Overlay Networks in Cloud Data Centers

Dino LOPEZ PACHECO

dino.lopez@univ-cotedazur.fr

Network Virtualization

- Network virtualization enables multi-tenancy in Cloud environments
  - Isolation
  - Gives control of the network to the tenant
    - Addressing
    - Routing
    - Partitioning
  - Flexibility
- Network virtualization is common in current Cloud
  - Overlay network to form any network topology

Network Virtualization at Different Levels

- Network device virtualization
  - Virtual switches, virtual routers
- Network Interface virtualization
  - Software only: veth
  - Hardware assisted: SR-IOV, MR-IOV
  - Link aggregation -> multiple links seen as one channel
- Other common virtualization techniques to create overlay networks
  - Application Layer -> Load balancers
  - Network Layer -> IP-in-IP, GRE
  - Link Layer -> VLAN, VxLAN, STT

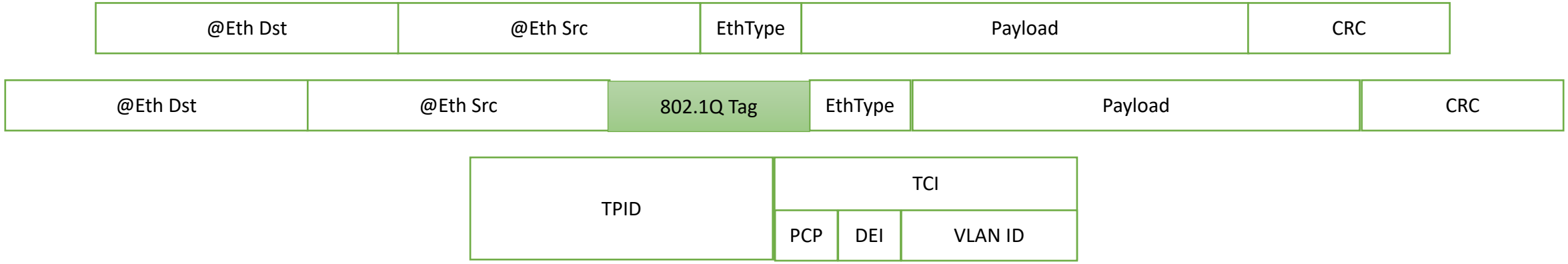
Some examples, non-exhaustive list

IEEE 802.1Q (Virtual LANs)

- IEEE 802.1Q offers support for virtual LANs in a 802.3 Ethernet network
- VLAN tagging (4 bytes)
  - Modify bridges and switches default forwarding mechanisms
  - Increases frame size

- Logical group of network devices
- Break broadcast domains
  - Subnets
- Isolation

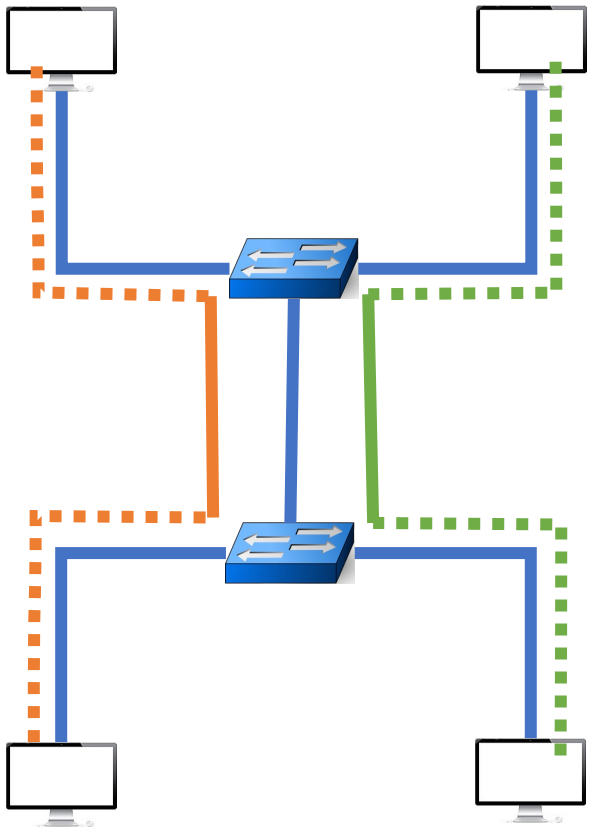
VLAN tag



- Tag Protocol Identifier (TPID – 16 bits)
  - 0x8100
- Tag Control Information (TCI – 16 bits)
  - Priority Code Point (PCP – 3 bits)
  - Drop Eligible Indicator (DEI – 1 bit)
  - VLAN ID (VID – 12 bits): 0x0000 – 0xFFF

- VLAN Tag – 4 bytes
- Minimum MTU – 64 bytes
  - 68 bytes with a VLAN tag
- Maximum MTU – 1518 bytes
  - 1522 bytes with a VLAN tag – jumbo frames

Common VLAN configuration



- Untagged (access) ports.
  - Receive untagged frames and add a VLAN tag
  - Strips the VLAN tag before sending a frame
- Tagged (trunk) ports.
  - Receive/send tagged frames (useful for switch interconnexion)
  - One single port can support multiple VLAN IDs

Additional considerations

- Frequent default VLAN ID: 1
- Native VLAN
  - Special VLAN whose frames will travel untagged between trunk ports
  - Usually, VLAN 1
  - VLAN Hopping Attack
- VLAN stacking – 802.1ad
  - One VLAN tag precedes another one

VLAN interconnection

One Router – Multiple Links

- One link per VLAN
- Easy but needs as many links/ports as VLANs available
- Router is connected to access ports

One Router – One Link

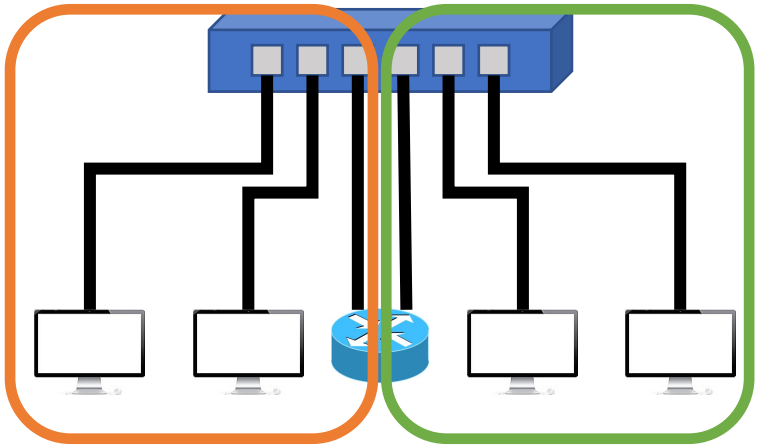
- Only one link
  - Router is connected to a trunk port
- Create as many sub interfaces as VLANs

MultiLayer Switch

- No need of extra router
- One switch having multiples ports on different VLANs can route packets between VLANs
  - *Layer 3 switch*

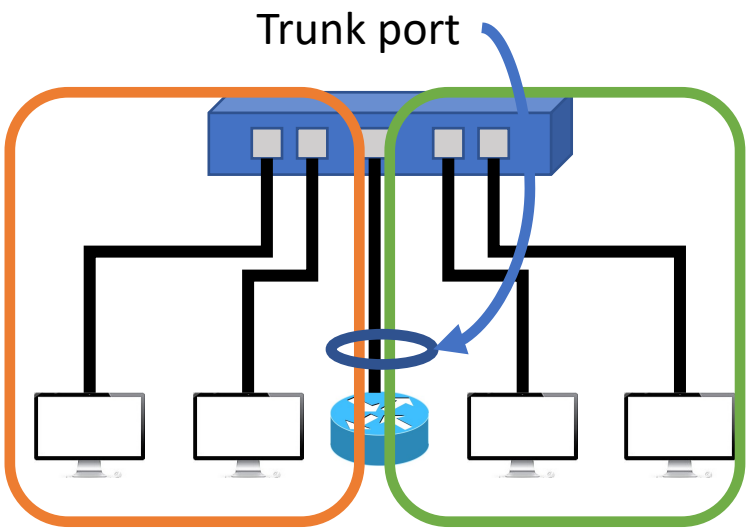
VLAN Interconnection

One Router – Multiple Links



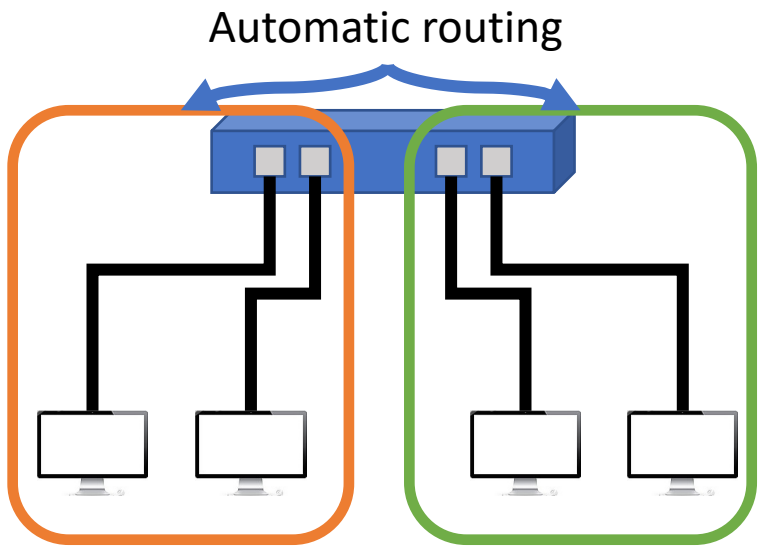
VLAN10 VLAN50

One Router – One Link



VLAN10 VLAN50

MultiLayer Switch



VLAN10 VLAN50

VLAN in OpenVSwitch and Linux

OpenVSwitch

- Set an access port
  - ovs-vsctl set port eth0 tag=10
    - Access port for VLAN 10
  - ovs-vsctl set port eth0 vlan_mode=access
    - Access port of native VLAN
- Set a trunk port
  - All interfaces are by default trunk ports
- Specify VLANs going through a trunk port
  - ovs-vsctl set port eth0 trunks=10,100
- vlan_mode
  - access, trunk, native-tagged, native-untagged, dot1q-tunnel

Linux

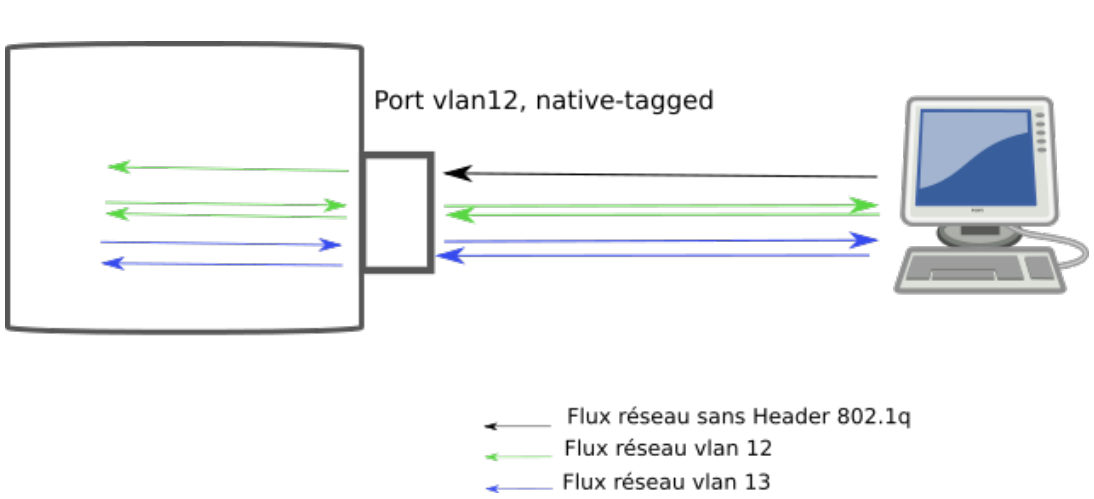
- All interfaces are a trunk port by default
  - Accept tagged and untagged frames
- VLAN tags are not stripped
- To strip a specific VLAN tag
  - Create an access port through a subinterface
  - Assign an IP address if needed
  - Enable the subinterface
- Exemple
  - ip link add link eth0 name eth0.2 type vlan id 2
  - ip a a 192.168.1.10/24 dev eth0.2
  - ip link set eth0.2 up

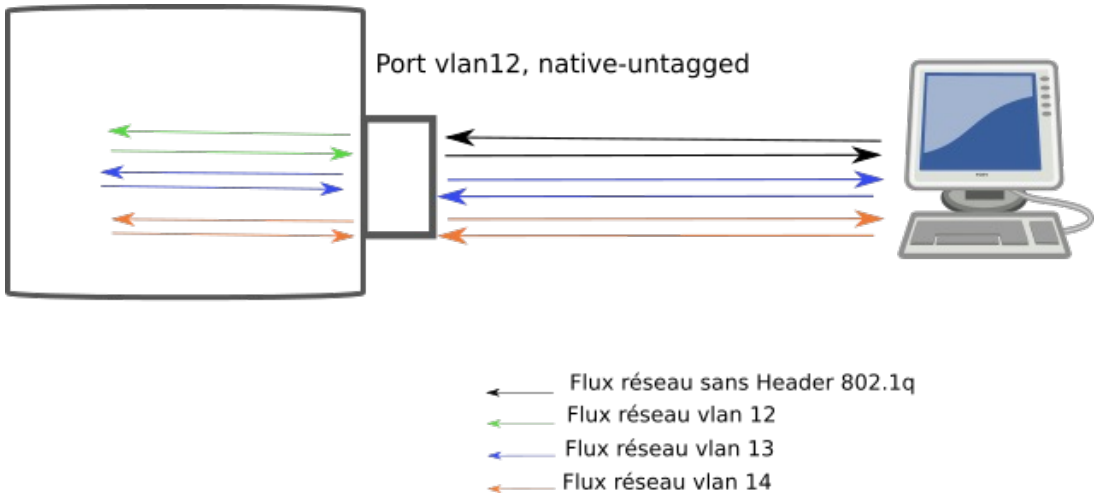
native-tagged vs native-untagged

(from https://mesangebleue.github.io/OpenVSwitch_CheatSheet/#le-param%C3%A8tre-vlan_mode)

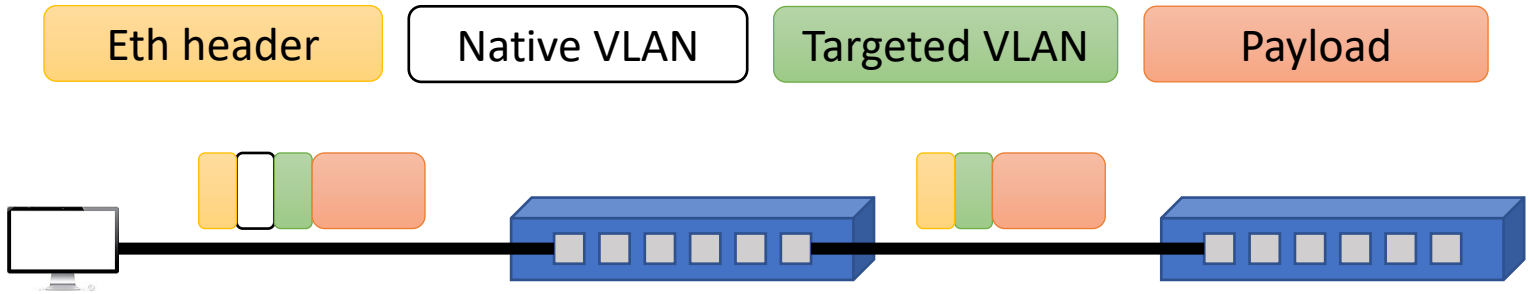
```
ovs-vsctl set port vnet0 tag=12
ovs-vsctl set port vnet0 trunks=13
ovs-vsctl set port vnet0 vlan_mode=native-tagged
```

```
ovs-vsctl set port vnet0 tag=12
ovs-vsctl set port vnet0 trunks=13,14
ovs-vsctl set port vnet0 vlan_mode=native-untagged
```





VLAN Hopping Attack – The double tagging technique



- Condition
  - Ingress port accepting tagged frames
  - Egress port able to forward in untagged form (native VLAN)