

GTS Tech+Futures Workshop (Copenhagen)

DREAMER and GN4-JRA2 on GTS

Pier Luigi Ventre

CNIT Research Unit of Rome
University of Rome Tor Vergata



Outline

- DREAMER (**D**istributed **RE**silient sdn **A**rchitecture **ME**eting carrier grade **R**equirements) Project:
 - OSHI (**O**pen **S**ource **H**ybrid IP/SDN);
 - MANTOO (**MAN**agement **TOO**ls);
 - ICONA (**I**nter **C**luster **ONOS** **N**etwork **A**pplication);
 - Deployment on GTS;
- GN4 Project and JRA2 activity:
 - SDN-IP and SDX (**S**oftware **D**efined e**X**change) L3;
 - SDX L2;
 - Future works on GTS;

DREAMER Project

<http://netgroup.uniroma2.it/DREAMER>

- The DREAMER project investigated how a network based on a SDN control plane can provide the same functionalities of an IP/MPLS control plane.
- Major outcomes:
 - OSHI – Open Source Hybrid IP/SDN networking
 - Inter Cluster ONOS Network Application (ICONA)
 - Mantoo/Topology 3D

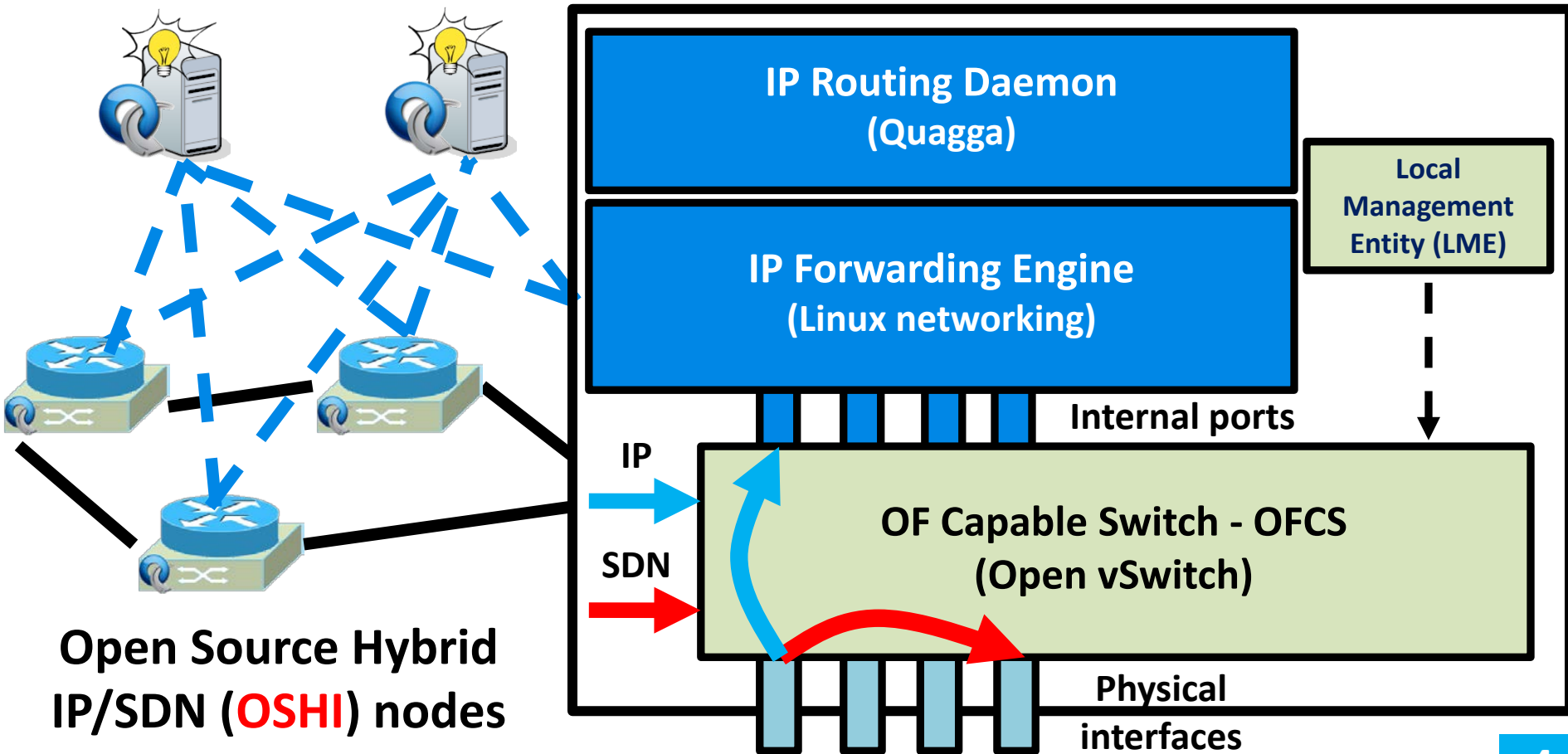


The DREAMER Project is one of the beneficiary projects of the GÉANT Open Call research initiative running from October 2013 to March 2015, see www.geant.net

OSHI Architecture

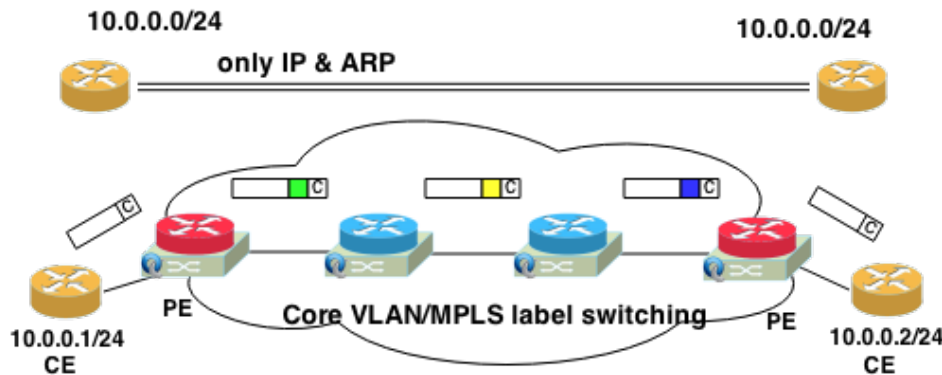
<http://netgroup.uniroma2.it/OSHI>

Hybrid IP/SDN resilient data plane

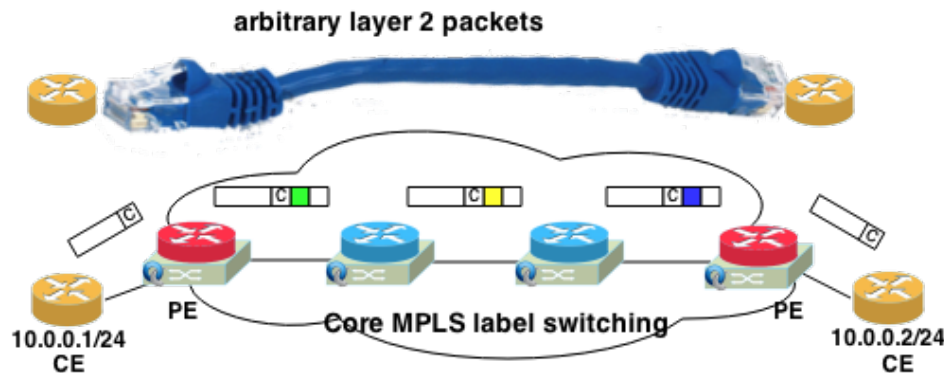


Virtual Circuit Services

IP Virtual Leased Line



L2 Pseudo Wire



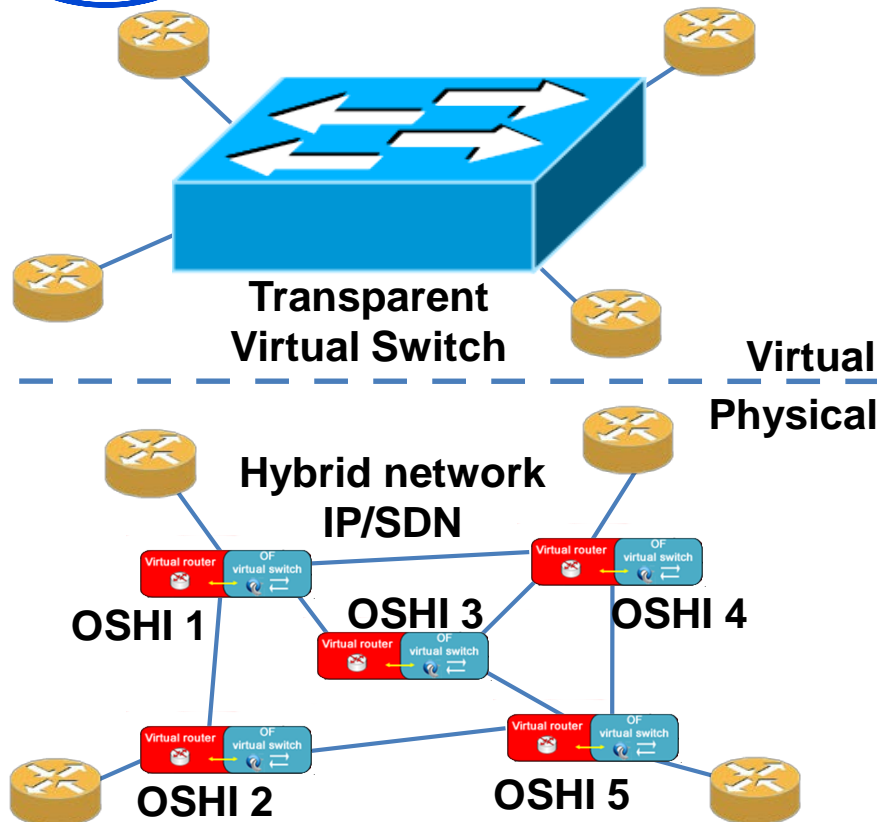
IP Virtual Leased Line (VLL)

- IPoMPLS tunnel or VLAN “tunnel”;
- MPLS-VLL can relay only IP and ARP packets;
- Supported by OpenFlow;

Pseudo Wire (PW)

- Described in RFC 3985 [1];
- EoMPLS tunnel;
- PW can relay arbitrary layer 2 packets;
- Not supported by OpenFlow, it has been realized through a GRE tunnel;

Virtual Switch Service



- **Virtual Switch Service (VSS)**
 - Described in RFC 4761 [2];
 - Built on top of PW service;
 - The network acts as big L2 switch;
 - One or more virtual switches are used to deliver this service;
 - We leverage the Steiner problem [3] to build the optimized tree of the virtual switches;

Experimental Tools (Mantoo)

<http://stud.netgroup.uniroma2.it/OSHI/TopoDesigner/>

OSHI - Topology 3D

Topology Layout Model (oshi) View (Data) Tools Deployment Info/Help

Drawing Palette-Hide

Editor Settings

oshi CR Core Router oshi PE Provider Edge CE Customer Edge Virtual Switch OF Controller Undo

deployment cer6

```
cer6 shell
> ip addr

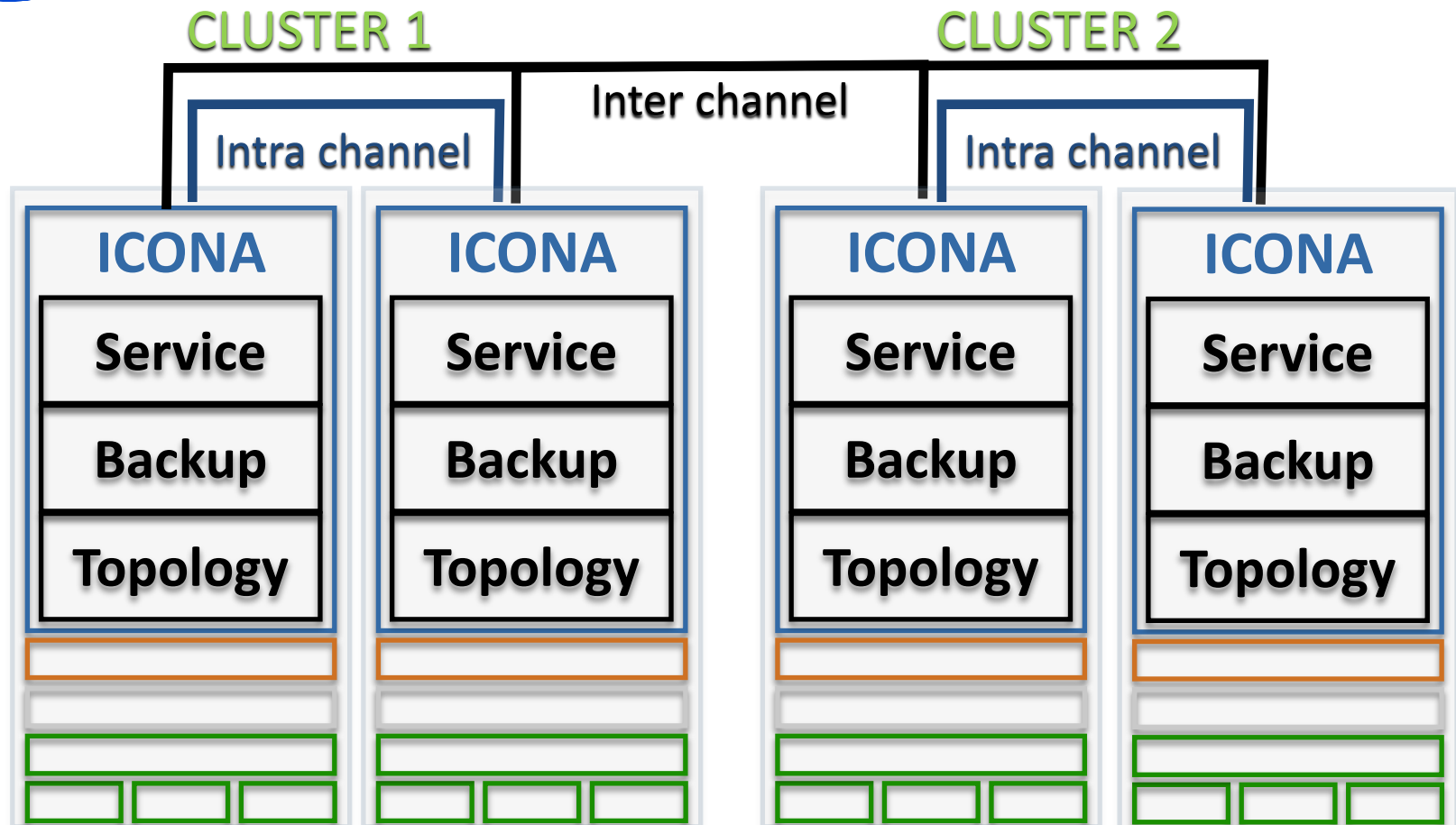
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN
link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00

inet 127.0.0.1/8 scope host lo
inet6 ::1/128 scope host
valid_lft forever preferred_lft forever
481: cer6-eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
link/ether ba:aa:6f:26:f4:8e brd ff:ff:ff:ff:ff:ff
inet 10.0.11.1/24 brd 10.0.11.255 scope global cer6-eth0
inet6 fe80::b8aa:6fff:fe26:f48e/64 scope link
valid_lft forever preferred_lft forever
```

ICONA

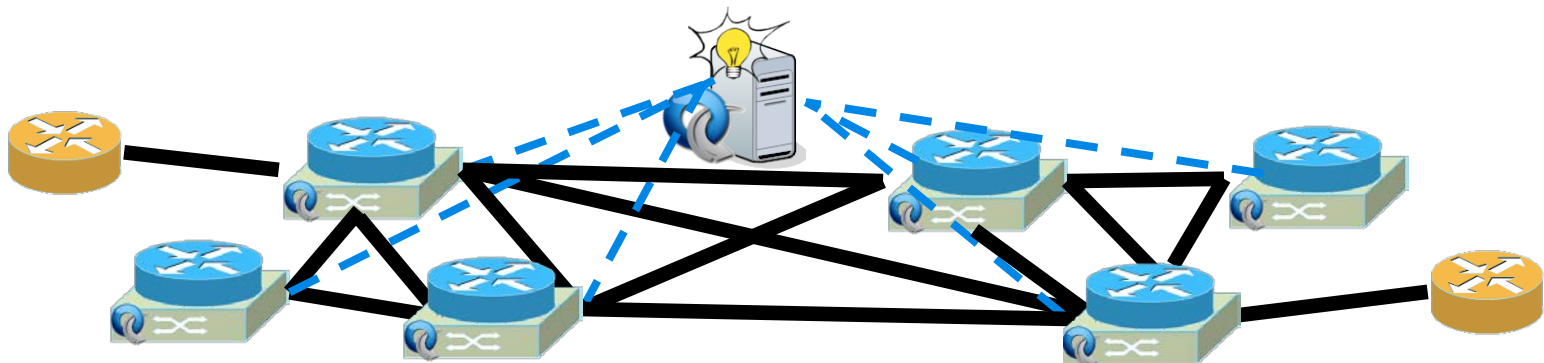
- ICONA partitions the Service Provider's network into several geographical regions, each one managed by a different cluster of ONOS instances.
- The network architect can select the number of clusters and their geographical dimension depending on requirements.
- The communication between different ICONA instances, both intra and inter-cluster, is based on Hazelcast, a multicast event-based Java Messaging System
- **ICONA internal modules:**
 - Topology; Backup; Service;

ICONA Architecture



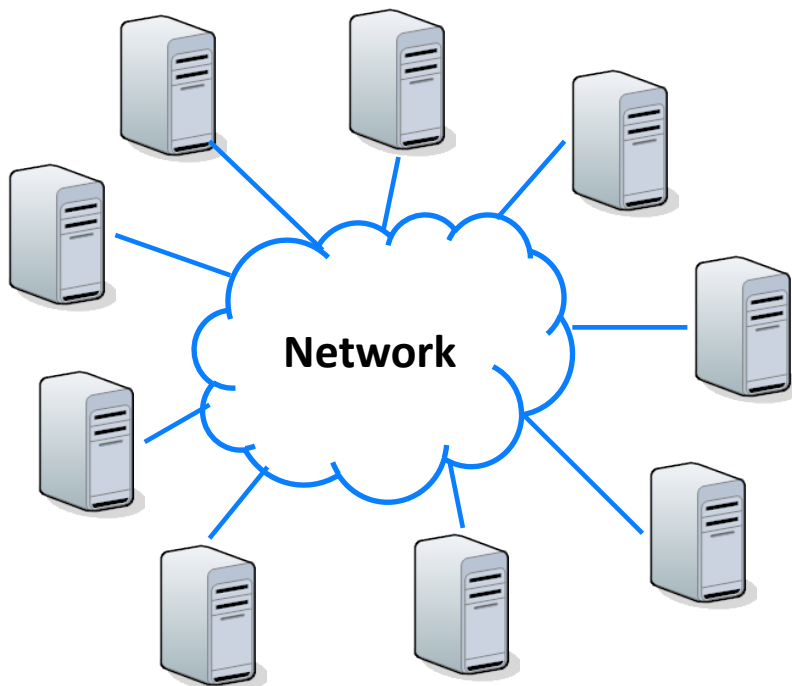
Deployment on GTS

- Mininet is a fantastic tool, we have validated a lot of functional requirements on this environment.
- For the project, we had to validate also non functional properties. In this case, Mininet could not help us;
- To give an example. We wanted to assess the feasibility of OSHI nodes: CPU overhead, memory footprint...
- We decided to use first GOFF testbed and then the GTS;

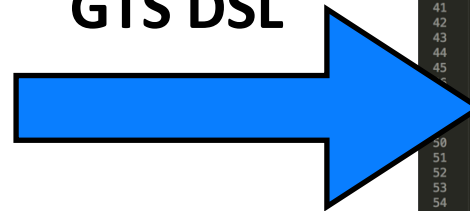


Tests Design

- Our requirements are simple:
 - VMs (Each VM is a node);
 - Interconnection among VMs;



GTS DSL



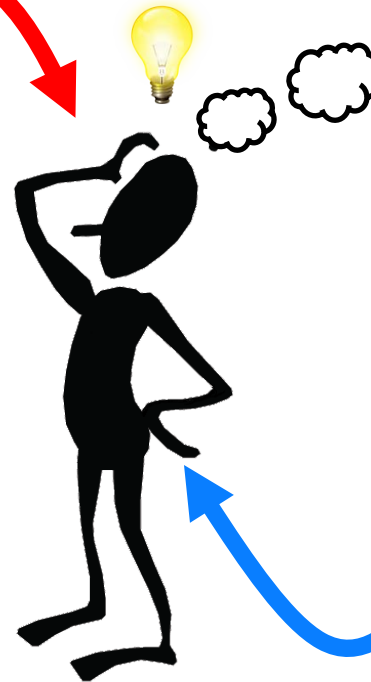
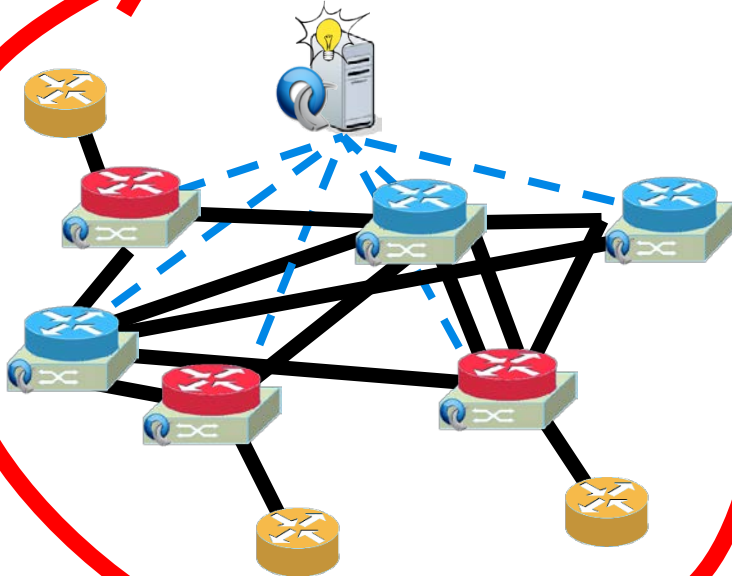
```

1 DREAMERtestbedDSL {
2   description="DREAMER testbed"
3   id="dreamerv1plv"
4   host {
5     id="ofxcontroller"
6     location="AMS"
7     port { id="eth1" }
8     port { id="eth2" }
9   }
10  ofx {
11    id="ofxAMS"
12    location="AMS"
13    fabricIPv4="10.0.0.1"
14    controllerPort="6633"
15    fabricSubnetMaskv4="255.255.255.0"
16    controllerIPv4="10.0.0.254"
17    port { id="port2" }
18    port { id="port3" }
19    port { id="port4" }
20    ...
21    port {
22      id="port1"
23      mode="CONTROL"
24    }
25  }
26  ofx {
27    id="ofxLJU"
28    location="LJU"
29    fabricIPv4="10.0.0.2"
30    controllerPort="6633"
31    fabricSubnetMaskv4="255.255.255.0"
32    controllerIPv4="10.0.0.254"
33    port { id="port2" }
34    port { id="port3" }
35    port { id="port4" }
36    ...
37    port {
38      id="port1"
39      mode="CONTROL"
40    }
41  }
42  host {
43    id="peoAMS01"
44    location="AMS"
45    port { id="port1" }
46  }
47  ...
48  host {
49    id="crlJU01"
50    location="LJU"
51    port { id="port1" }
52  }
53  link {
54    id="lkcerAMS01"
55    port { id="src" }
56    port { id="dst" }
57  }
58  ...
59  link {
60    id="lkcerAMS02"
61    port { id="src" }
62    port { id="dst" }
63  }
64  adjacency ofxcontroller.eth1, lkctrl1.src
65  adjacency ofxAMS.port1, lkctrl1.dst
66  ...
67  }
68

```

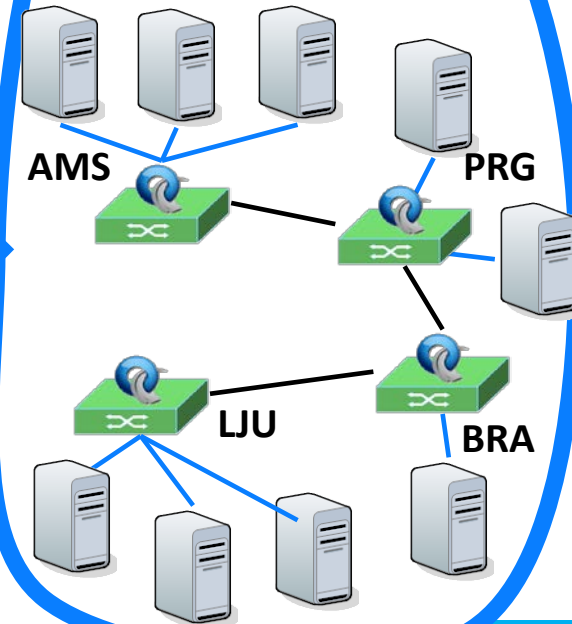
Emulation – The idea...

Experimental
Topology



VMs and
Tunneling

GTS
Topology



How to map an arbitrary topology on a set of fixed resources, with minimal configuration effort ?

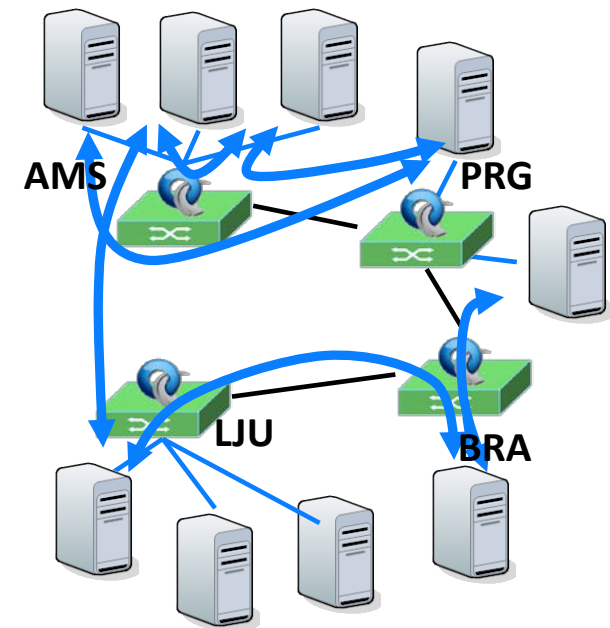
Emulation – The idea...(2)

Overlay Nodes → VMs

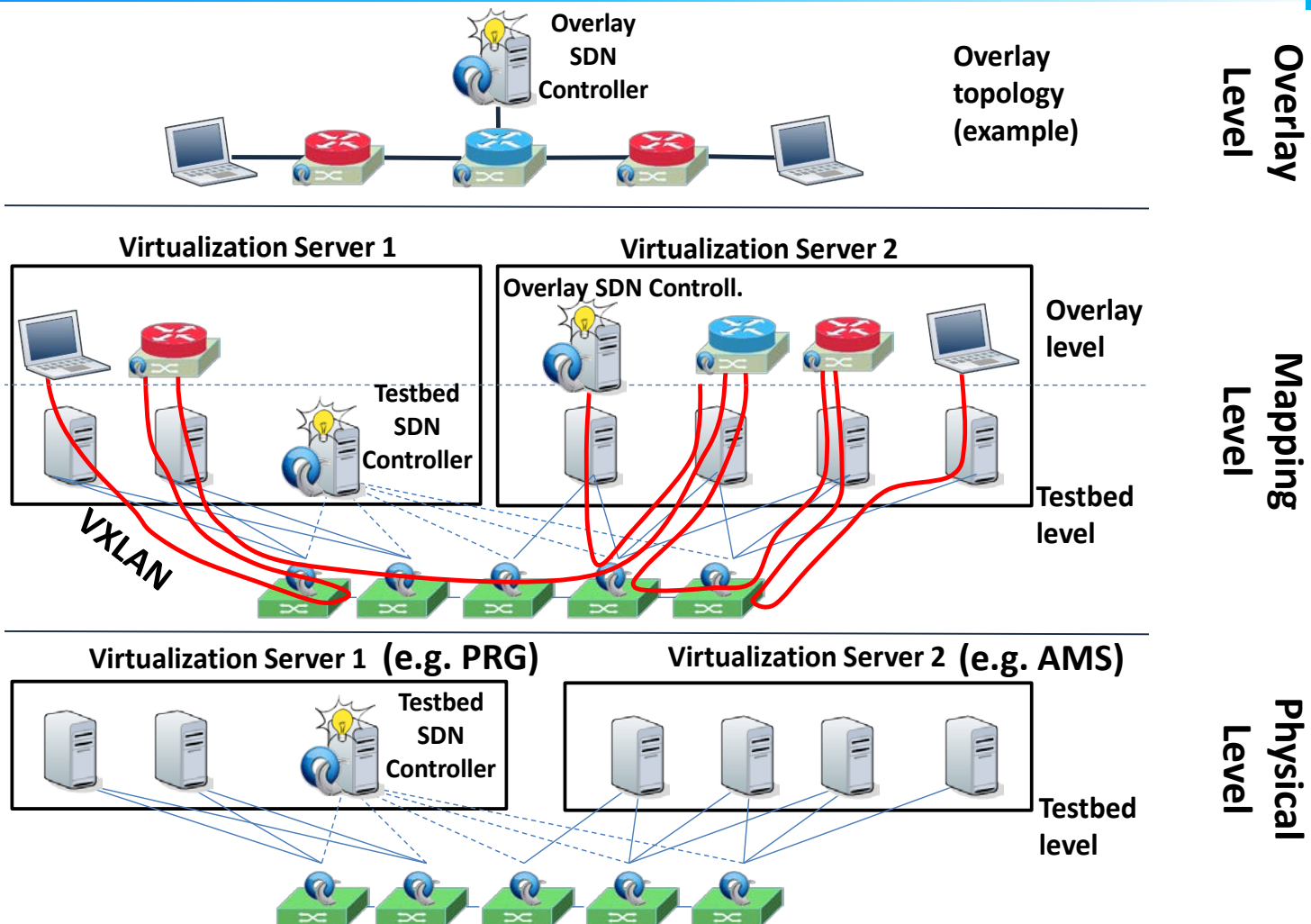
Overlay Links → Ethernet over UDP tunnels

■ Our toolset:

- VXLAN (or OpenVPN) for making tunnels
- Bash and Python scripts to automate VMs setup
- DSH for distributed setup and maintenance



Emulation on GTS



GN4-JRA2

The Activity aims to:

- Assess the feasibility of SDN/NFV/NaaS;
- Define advanced service models;
- Evaluate and extend software products for the realization of the service defined;
- Implement PoCs for use-cases;

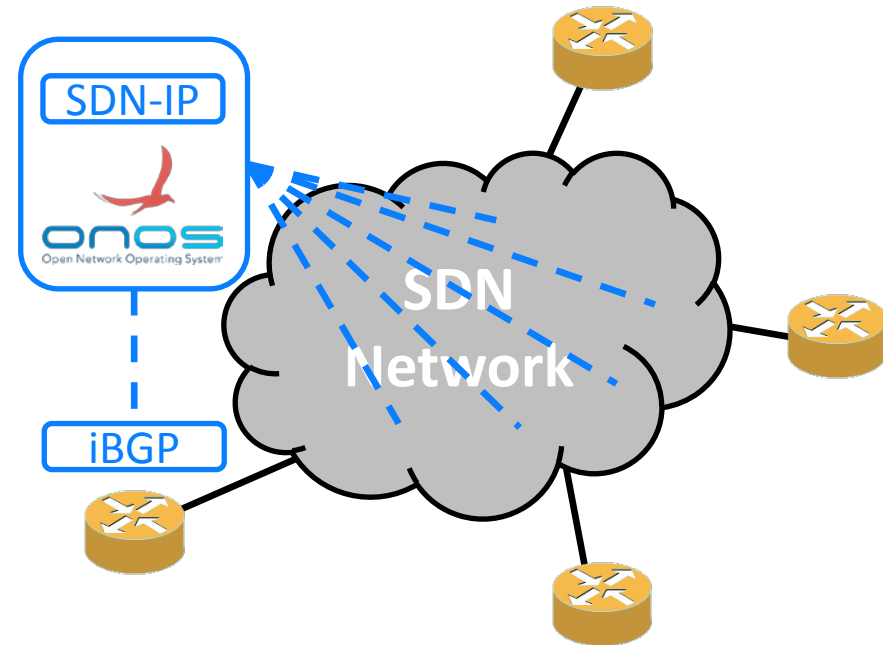
Use cases:

- Advanced path computation based on traffic demands
- Infrastructure and Network as a Service (INaaS)
- Network in the Campus
- SDN-based Optical layer service offerings
- SDN-based Bandwidth on Demand services
- SDN-IP and SDX Layer 3
- SDX Layer 2



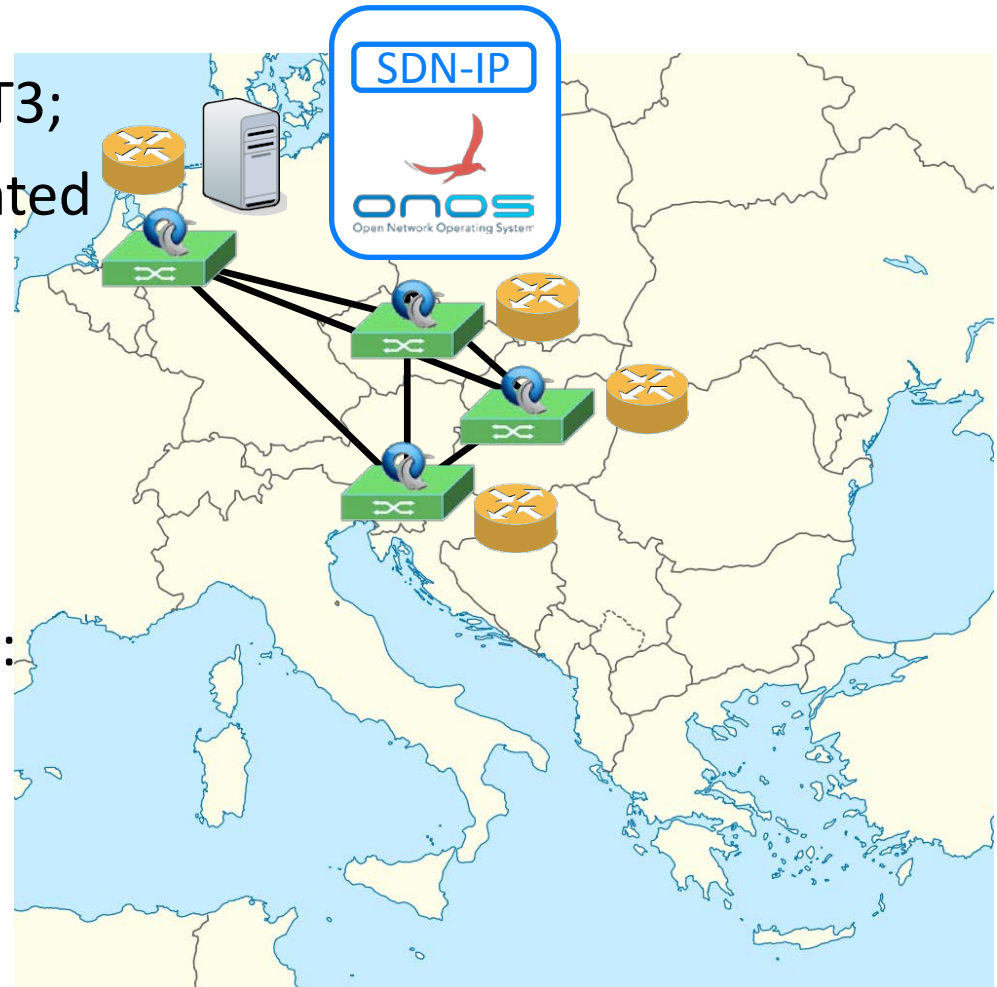
SDN-IP and SDX Layer 3

- SDX aka Software Defined internet eXchange point;
- SDN-IP[4] is a use case application developed for ONOS;
- Augment IXP functionalities using SDN capabilities:
 - Fine-grained inter-domain policies;
 - IXP peering flows are programmed;
 - Greater security and traffic control;
 - Overlay public peering or PNI among peers;



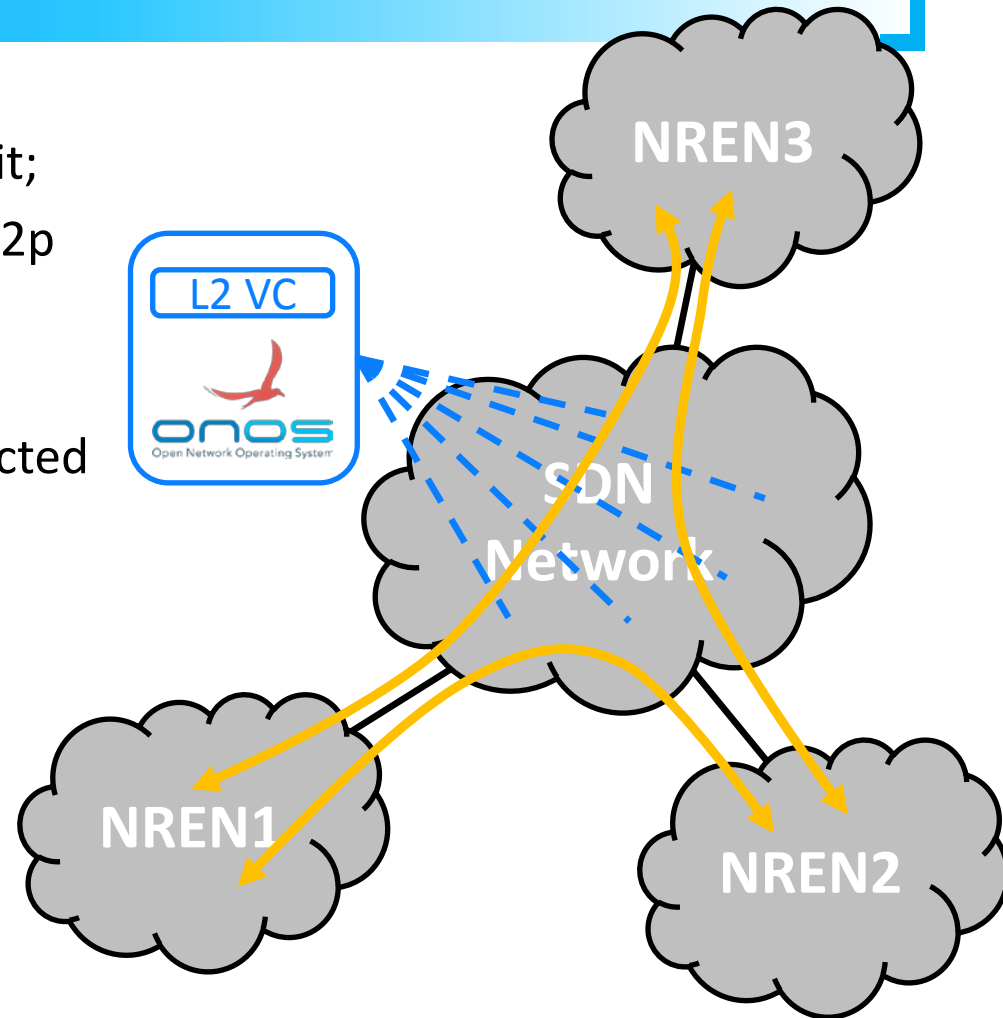
Next Steps on GTS

- We are working on T2 and T3;
- Currently, we have to validated for this UC:
 - IPv4 support - BGP Transport between BGP peers;
 - IPv6 support - BGP Transport between BGP peers;
- Resources for the scenarios:
 - 4 OF switches;
 - 6 VMs
 - 11 Links



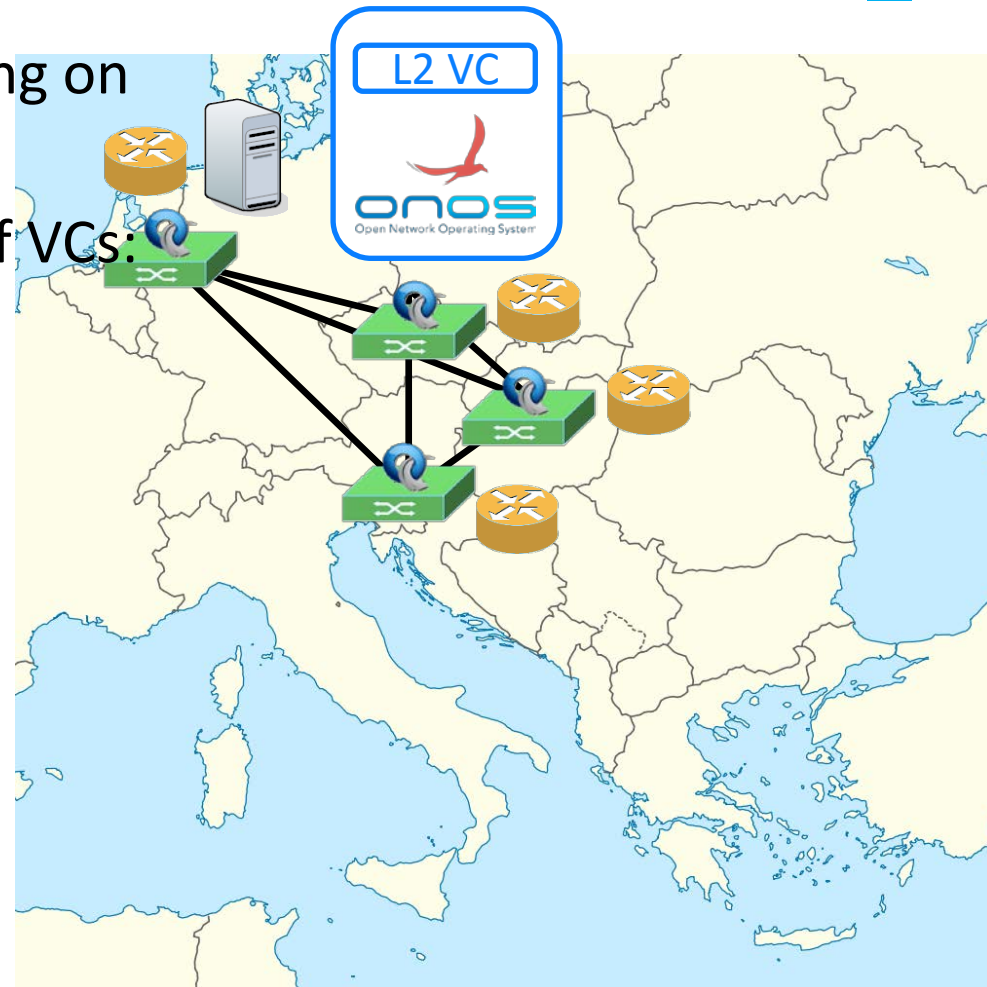
SDX Layer 2

- Traffic exchange through virtual circuit;
- Users request the provisioning of p2p services (Layer 2);
- In general it does not involve BGP;
- The IXP Layer 2 could be interconnected for multi-domain connectivity;
- Augment functionalities using SDN capabilities:
 - Speed up in the provisioning;
 - Easy expansion of the infrastructure;
 - Virtualization of the resources
 - Super SDX



Next Steps on GTS

- Also in this case, we are working on T2 and T3;
- We have to test the creation of VCs:
 - Port to port;
 - VLAN based;
 - VLAN translation;
 - MPLS;
 - ...
- Resources for the scenario:
 - 4 OF switches;
 - 5 VMs
 - 11 Links



Conclusions

- DREAMER:
 - OSHI: IP and SDN hybrid networking over Linux;
 - ICONA: Extending ONOS controller to a geographically distributed scope
 - Mantoo: Easy design, deployment and control of the experiments
- GTS:
 - Valuable and flexible platform for testing: OSHI and ICONA have been validated on GTS;
 - DSL is a good tool to create one's testbed;
 - Simple learning curve;
 - However the platform has to evolve continuously in order to satisfy its user;

References

- [1] S. Bryant, P. Pate, “Pseudo Wire Emulation Edge-to-Edge (PWE3) Architecture”, IETF RFC 3985;
- [2] K. Kompella, Y. Rekhter , “Virtual Private LAN Service (VPLS) Using BGP for Auto-Discovery and Signaling”, IETF RFC 4761;
- [3] Steiner tree – <http://mathworld.wolfram.com/SteinerTree.html>;
- [4] SDN-IP home page - <https://wiki.onosproject.org/display/ONOS/SDN-IP>