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)
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

1. OSHI objectives & architecture

- 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
1. OSHI objectives & architecture

**OSHI architecture**

**Hybrid IP/SDN resilient data plane**

- **Open Source Hybrid IP/SDN (OSHI) nodes**
- **IP Routing Daemon (Quagga)**
- **IP Forwarding Engine (Linux networking)**
- **OF Capable Switch - OFCS (Open vSwitch)**
- Physical interfaces
- Virtual ports

Pier Luigi Ventre - Borsisti Day 2015
Performance evaluation (brief)

1. OSHI objectives & architecture

Effects of larger flow table on the OVS performance

Effects of the kernel cache on the OVS performance
2. Services

Services: Virtual circuits

- **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 [6];
  - EoMPLS tunnel;
  - PW can relay arbitrary layer 2 packets;
  - Not supported by OpenFlow, it has been realized through a GRE tunnel;
2. Services

VLL versus PW (brief)

Performance assessment of PW service
2. Services

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)
3. Experimental tools (Mantoo)

**Workflow for the experiments**

- **Topology 3D GUI**
  - Topology and Services
  - Design, Deploy and Direct

- **Mininet emulation**

- **Distributed testbeds**

- **Networkx** (automatic topology generator)

- **Models of technology domains**

- **Topology representation file (JSON)**

- **Topology to testbed mapping**

- **Deployer scripts**
  - Mininet Extension library
  - Topology Parser
  - OSHI Deployer
  - GOF - OSHI
  - OFELIA - OSHI Deployer

- **Testbed Deployer library**

- **Setup scripts**

- **Config scripts**

- **Management tools**

- **Measurement tools**

---

Pier Luigi Ventre - Borsisti Day 2015
• “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];
4. Monitoring & Traffic Engineering

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

Pier Luigi Ventre - Borsisti Day 2015
Flow assignment problem [9][10]

• **Input:**
  - Traffic matrix
  - topology
  - links capacity

• **Output:**
  - minimize average (global) delay

• **Problem variables:**
  - Routes of the flows
  - $P_{\{ij\}} = \{(i,n1),(n1,n2)\ldots (ns,j)\}$
  - $\lambda_{kz}$ [pack/s] load on the link
  - $T$
  - $\gamma_{ij}$ [pack/s] between node i and j
  - $C_{kz}$ [bit/s] capacity of the link that interconnects node k and z

An heuristic has been implemented, exact solution is computationally complex.
4. Monitoring & Traffic Engineering

The algorithm

START

INIT

CSPF

HEURISTIC RE-ASSIGNMENT

END
4. Monitoring & Traffic Engineering

Init phase

START

Topology reconciliation

Flows reconciliation

CSPF
4. Monitoring & Traffic Engineering

The algorithm (2)

START

INIT

CSPF

HEURISTIC RE-ASSIGNMENT

END
4. Monitoring & Traffic Engineering

CSPF phase

\[ W_{i,j} = \frac{\text{BIGK}}{C_{i,j} - F_{i,j}} \]

INIT

CSPF

YES

Save the Path

Path ?

YES

Update the load on the links

Done ?

YES

T

NO/NEXT FLOW

HEURISTIC RE-ASSIGNMENT
4. Monitoring & Traffic Engineering

The algorithm (3)

START

INIT

CSPF

HEURISTIC RE-ASSIGNMENT

END
4. Monitoring & Traffic Engineering

Heuristic re-assignment

\[ l_{ij} = \frac{dT}{dF_{ij}} = \frac{d}{dF_{ij}} \left( \frac{1}{Y} \sum_{i,j} \frac{F_{ij}}{C_{ij} - F_{ij}} \right) = \frac{1}{Y} \frac{C_{ij}}{(C_{ij} - F_{ij})^2} \]
4. Monitoring & Traffic Engineering

Implementation architecture

- 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;
4. Monitoring & Traffic Engineering

Example

- Topology and link capacity [kb/s]:

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

**Results**

**Shortest Path First:**

**Heuristic re-assignment:**
4. Monitoring & Traffic Engineering

Example (2)

- Topology and link capacity [kb/s]:

![Network Diagram]

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

Results (2)

CSPF:

Heuristic re-assignment:
4. Monitoring & Traffic Engineering

Results (3)

CSPF:
\[ T = 77.30 \text{ [msec.]} \]
\[ F_{ij} \text{(MAX)} = 84 \text{ [kbit/sec.]} \]

Heuristic re-assignment:
\[ T = 52.95 \text{ [msec.]} \]
\[ F_{ij} \text{(MAX)} = 67 \text{ [kbit/sec.]} \]
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);
5. Conclusions and future directions

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)


6. S. Bryant, P. Pate, “Pseudo Wire Emulation Edge-to-Edge (PWE3) Architecture”, IETF RFC 3985


8. L. Kou et al., “A Fast Algorithm for Steiner Trees”
