service function paths
Worldwide Network Function Virtualization
Gerben van Malenstein & Migiel de Vos
11-november 2015

SC15 Live Traffic Monitoring



SC15 Booth

OpenDaylight SFC OpenFlow



Our Experiences

Each new Rendered Service Path uses a new VLAN ID.
(latest VLAN ID + 100)
Counts to infinity.

Tuning needed to get 3 Gbit/s throughput.
MTU 9000 on all interfaces, including all OpenStack internal bridges.

Service Function IP and MAC addressing needed tweaking. Receiving application needs to think it is getting traffic from the sending application, not one of the virtual network functions. (next slides)

OpenDaylight uses multiple tables. Nice, but did not work on our hardware OpenFlow switches. (next slides)

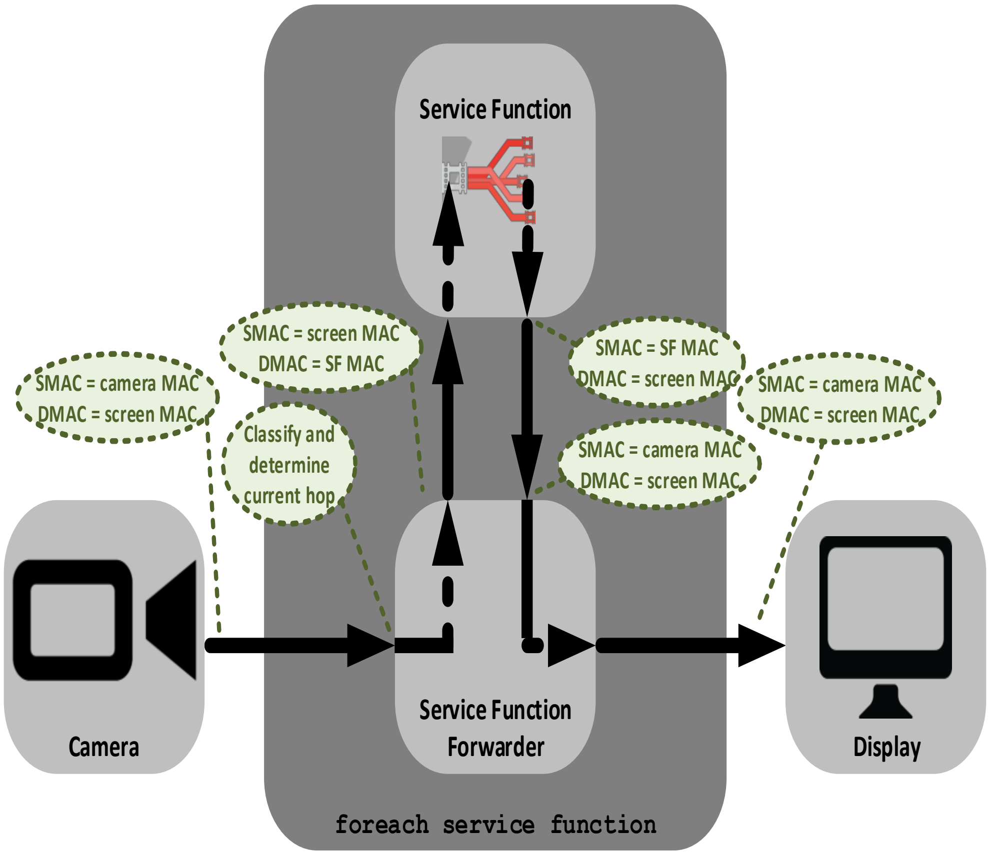
Address Tweaking

Ultragrid 4k streaming video application uses UDP, so no (TCP) sequence numbers. Good!

Usually application cares about IP tuple (src/dst IP+port). SF must be transparent. Our application did not care. Good!

But Service Function cares about destination MAC and IP. Packets have MAC and IP of final receiver. Tweaking needed.

MAC Rewriting



Service Function Addressing

DNAT

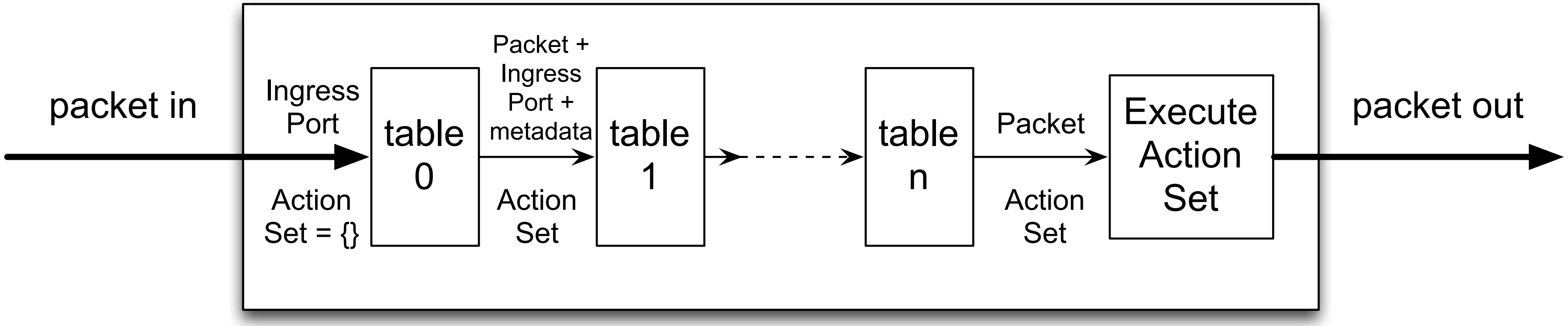
Redirect specific traffic to localhost by rewriting destination IP address
Outgoing traffic will not have the spoofed address

Transparent proxy support

Make non-local sockets work

- Redirect packets for destination address to a local socket
- Allow application to use non-local IP to transmit

OpenFlow 1.3 Pipeline



OpenDaylight SFC Table Use

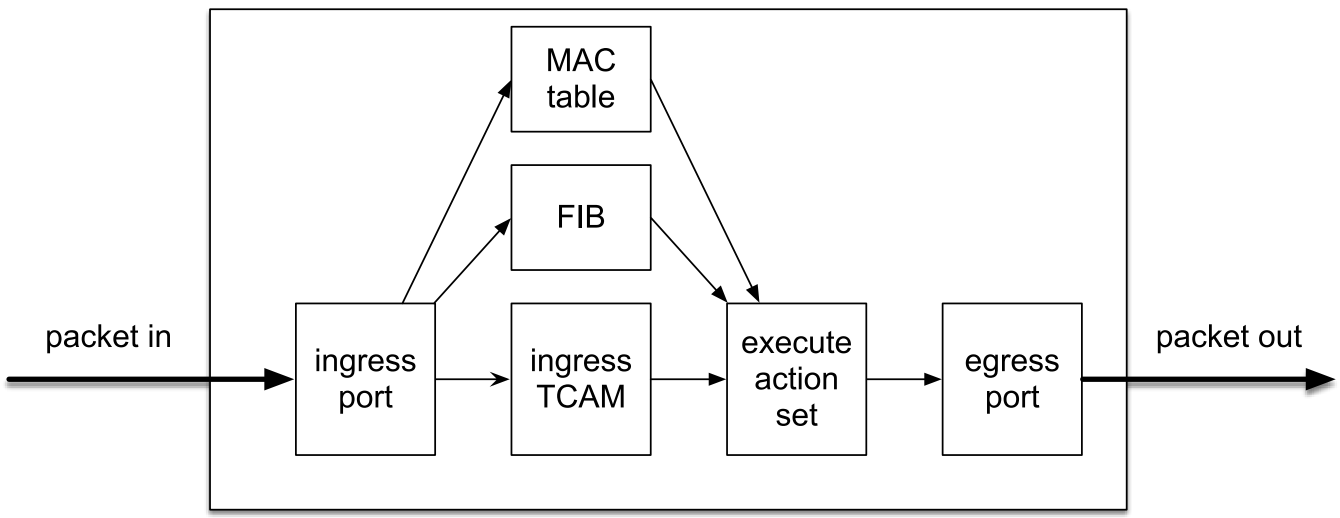
Table 0, Transport Ingress

Table 1, Path Mapper

Table 2, Next Hop

Table 10, Transport Egress

Broadcom ASIC Pipeline (simplified)



Possible Solutions

Table Type Patterns

- **Switch tells controller about its pipeline capabilities**
- **Extra complexity in the controller**

More suitable OpenFlow hardware

- **Network Processors (NoviFlow)**
- **FPGAs (Corsa)**

Protocol IndePendent Packet Processing (P4)

- **Driven by Nick Feamster and Jennifer Rexford (Princeton)**
- **Define your pipeline and send it to the switch**
- **Currently mostly software switches, some work on hardware**

Conclusions

Many Open Source projects working on NFV/SFC. Good! But takes a lot of time to keep up with all the new developments.

OpenDaylight SFC was already quite usable (2H2015). Very helpful developer community.

Focus seems on OVSDB, less on hardware OpenFlow switches. Interesting to see how we get the 40/100 Gbit/s performance we need to be prepared for.
(we are also evaluating DPDK)

We need a better understanding of how addressing of Service Functions is handled. OPNFV? Tacker?

Ronal van der Pol
Ronald.vanderPol@SURFnet.nl

WHAT SURF CAN DO