Open Source Hybrid IP/SDN (OSHI) networking: architecture, services and traffic engineering

Borsista

Pier Luigi Ventre

(pl.ventre@gmail.com)

Tutor

Stefano Salsano

(stefano.salsano@uniroma2.it)

6° Borsisti Day

24/03/2015

Roma – Consortium GARR

Outline

1. OSHI objectives & architecture
2. Services:
   - Virtual Leased Lines (VLL) and Pseudo Wires (PW)
   - Virtual Switch Service (VSS);
3. Experimental tools (Mantoo)
4. Monitoring & Traffic Engineering
5. Conclusions and future directions;

Open Source Hybrid IP/SDN

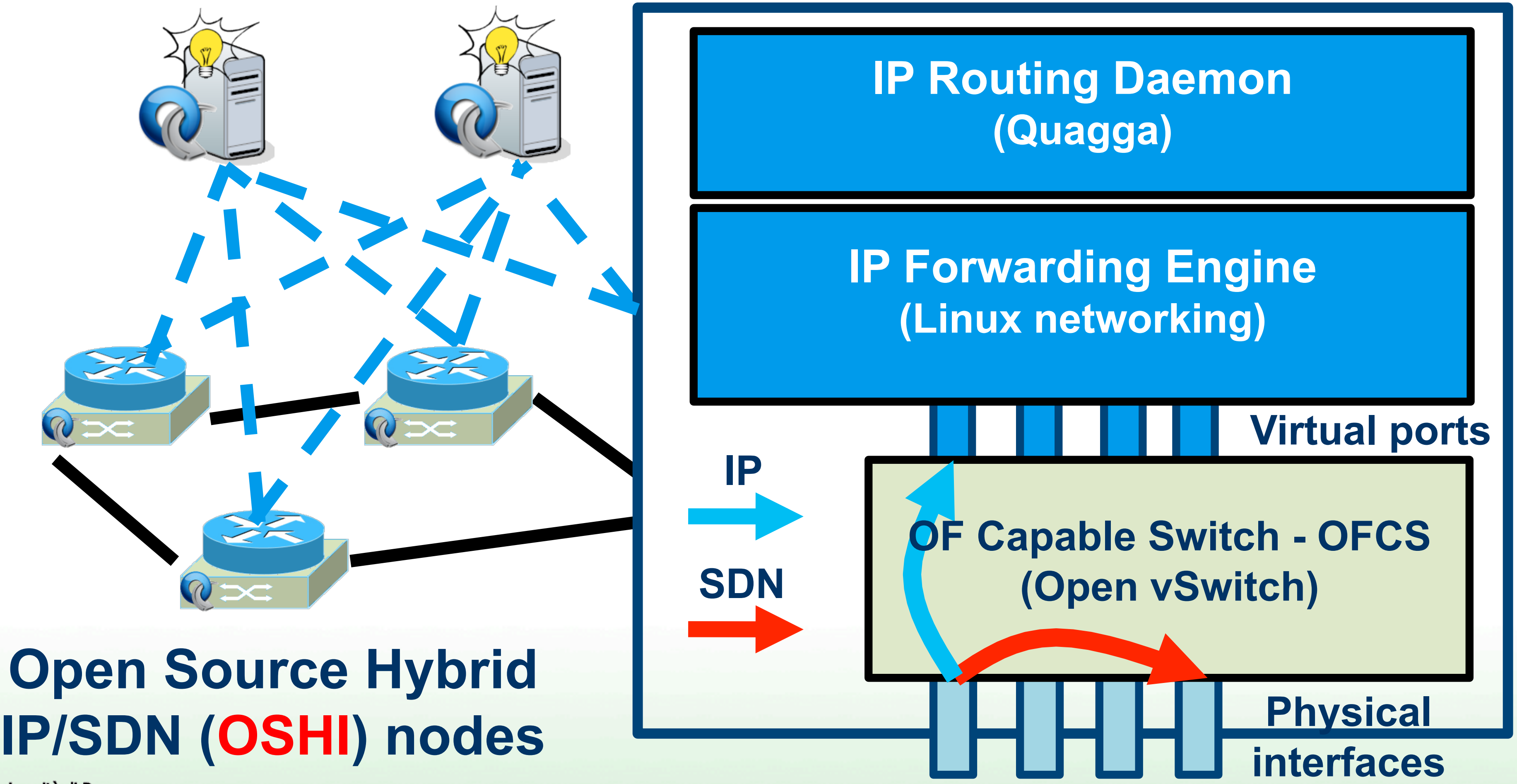
- **Investigate how to introduce SDN/OpenFlow in large-scale IP backbones:**
  - How to replicate the services of IP/MPLS networks… and their non-functional properties (“carrier grade”) ?
  - How to scale SDN/OpenFlow from data-centers to IP WAN backbones ?

- **Do it in an open way !!**
  - Open source components
  - Simple tools for setting up and performing experiments

- **Provide an experimental platform with no entry barrier**

OSHI architecture

Hybrid IP/SDN resilient data plane



IP Routing Daemon
(Quagga)

IP Forwarding Engine
(Linux networking)

Virtual ports

IP

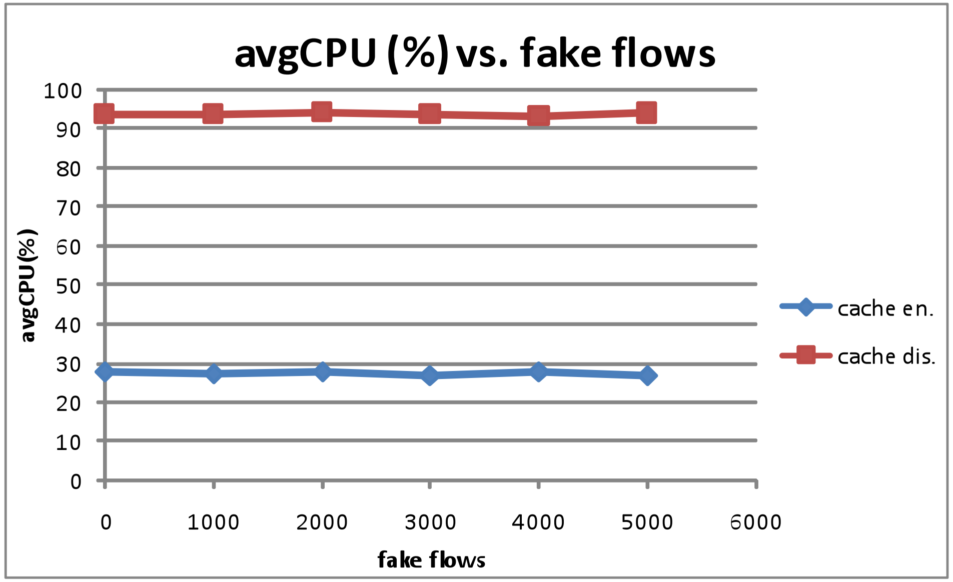
SDN

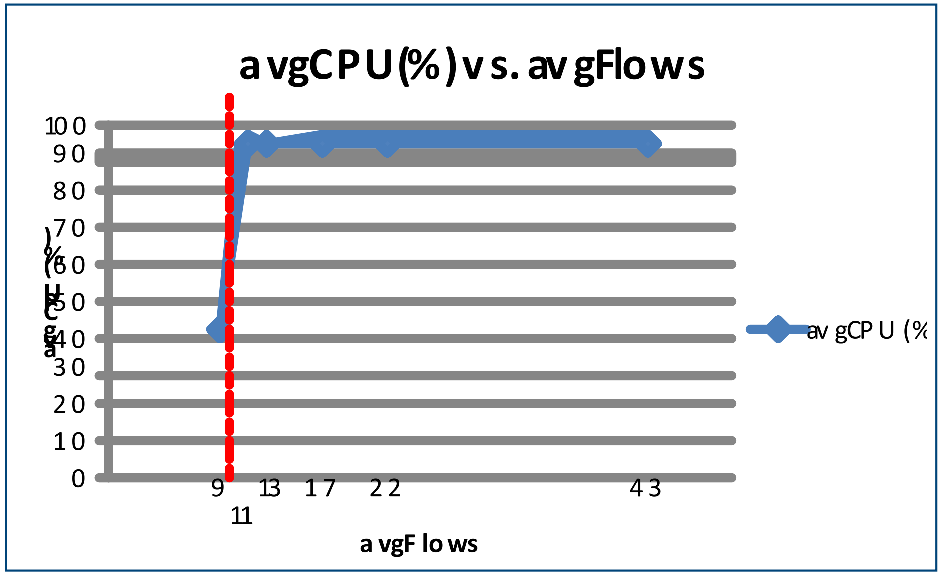
OF Capable Switch - OFCS
(Open vSwitch)

Physical interfaces

Open Source Hybrid IP/SDN (**OSHI**) nodes

Performance evaluation (brief)

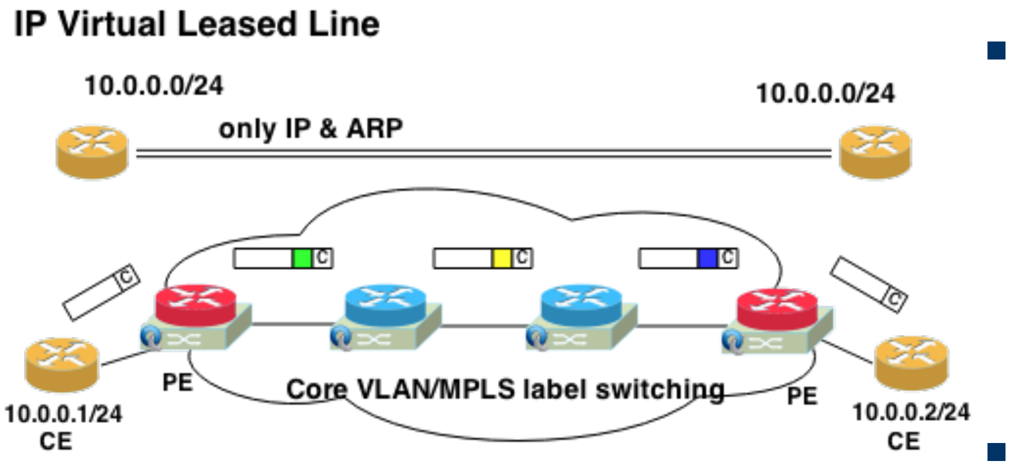


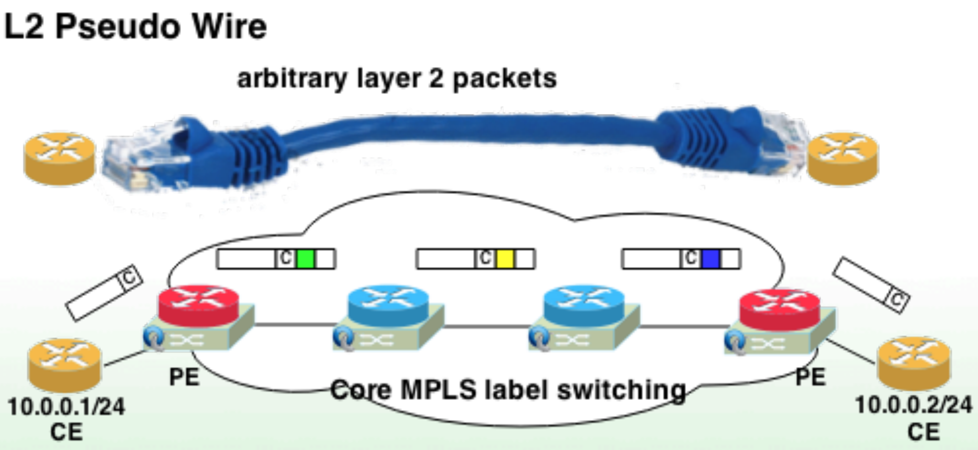


Effects of larger flow table on the OVS performance

Effects of the kernel cache on the OVS performance

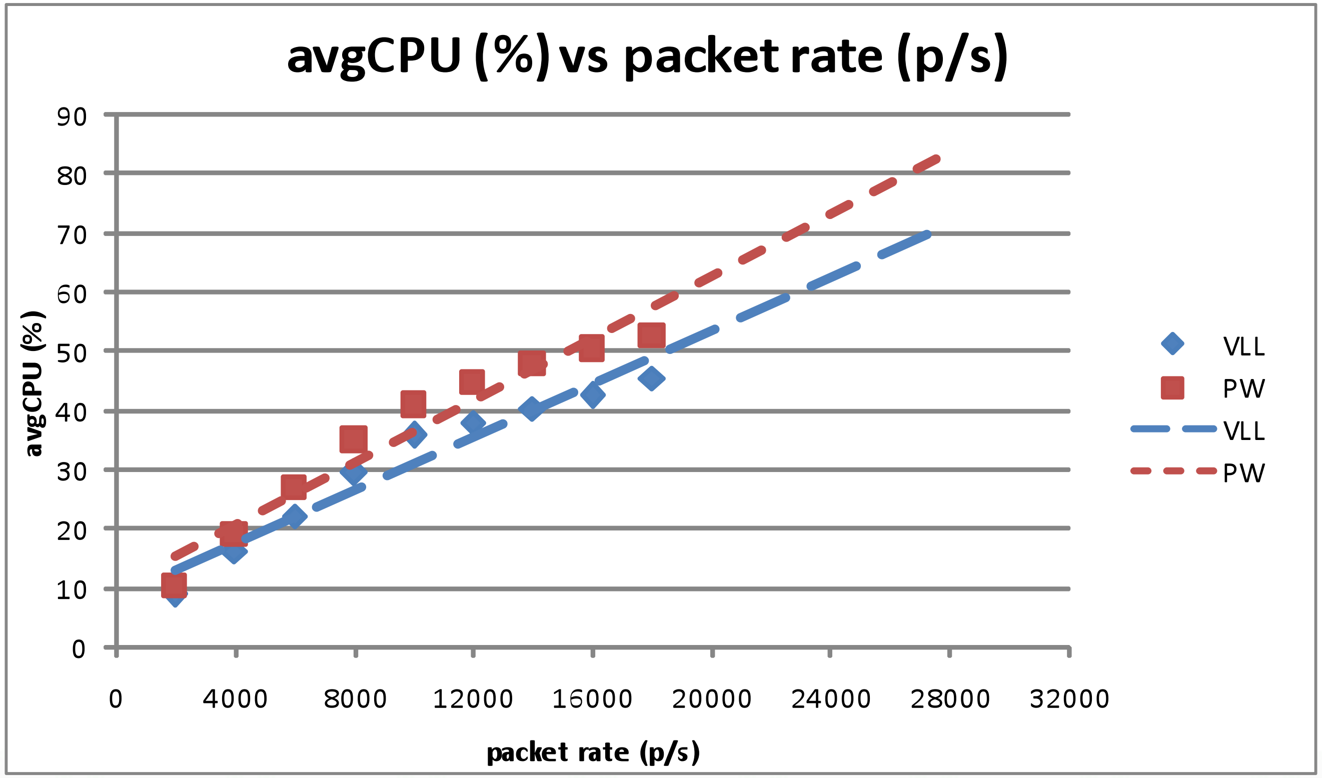
Services: Virtual circuits





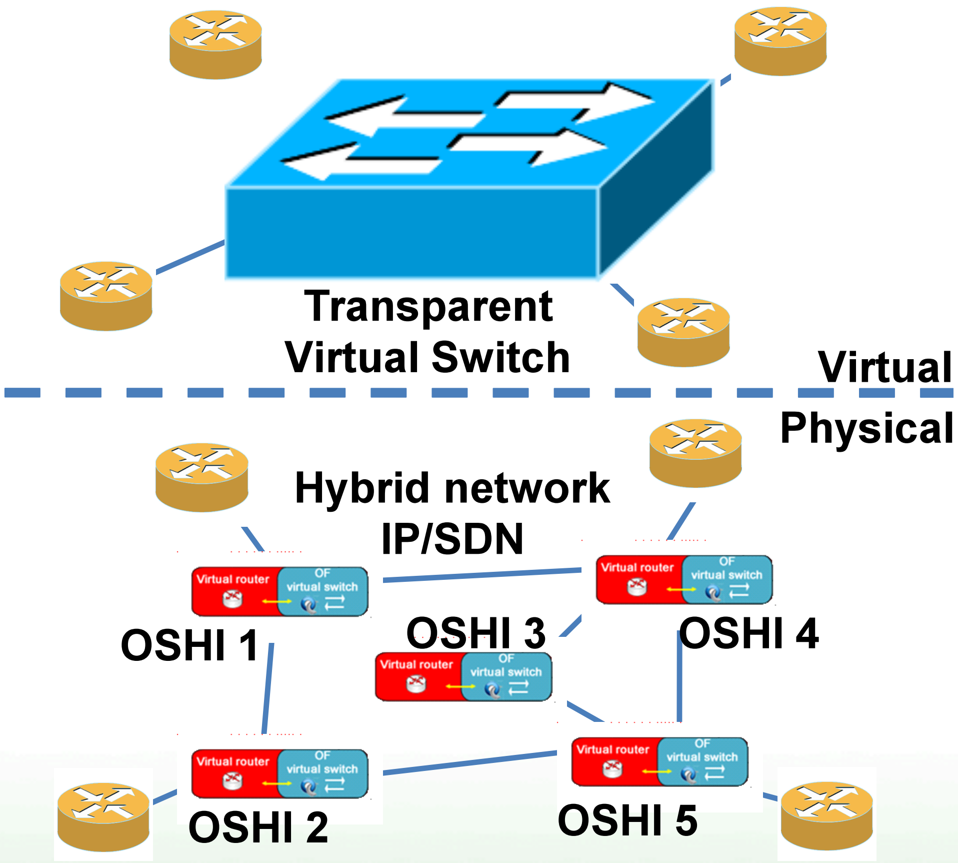
- **IP Virtual Leased Line (VLL)**
  - IPoMPLS tunnel or VLAN “tunnel”;
  - MPLS-VLL can relay only IP and ARP packtes;
  - Supported by OpenFlow;
- **Pseudo Wire (PW)**
  - Described in RFC 3985 [6];
  - EoMPLS tunnel;
  - PW can relay arbitrary layer 2 packets;
  - Not supported by OpenFlow, it has been realized through a GRE tunnel;

VLL versus PW (brief)



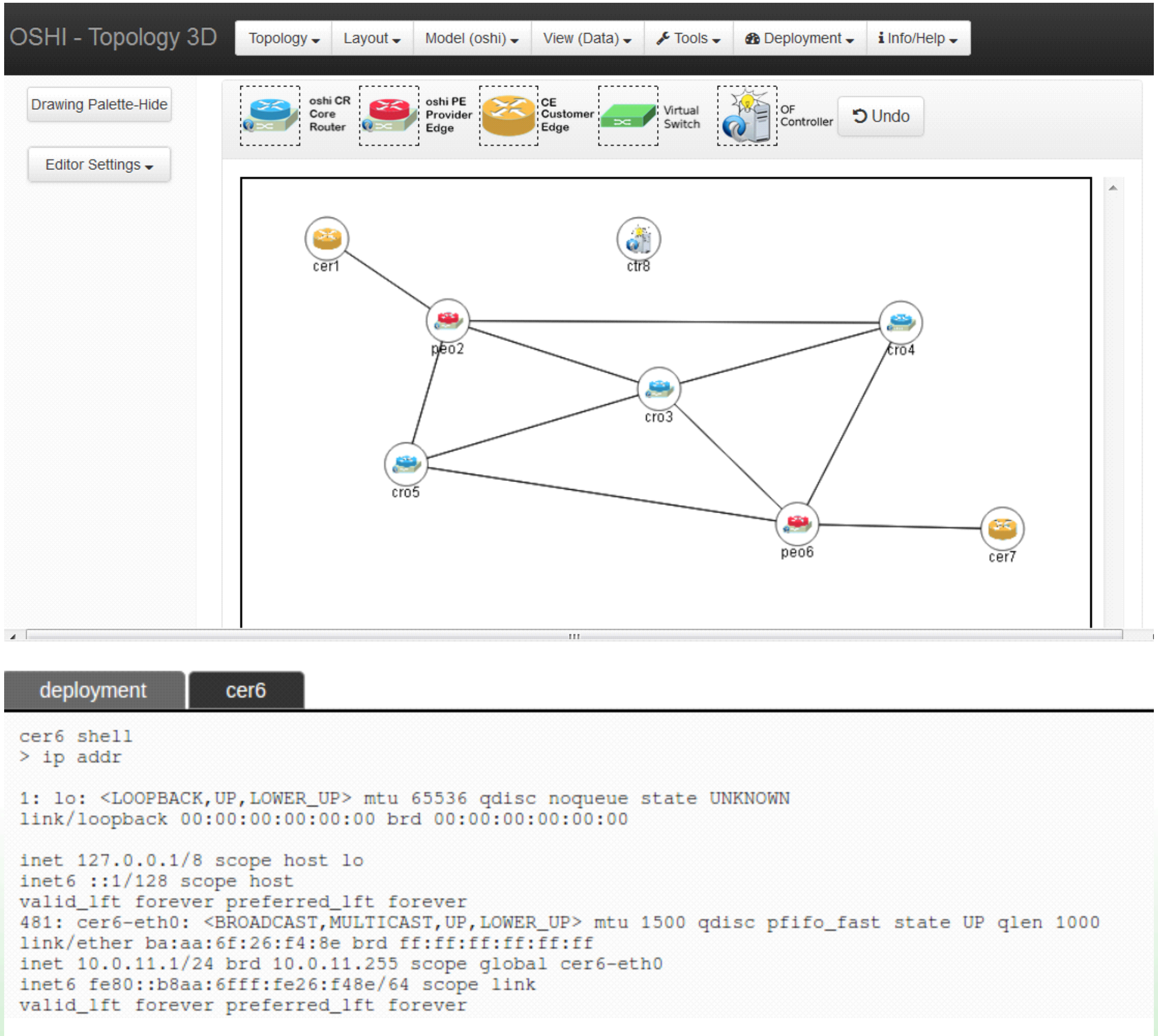
Performance assessment of PW service

Virtual Switch Service



- **Virtual Switch Service (VSS)**
  - Described in RFC 4761 [7];
  - Built on top of PW service;
  - The network acts as big L2 switch;
  - One or more virtual switches are used to deliver this service;

Experimental tools (Mantoo)

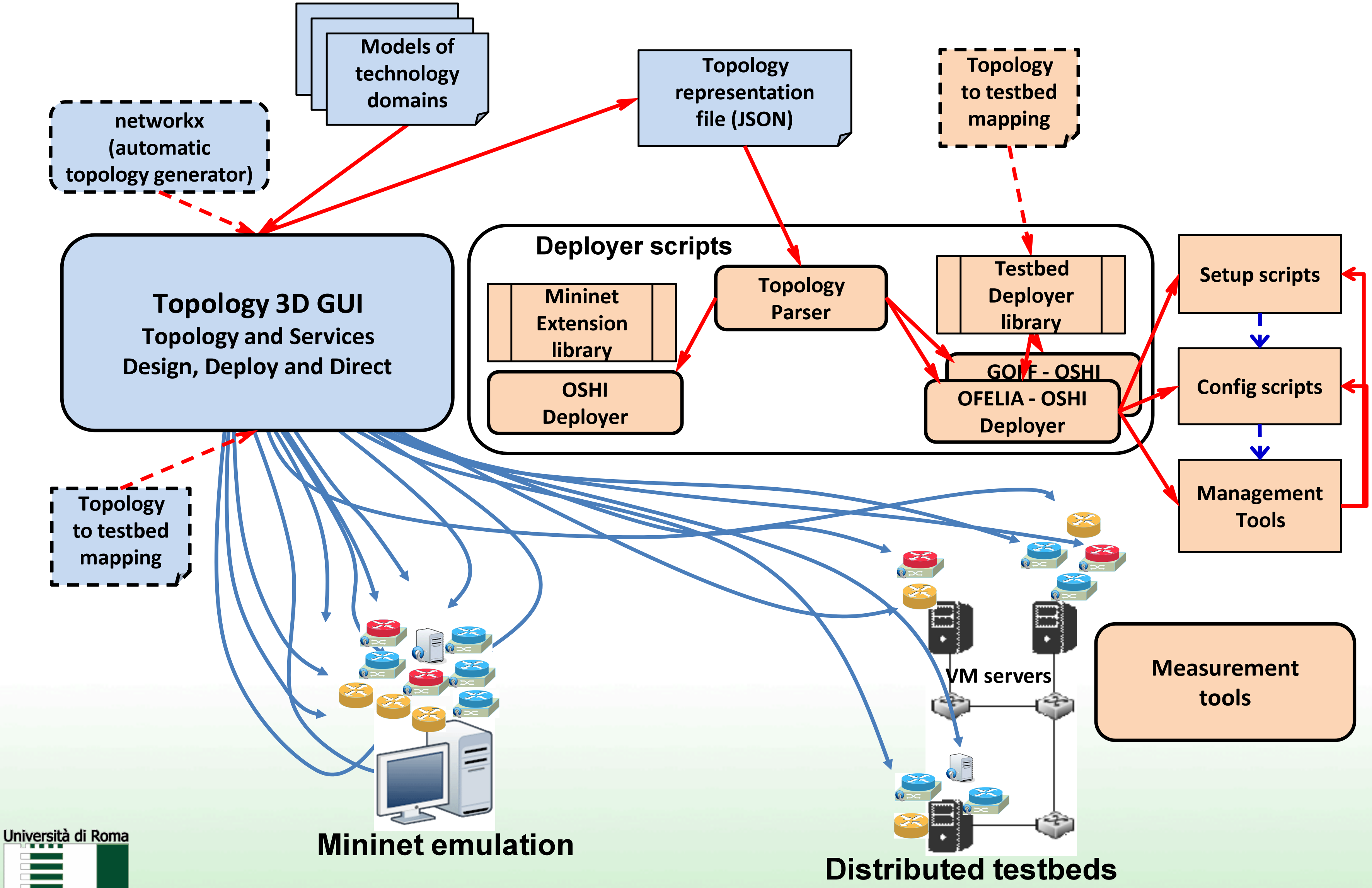


```
cer6 shell
> ip addr

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
```

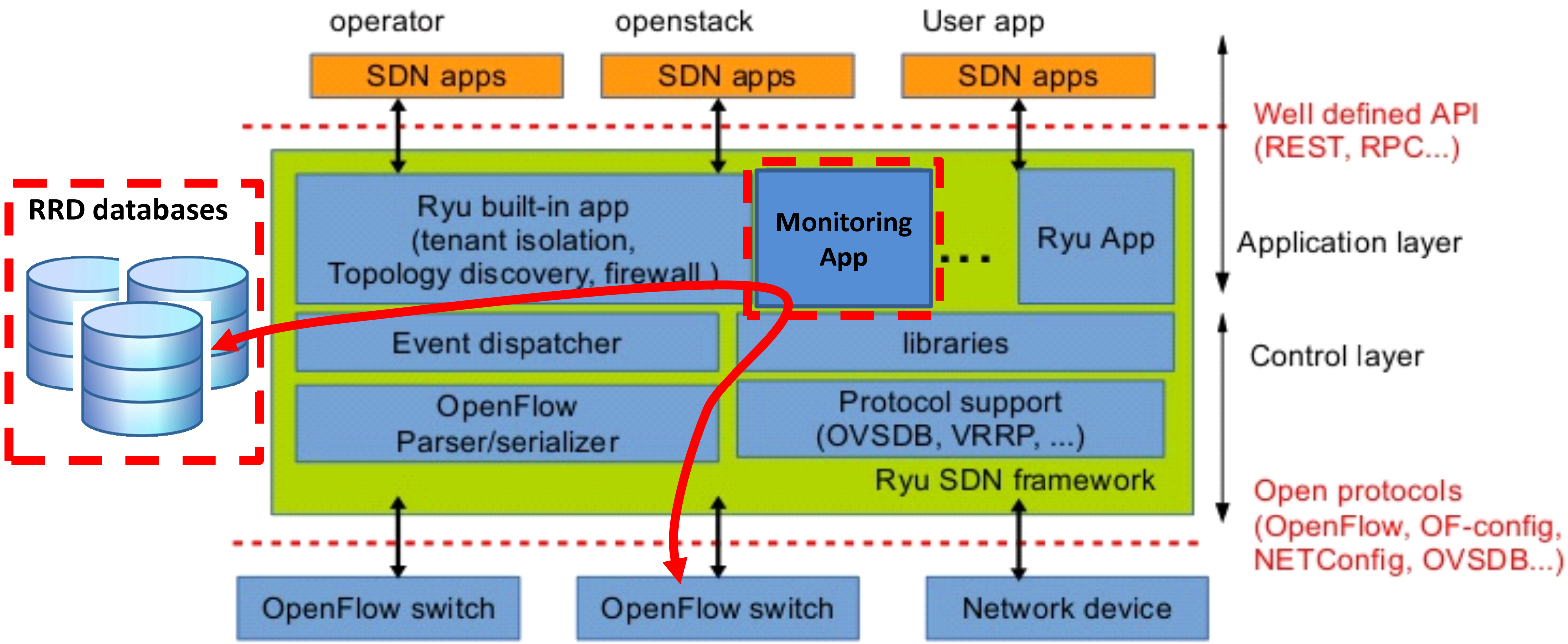
Workflow for the experiments


Models of technology domains
networkx (automatic topology generator)
Topology representation file (JSON)
Topology to testbed mapping
Topology 3D GUI
Topology and Services Design, Deploy and Direct
Deployer scripts
Mininet Extension library
OSHI Deployer
Topology Parser
Testbed Deployer library
GOFF - OSHI
OFELIA - OSHI Deployer
Setup scripts
Config scripts
Management Tools
Topology to testbed mapping
VM servers
Measurement tools
Mininet emulation
Distributed testbeds

Traffic Engineering

- “The goal of TE is to share bandwidth among competing applications, possibly using multiple paths” [1];
- TE steers the traffic across to the backbone in order to obtain the efficient use of available bandwidth in the links [2];
- “Improving user performance and making more efficient use of network resources requires adapting the routing of traffic to the prevailing demands” [3];
- TE is: “the practice of reserving bandwidth for specific workloads and mapping traffic onto particular paths and links in order to optimize network resource allocation and enforce policies” [4];

Monitoring of the OSHI network



- Introduces Monitoring App in the RYU framework
- Leverages on the OpenFlow stats;
- Saves the statistics into RRD databases
- Not yet completed, next step will implement the monitoring GUI and REST interfaces

Flow assignment problem [9][10]

- **Input:**
  - Traffic matrix
  - topology
  - links capacity
- **Output:**
  - minimize average (global) delay
- **Problem variables:**
  - Routes of the flows

$\{\gamma_{ij}\}$ [pack/s] between node i and j

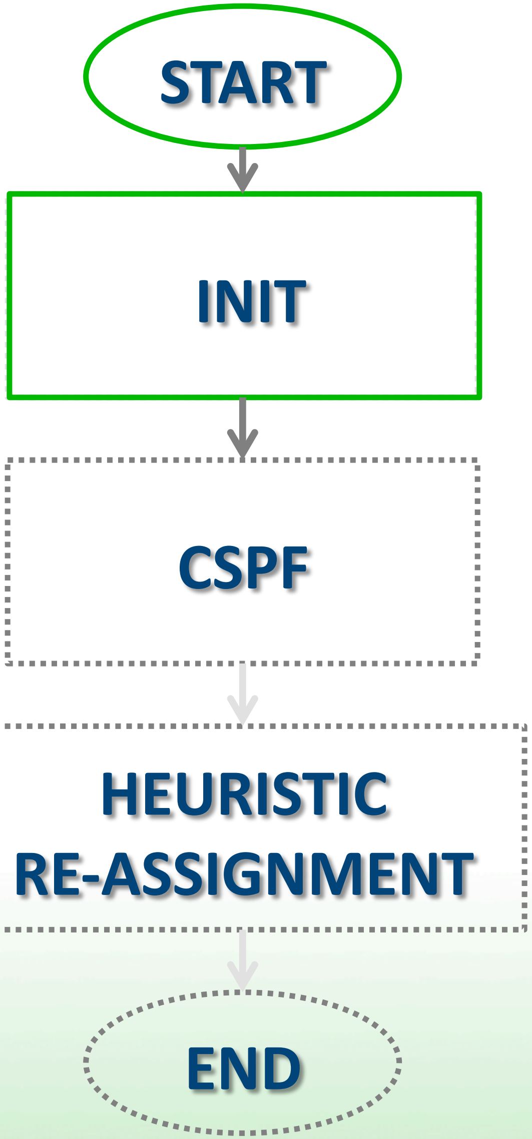
$\{C_{kz}\}$ [bit/s] capacity of the link that interconnects node k and z

T

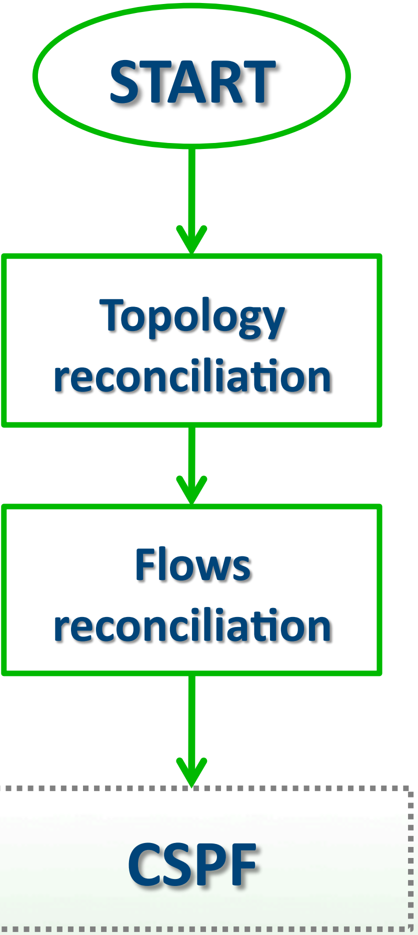
$P(_{ij}) = \{(i,n1),(n1,n2)\ldots (ns,j)\}$
$\Rightarrow \{\lambda_{kz}\}$ [pack/s] load on the link

An heuristic has been implemented, exact solution is computationally complex

The algorithm



Init phase



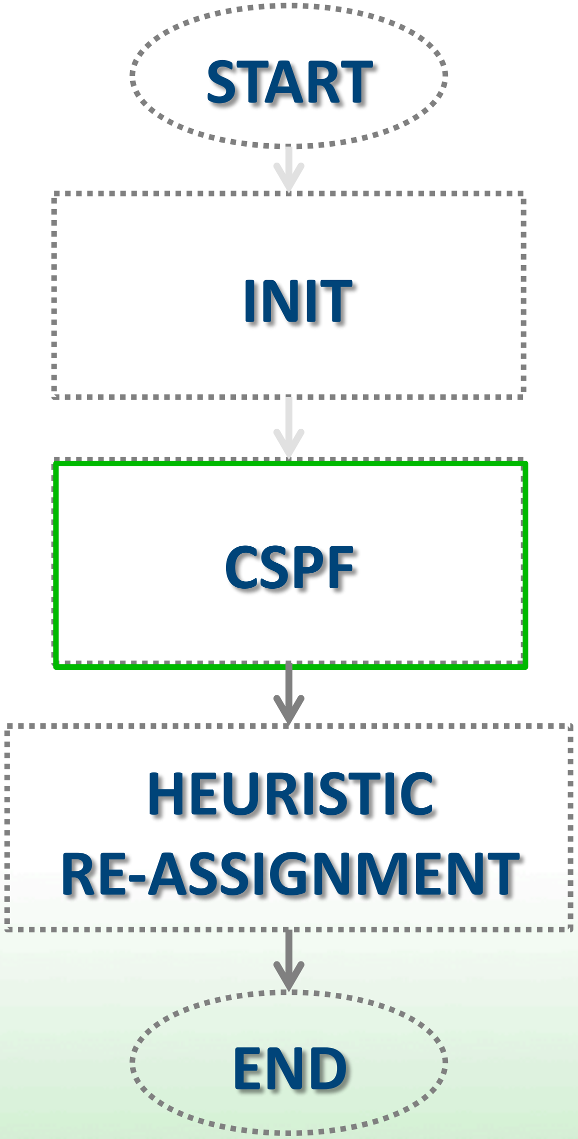
START

Topology reconciliation

Flows reconciliation

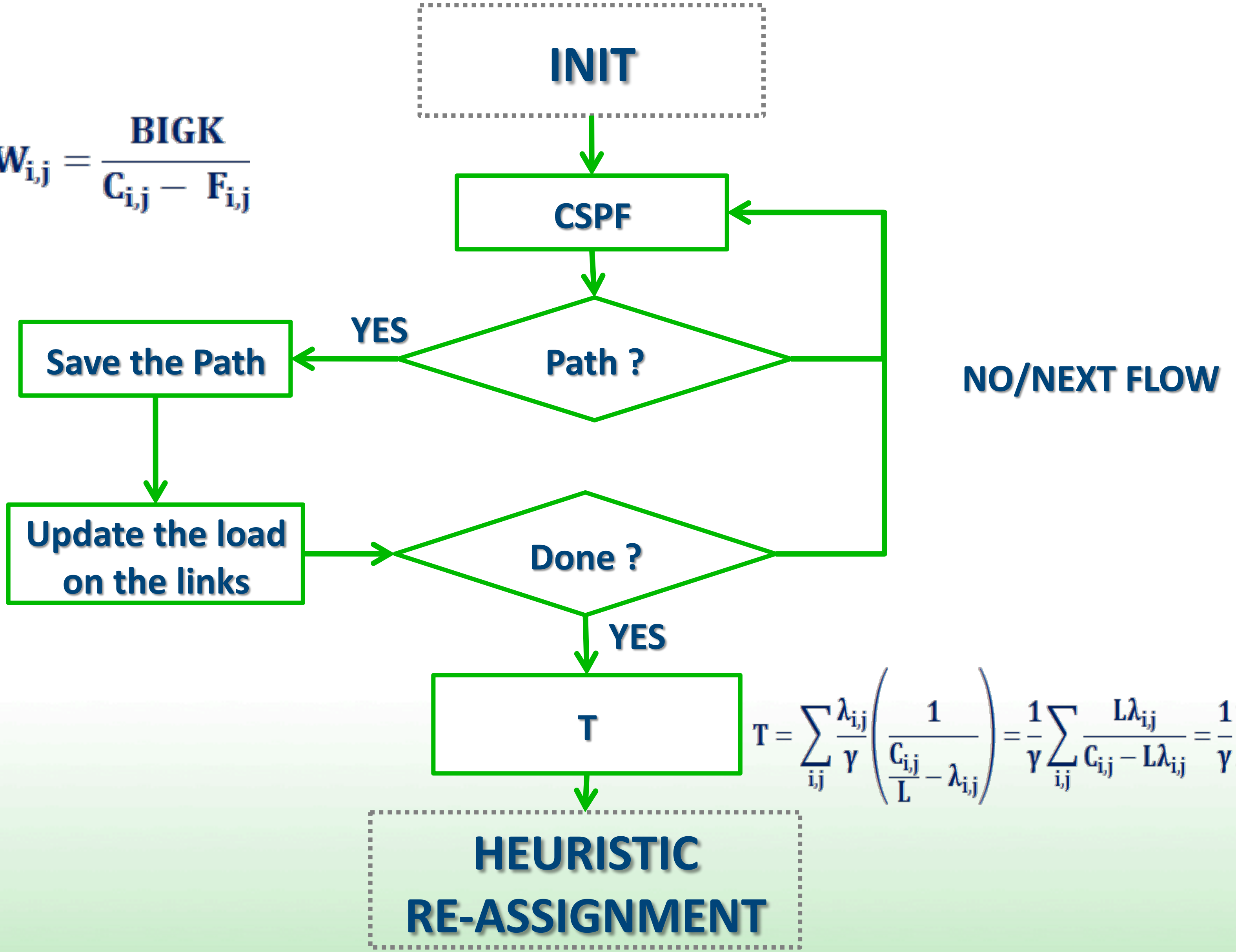
CSPF

The algorithm (2)



CSPF phase

$$W_{i,j} = \frac{BIGK}{C_{i,j} - F_{i,j}}$$



INIT

CSPF

Path ?

YES

Save the Path

Update the load on the links

Done ?

NO/NEXT FLOW

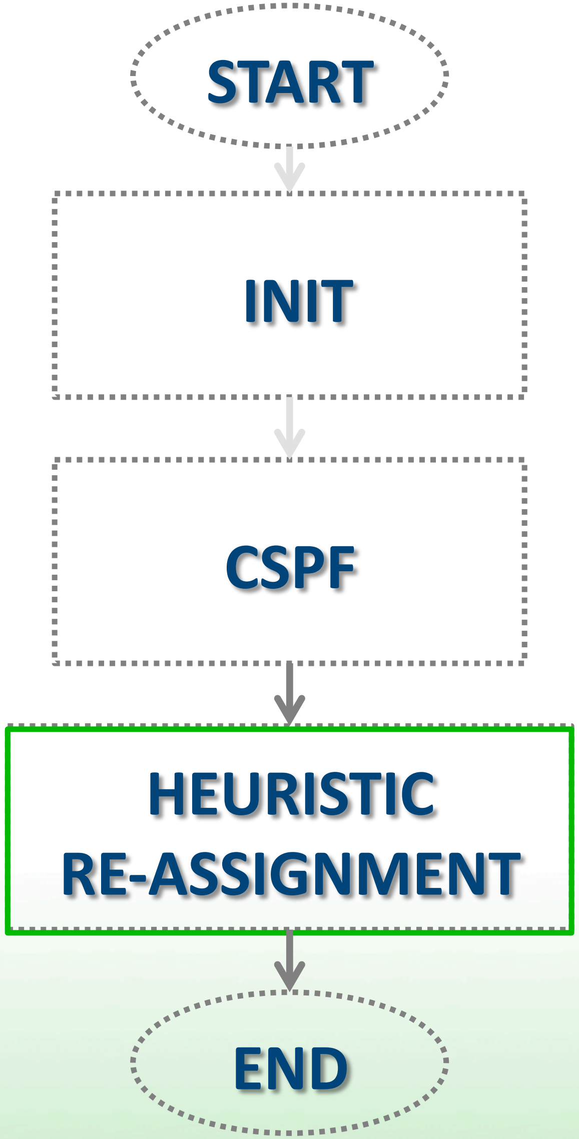
YES

T

HEURISTIC RE-ASSIGNMENT

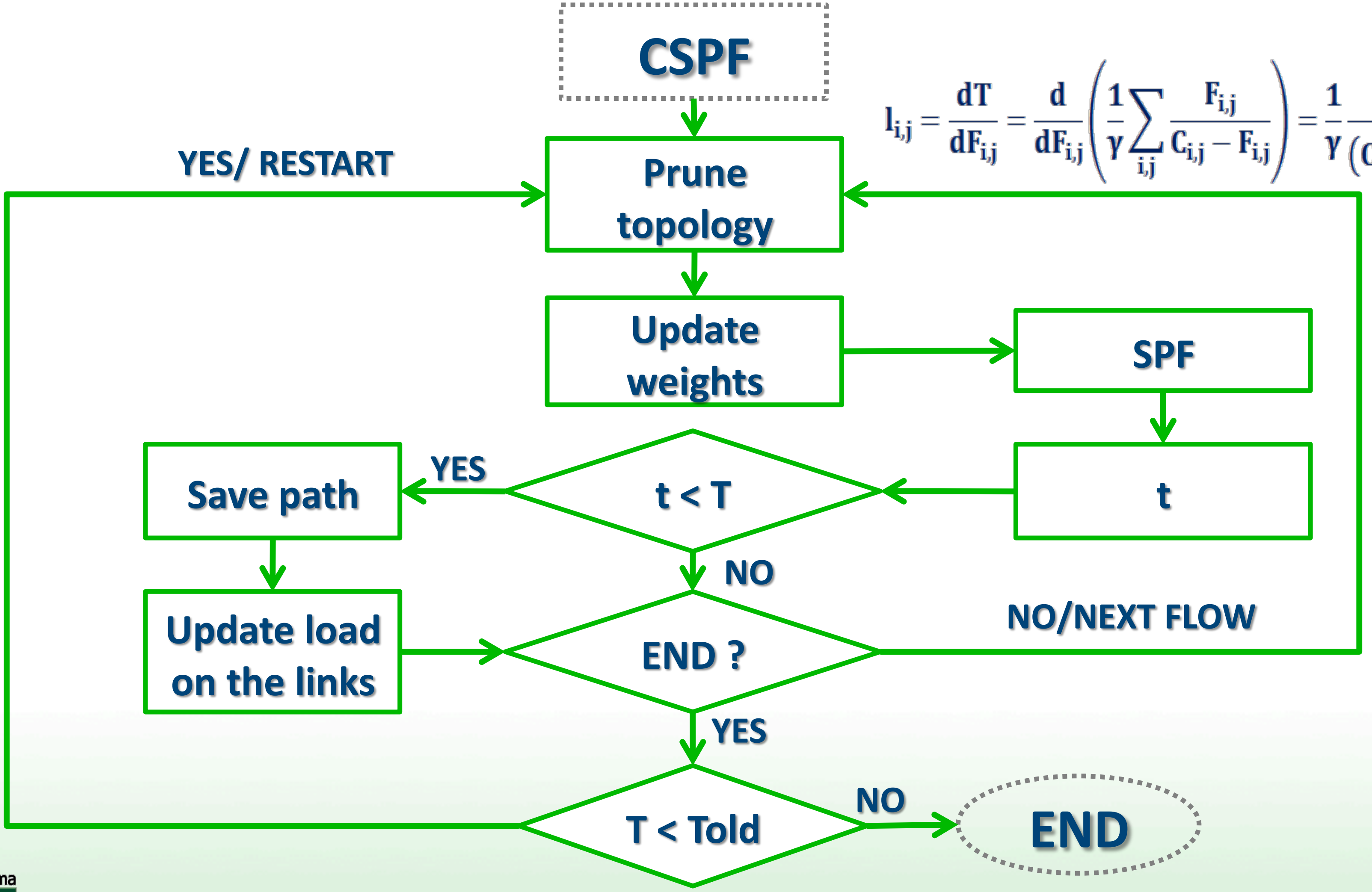
$$T = \sum_{i,j} \frac{\lambda_{i,j}}{\gamma} \left(\frac{1}{\frac{C_{i,j}}{L} - \lambda_{i,j}} \right) = \frac{1}{\gamma} \sum_{i,j} \frac{L\lambda_{i,j}}{C_{i,j} - L\lambda_{i,j}} = \frac{1}{\gamma} \sum_{i,j} \frac{F_{i,j}}{C_{i,j} - F_{i,j}}$$

The algorithm (3)

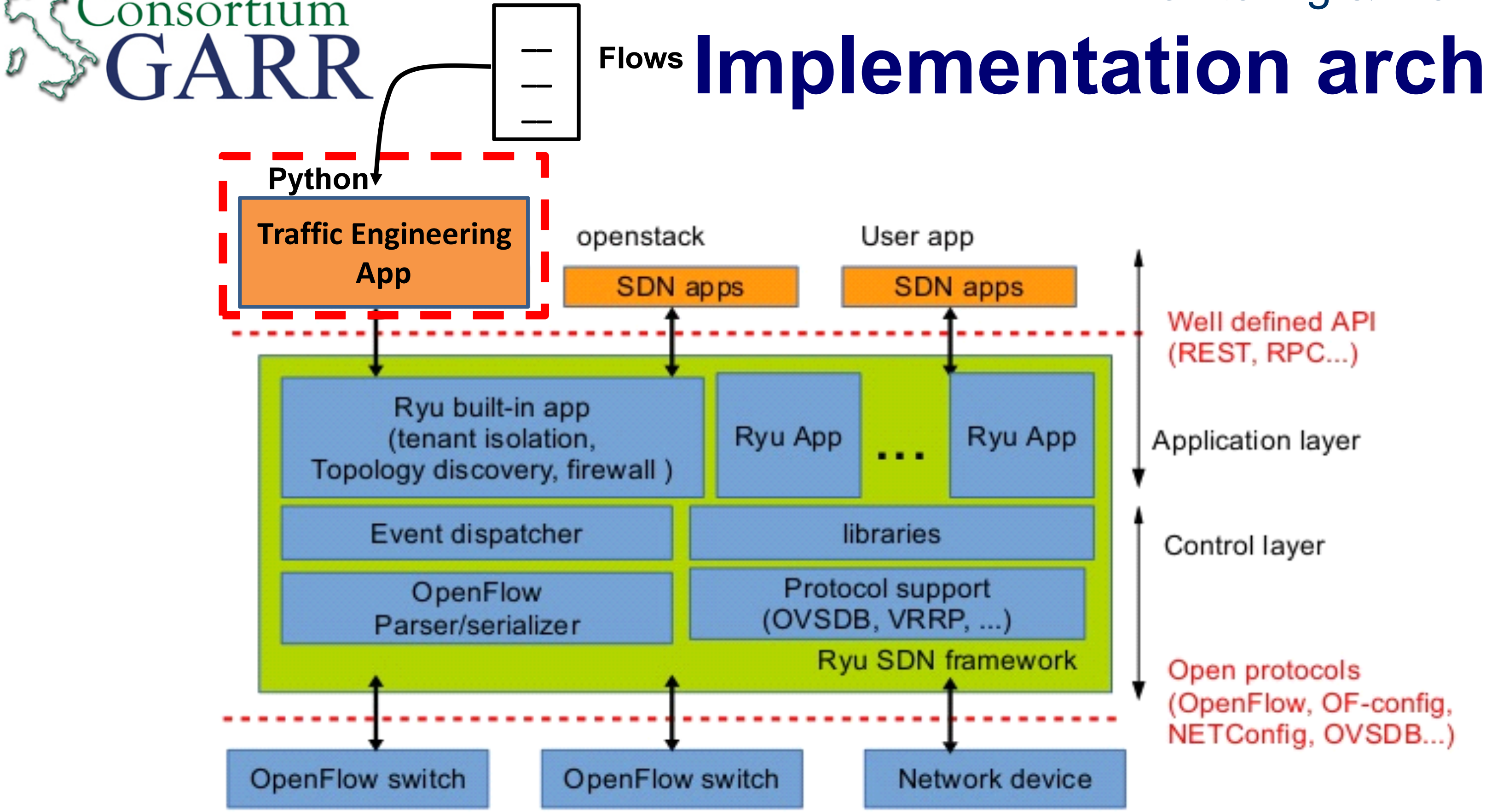


Heuristic re-assignment

$$l_{i,j} = \frac{dT}{dF_{i,j}} = \frac{d}{dF_{i,j}}\left(\frac{1}{\gamma}\sum_{i,j}\frac{F_{i,j}}{C_{i,j}-F_{i,j}}\right) = \frac{1}{\gamma}\frac{C_{i,j}}{\left(C_{i,j}-F_{i,j}\right)^2}$$



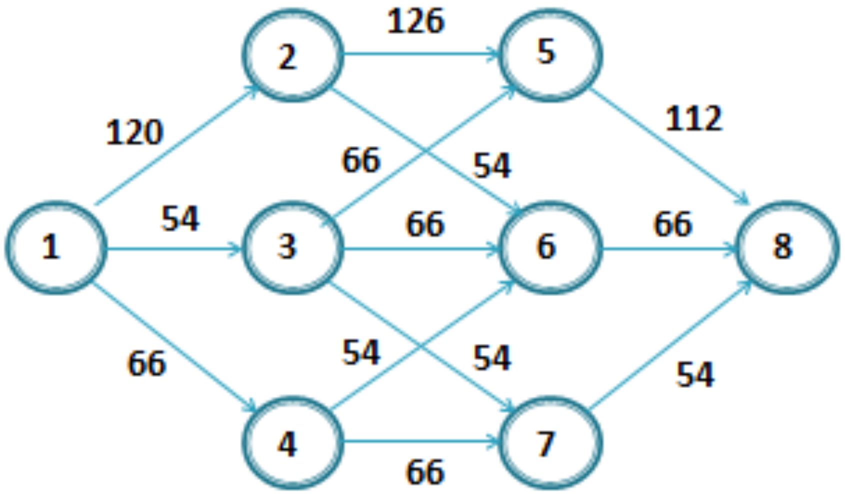
Implementation architecture


Flows
Python
Traffic Engineering App
openstack
User app
SDN apps
SDN apps
Well defined API (REST, RPC...)
Ryu built-in app (tenant isolation, Topology discovery, firewall)
Ryu App
...
Ryu App
Application layer
Event dispatcher
libraries
Control layer
OpenFlow Parser/serializer
Protocol support (OVSDB, VRRP, ...)
Ryu SDN framework
Open protocols (OpenFlow, OF-config, NETConfig, OVSDB...)
OpenFlow switch
OpenFlow switch
Network device

- Traffic Engineering App is a Python App
- Uses the API REST of the Topology module and OFCTL module;
- In the next version will be integrated with the monitoring infrastructure;

Example

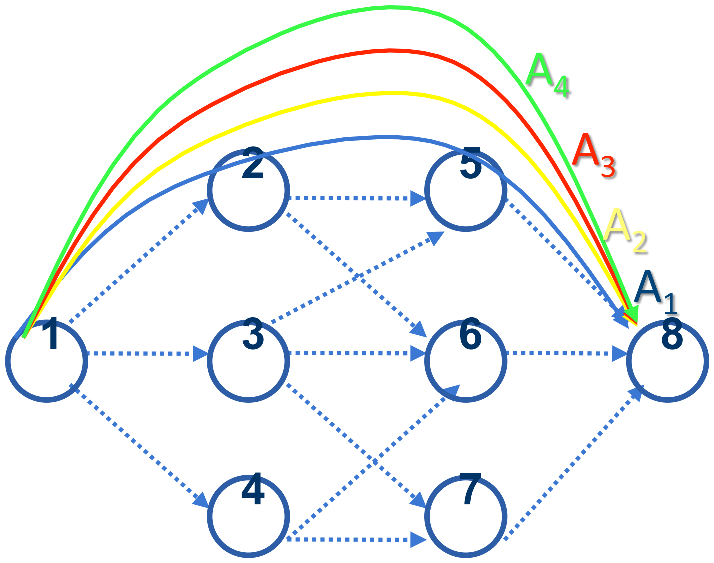
- **Topology and link capacity [kb/s] :**



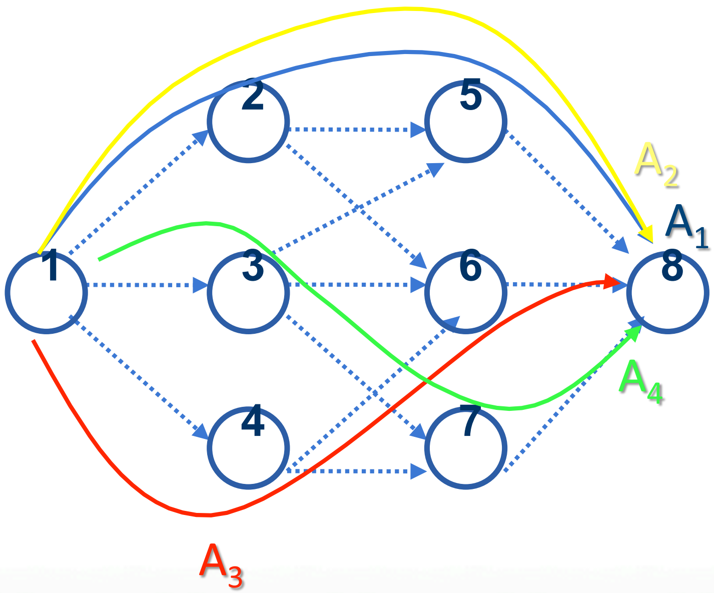
- **Traffic relations A_j (src, dst, rate [kb/s]):**
  - A_1 = (1,8,35) ; A_2 = (1,8,30) ; A_3 = (1,8,23) ; A_4 = (1,8,17) ;

Results

Shortest Path First:

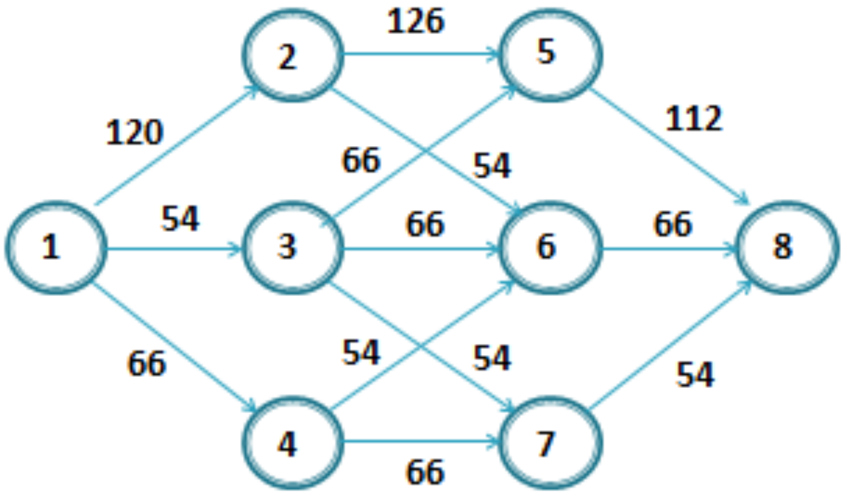


Heuristic re-assignment:



Example (2)

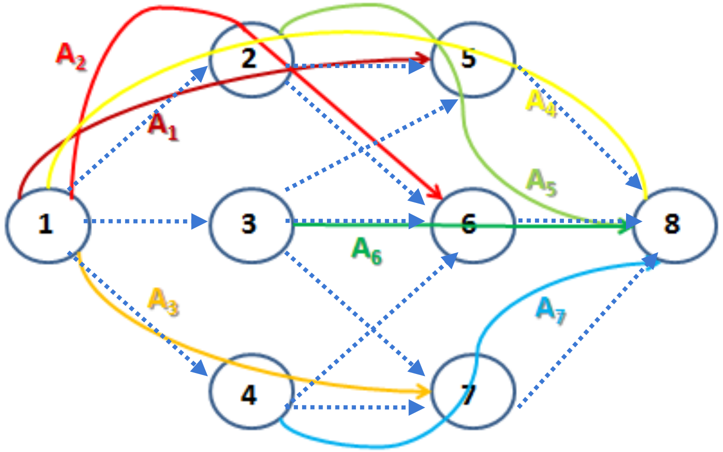
- **Topology and link capacity [kb/s] :**



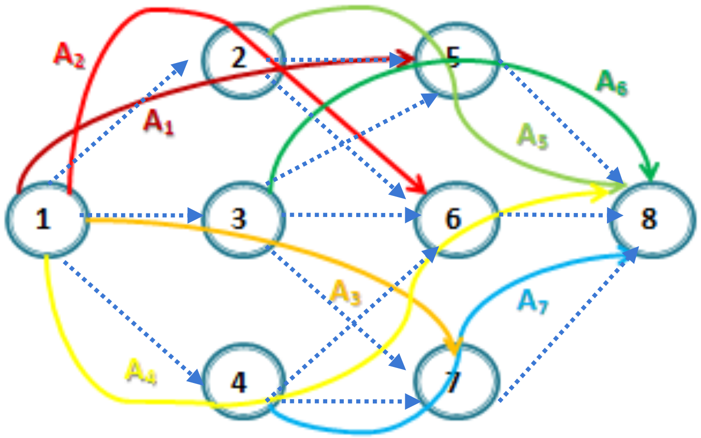
- **Traffic relations A_j (src, dst, rate [kb/s]):**
  - A_1 = (1,5,35) ; A_2 = (1,6,30) ; A_3 = (1,7,23) ; A_4 = (1,8,17) ; A_5 = (2,8,32) ; A_6 = (3,8,26) ; A_7 = (4,8,32) ;

Results (2)

CSPF:



Heuristic re-assignment:



Results (3)

CSPF:

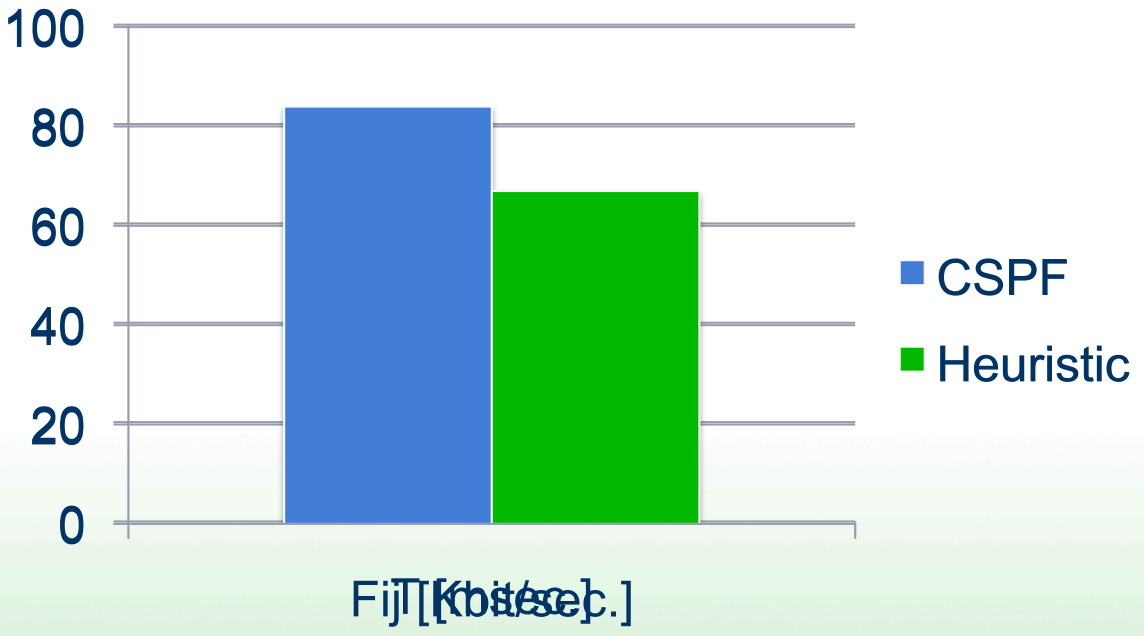
T = 77.30 [msec.]

F_{ij}(MAX)=84 [kbit/sec.]

Heuristic re-assignment:

T = 52.95 [msec.]

F_{ij}(MAX)=67 [kbit/sec.]



Conclusions and future directions

- **We designed and implemented an Open Source Hybrid IP/SDN solution (OSHI):**
  - Hybrid IP/SDN node (Linux based);
  - Network architecture with a set of services;
  - Graphical designer and a deployer for Mininet and distributed SDN testbeds;
  - Monitoring infrastructure;
  - TE in order to improve the developed services;
- **Future works:**
  - Improvement of the Monitoring solution;
  - Integration of the TE app with the Monitoring solution
  - Leveraging of Segment Routing solution (already supported in OSHI);

Publications

- Mauro Campanella, Luca Prete, Pier Luigi Ventre, Matteo Gerola, Elio Salvadori, Michele Santuari, Stefano Salsano, Giuseppe Siracusano “Bridging OpenFlow/SDN with IP/MPLS”, TNC2014, 19 - 22 May 2014, Dublin, Ireland (poster);
- Stefano Salsano, Pier Luigi Ventre, Luca Prete, Giuseppe Siracusano, Matteo Gerola, Elio Salvadori, "OSHI - Open Source Hybrid IP/SDN networking (and its emulation on Mininet and on distributed SDN testbeds)", GTTI 2014, June 19, 2014, Palermo, Italy. **Winner of "Premio Carassa 2014"** for the best paper on the "Networking" topic co-authored and presented by a young researcher (paper);
- S. Salsano, P. L. Ventre, L. Prete, G. Siracusano, M. Gerola, E. Salvadori, “Open Source Hybrid IP/SDN networking (and its emulation on Mininet and on distributed SDN testbeds)”, 3rd European Workshop on Software Defined Networks, EWSDN 2014, 1-3 September 2014, Budapest, Hungary (paper)
- S. Salsano, N. Blefari-Melazzi, F. Lo Presti, G. Siracusano, P. L. Ventre, “Generalized Virtual Networking: an enabler for Service Centric Networking and Network Function Virtualization”, 16th International Telecommunications Network Strategy and Planning Symposium, Networks 2014, 17-19 September 2014, Funchal, Portugal (paper)

Publications (2)

- Matteo Gerola, Michele Santuari, Elio Salvadori, Stefano Salsano, Pier Luigi Ventre, Mauro Campanella, Francesco Lombardo, Giuseppe Siracusano “ICONA: Inter Cluster Onos Network Application” , NetSoft 2015 , 13 – 17 April, London, United Kingdom (demo paper);
- Matteo Gerola, Michele Santuari, Elio Salvadori, Stefano Salsano, Mauro Campanella, Pier Luigi Ventre, Ali Al-Shabibi, William Snow “ICONA: Inter Cluster Onos Network Application”, SOSR 2015, 15 - 18, June, Santa Clara, CA, United States (paper under revision);
- “OSHI - Open Source Hybrid IP/SDN networking and Mantoo - a set of management tools for controlling SDN/NFV experiments” – to be submitted (journal paper);
- “Experimental comparison of caching strategies for ICN over SDN using physical and virtual testbeds” – to be submitted (journal paper);

Thank you! (questions)

References

1. Jain, Sushant, et al. "B4: Experience with a globally-deployed software defined WAN." *ACM SIGCOMM Computer Communication Review*. Vol. 43. No. 4. ACM, 2013.
2. http://www.ciscopress.com/articles/article.asp?p=426640
3. Fortz, Bernard, Jennifer Rexford, and Mikkel Thorup. "Traffic engineering with traditional IP routing protocols." *Communications Magazine, IEEE* 40.10 (2002): 118-124.
4. http://www.packetdesign.com/solutions/traffic-engineering
5. S. Vissicchio et al., “Opportunities and Research Challenges of Hybrid Software Defined Networks”, ACM SIGCOMM Computer Communications Review, Editorial Zone (April 2014).
6. S. Bryant, P. Pate, “Pseudo Wire Emulation Edge-to-Edge (PWE3) Architecture”, IETF RFC 3985
7. K. Kompella, Y. Rekhter , “Virtual Private LAN Service (VPLS) Using BGP for Auto-Discovery and Signaling”, IETF RFC 4761
8. L. Kou et al., “A Fast Algorithm for Steiner Trees”
9. L. Fratta, M. Gerla, L. Kleinrock, The flow deviation method: an approach to store-and-forward communication network design, Network, 3(2):97-133, 1973, John Wiley & Sons
10. M. Gerla, L. Kleinrock, On the topological design of distributed computer networks, IEEE Transactions on Communications, 25(1):48-60, 1977.