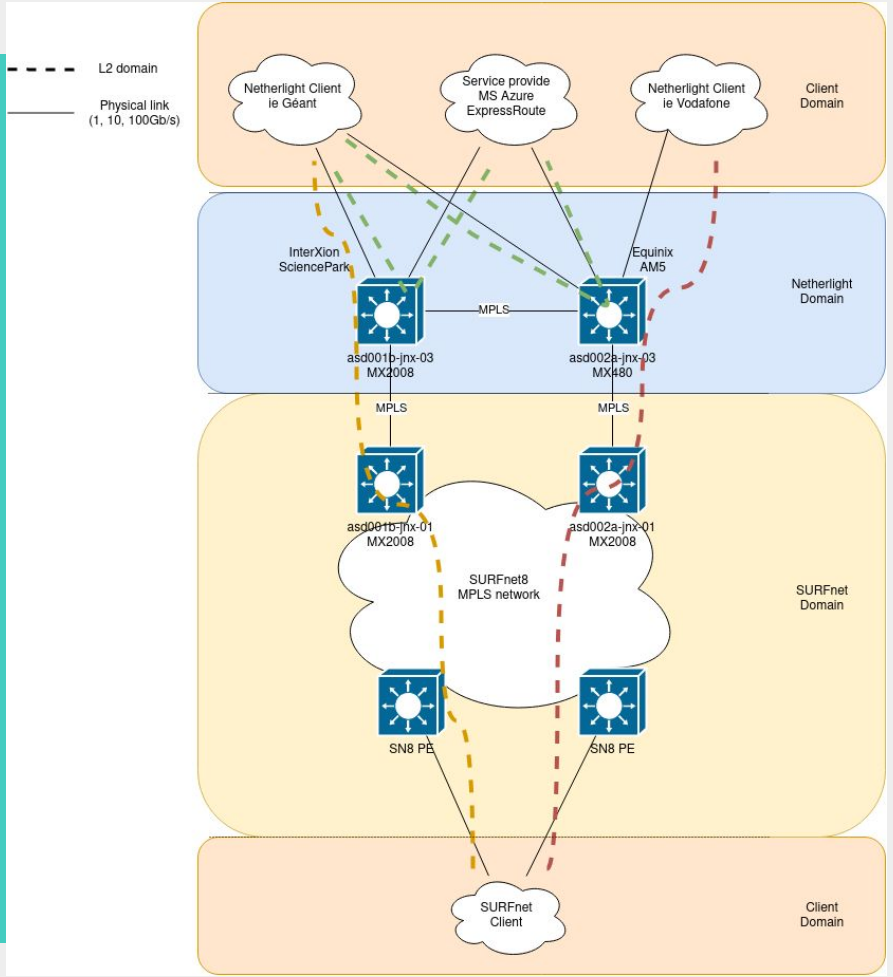
Investigative Research for an IP Peering Service for NetherLight

Assessor: Cees de Laat
Supervisors:
Gerben van Malenstein
Migiel de Vos
Max Mudde

Research Project 2 - #100
Arnold Buntsma
Mar Badias Simó

NetherLight: open lightpath exchange

- Built and operated by **SURFnet**
- **High bandwidth P2P & multipoint connections** for ~70 clients
- Their clients are **research and education networks** and **service providers** that want to connect among them



NetherLight investigates offering a new service

- **Peering Service**
- Common **layer 2 domain** for several clients
- To allow their clients to set up **BGP peering**
- Similar to an **Internet eXchange Point**

RESEARCH QUESTION

How can NetherLight facilitate a state-of-the-art peering service which is flexible, secure, manageable and has a uniform setup?

- Requirements
- Options & Best practices
- Protocol behaviour
- On-boarding procedure

Methodology

1. Set requirements

2. Contact IXPs

3. Study literature

4. Research solutions

5. Compare solutions

6. Recommend

Requirements

- A detailed explanation of the service
- Uniform onboarding process
- Well-manageable, Secure & Scalable
  - *Uniform*
  - *Spoofing & Hijacking*
  - *Hundreds of clients*
- At least one of the solutions can be implemented on the current platform

Interviews & Literature

- Most of **peering services** of IXPs built on top of **VPLS**, some **EVPN**
- Broadcast traffic is a problem: **ARP storms**
- Protect the peering platform: **control the types of traffic** going on the network
- **Prevent propagation of wrong routing information**

Generic Components for all solutions

Route Server

- Scaling
  - BGP sessions
- Manageability
  - Uniform peering relations
  - Ability to block prefixes
- Security
  - Filtered Routes
  - RPKI validation

Security

- MANRS[2]
- 1 MAC & IP per interface
- Whitelist EtherTypes

IP Space

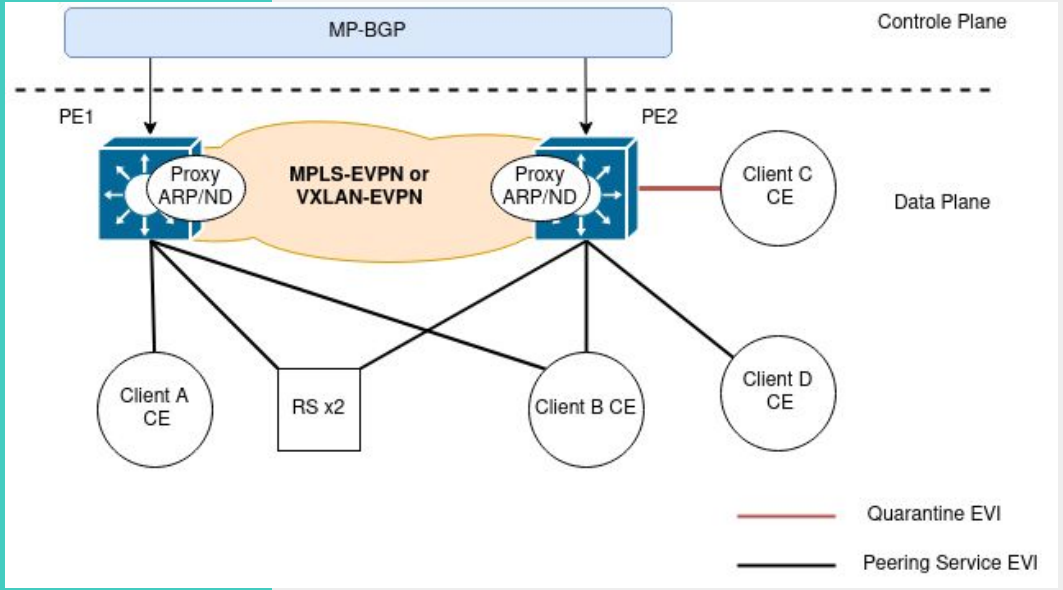
- IPv4 /24 (x2)
- IPv6 /64

[2] https://www.manrs.org/ixps/

SOLUTIONS 1.1 & 1.2: MPLS-EVPN & VXLAN-EVPN

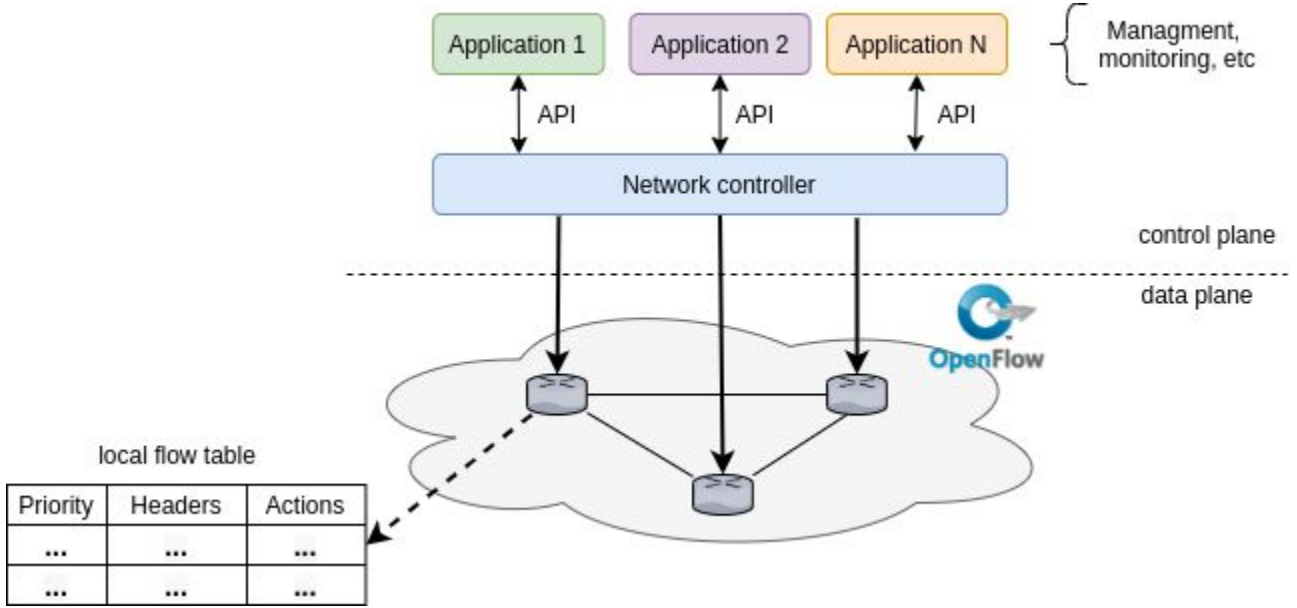
EVPN Solutions

- VXLAN-EVPN vs MPLS-EVPN
- Quarantine EVI
- Single VLAN
- Management via Orchestration and Automation tools
  - Cisco NSO
- Monitoring
  - SNMP
  - sFlow
- Also includes Generic Components



SOLUTION 2: SDN / OpenFlow

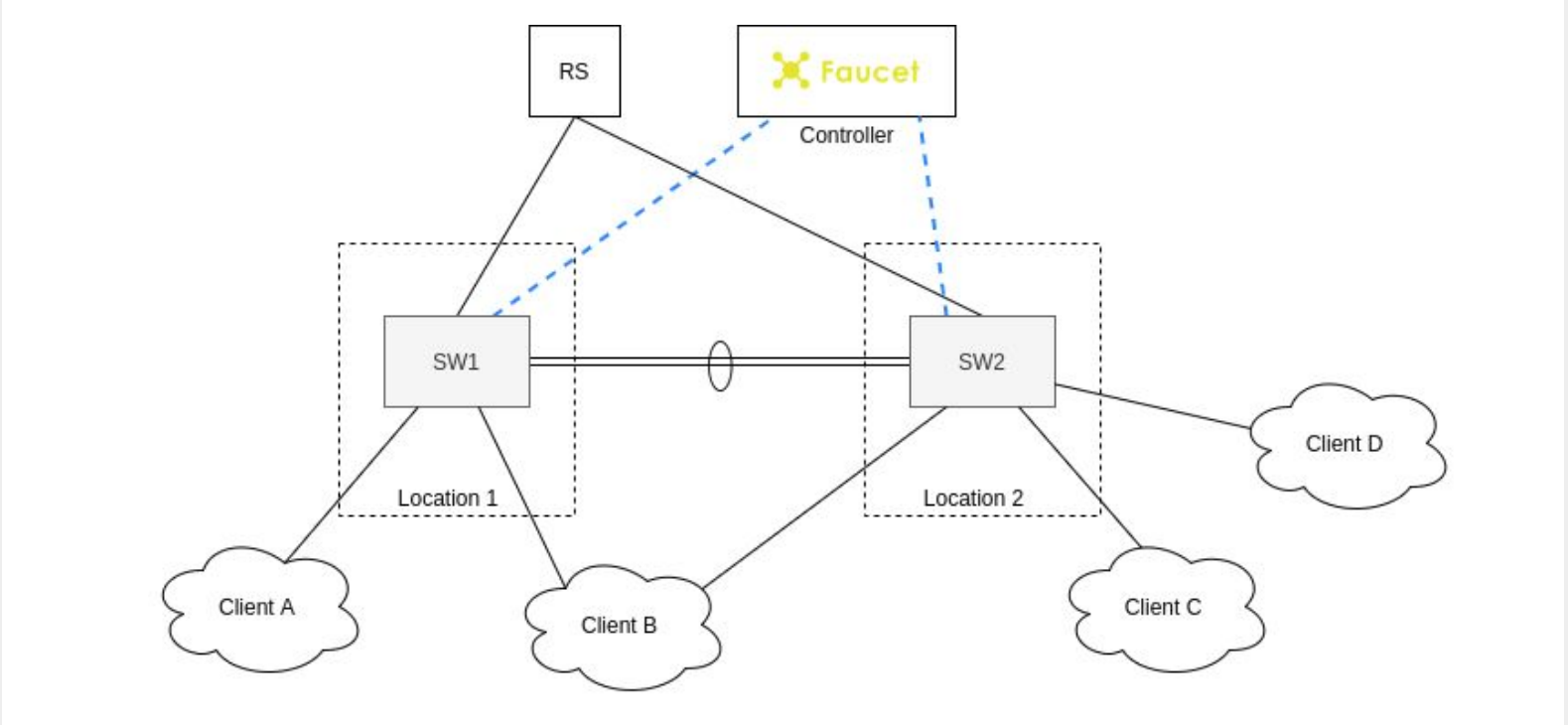
OpenFlow


Application 1
Application 2
Application N
Managment, monitoring, etc
API
API
API
Network controller
control plane
data plane
OpenFlow
local flow table
Priority
Headers
Actions
...
...
...
...
...
...

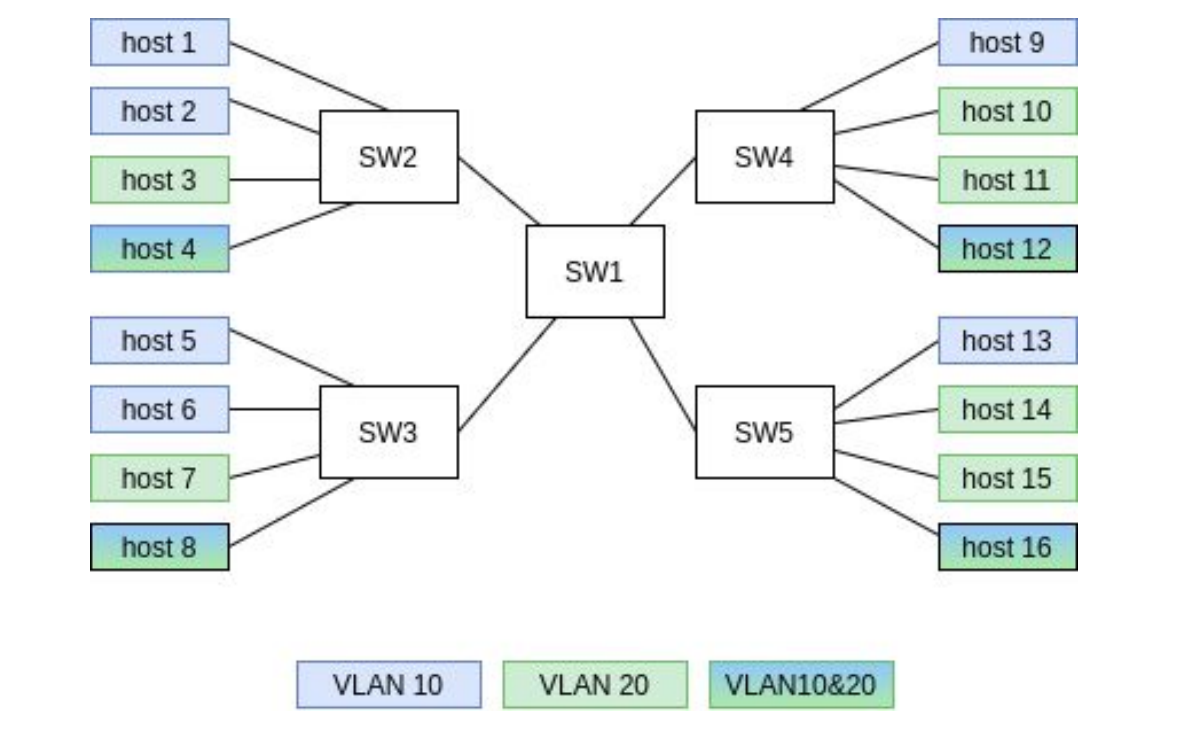
Benefits of OpenFlow

- Following the directives of **Umbrella rule set**
- **Fine-grained control capabilities,** can provide high responsiveness
- Easy network **management**
- We consider **NetherLight an ideal place to innovate**
- **Offers solutions** to peering services known problems

OpenFlow Implementation



Testing Faucet on Mininet



https://github.com/Reseach-Project-2/testfaucet

Programming the service

- Programmed based on **Umbrella rule set**
- A VLAN can be created and retagging frames is possible
- Fine-grained traffic control. **Drop anything that does not match the rules**
- **No quarantine VLAN/EVI** needed
- MAC address known in advance: **elimination of ARP storms**

Peering service with OpenFlow

Monitoring

sFlow or
Gauge+Faucet

Management

Adapting IXP Manager or
developing a new tool

Scalability

Theoretically,
highly scalable

IXP MANAGER

On- and off-boarding workflow

The client provides:

- Desired bandwidth
- Location
- MAC address(es)
- AS number(s)

NL Provides:

- VID
- IP addresses
- ASN of RS
- Configuration template

➔ Off-boarding procedure is more simple :)

Comparison: EVPN vs OpenFlow

EVPN vs OpenFlow results

	Scalability	Manageability	Security	Implementation (for NL)
MPLS-EVPN	++	+	++	+
VXLAN-EVPN	++	+	++	−
OpenFlow	++	++	++	−−

Table 1: Comparison of peering service solutions

Scalable: At least hundreds of clients. No hard limit.
Management: Clients use the service in a uniform way. Configuration errors should be eliminated and minimal management effort needed from the NL team.
Security: Clients unable to interfere with connections of other clients by for example MAC/IP spoofing and BGP hijacking.

Discussion & Conclusion

To date, NetherLight can best create a peering service by adopting the first solution (MPLS-EVPN).

As a more advanced solution over time, NetherLight should consider implementing the second solution proposed (OpenFlow) because of less management effort, fine-grained control of traffic, and vendor independency.

Future Work

- First (small) implementation of MPLS-EVPN solution
- PoC of OpenFlow solution
    - OpenFlow scalability research in production
- Research the ability to use Umbrella rule set in other OpenFlow controllers

Questions?

To date, NetherLight can best create a peering service by adopting the first solution (MPLS-EVPN).

As a more advanced solution over time, NetherLight should consider implementing the second solution proposed (OpenFlow) because of less management effort, fine-grained control of traffic, and vendor independency.

Route Servers

- Scaling
  - BGP sessions
- Manageability
  - Uniform peering relations
  - Ability to block prefixes
- Security
  - Filtered Routes
  - RPKI validation

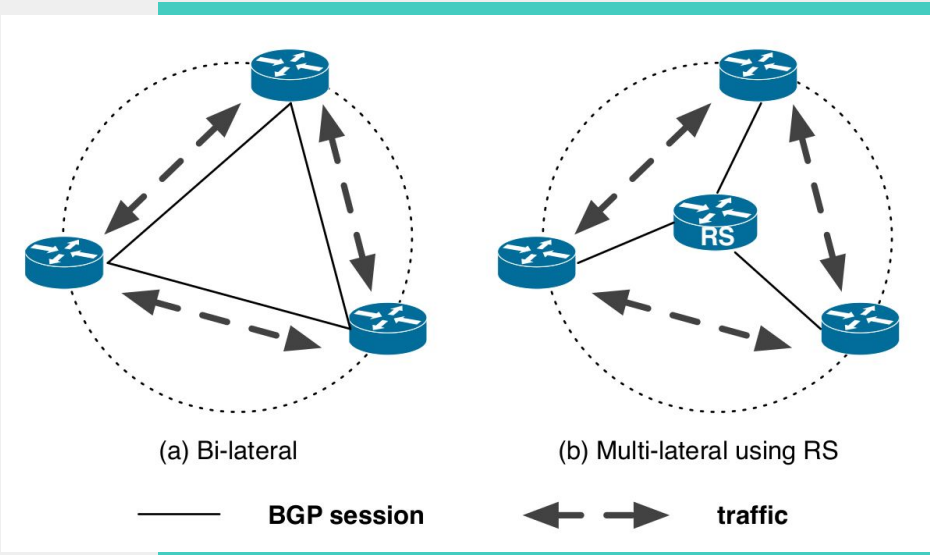


Fig. 1 Peering options (Richter, P et al. 2014)

Faucet multi table