universidade de aveiro
theoria poiesis praxis

Arquitetura de Comunicações

VyOS Initial Setup

After the first boot of **each VyOS device**, load the default configuration and reboot:

```
sudo cp /opt/vyatta/etc/config.boot.default /config/config.boot
reboot
```

Check network interface names: `ip addr`

To change the keyboard layout: `set console keymap`

Use the following commands to verify configuration:

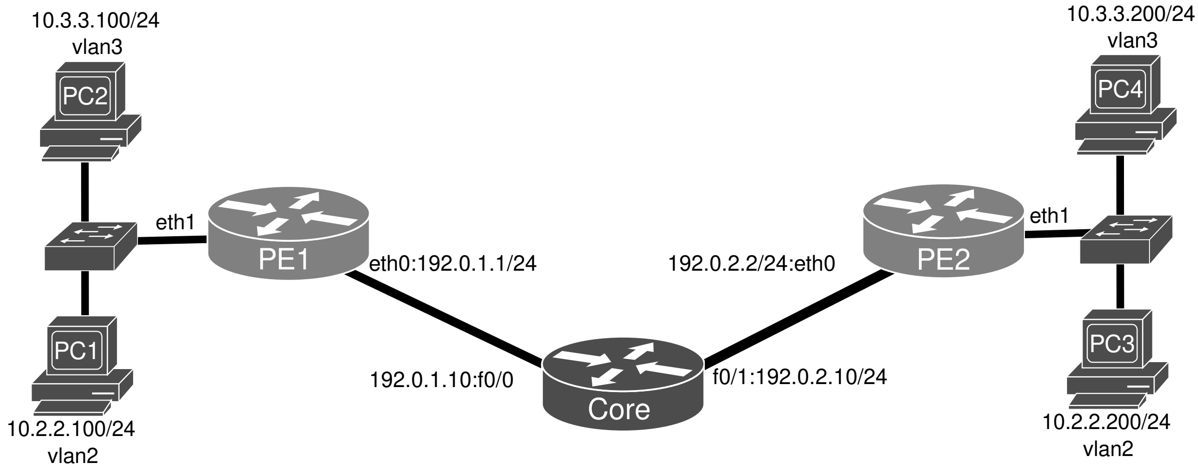
```
show configuration
show configuration commands
```

For QEMU GNS3 template use the following parameters: RAM: 512M, Console type: telnet (or none with auto start console checked), HDD Disk interface: ide, Network Adapters: 6, Network Name format: eth{0}.

For VirtualBox GNS3 template use the following parameters: RAM: 512M, Console type: telnet (or none with auto start console checked), Network Adapters: 4, Network Name format: eth{0}, check Network option "Allow GNS3 to use any ... adapter".

VyOS user guide: https://docs.vyos.io/en/latest/

VXLAN



1. The Core router should be a Cisco device and Routers PE1 and PE2 should be a VyOS device. Configure all IPv4 addresses and OSPF protocol at the core connections. For PE1:

```
PE1$ configure
PE1# set interfaces ethernet eth0 address 192.0.1.1/24
PE1# set protocols ospf area 0 network 192.0.1.0/24
PE1# set system host-name PE1
PE1# commit
PE1# save
```

Make a similar configuration in PE2.

>> Verify the IPv4 routing table at the PE routers (`$ show ip route`).

>> Verify the connectivity between PE1 and PE2.

2. Configure the PC IPv4 addresses (without gateway) and the Layer 2 switches. PC1 and PC3 belong to VLAN2 and PC2 and PC4 belong to VLAN3, configure the respective VLAN ports at the switches.

The connection between the Layer 2 switches and the PE routers should be a 802.1Q trunk. In the switches, configure a trunk port (dot1q) to connect the switches to the PE routers. At the PE routers configure sub-interfaces for VLAN 2 and VLAN 3. For PE1:

```
PE1$ configure
PE1# set interfaces ethernet eth1 vif 2
PE1# set interfaces ethernet eth1 vif 3
PE1# commit
PE1# save
```

Make an equal configuration in PE2.

3. Create two VXLAN connections between the PE. For PE1:

```
PE1# set interfaces vxlan vxlan102 vni 102
PE1# set interfaces vxlan vxlan102 mtu 1500
PE1# set interface vxlan vxlan102 remote 192.0.2.2
PE1# set interfaces vxlan vxlan103 vni 103
PE1# set interfaces vxlan vxlan103 mtu 1500
PE1# set interface vxlan vxlan103 remote 192.0.2.2
PE1# commit
```

Make a similar configuration in PE2.

Create two virtual bridges and add to each one the respective VXLAN interface and Ethernet sub-interface. For PE1:

```
PE1# set interfaces bridge br102 member interface 'eth1.2'
PE1# set interfaces bridge br102 member interface 'vxlan102'
PE1# set interfaces bridge br103 member interface 'eth1.3'
PE1# set interfaces bridge br103 member interface 'vxlan103
PE1# commit
```

Make an equal configuration in PE2.

Start a packet capture in one of the core links. Test the connectivity between the PC of the same VLAN.

>> Analyze capture packets.

>> Explain how the packets are being sent from PE1 to PE2 (and vice-versa).

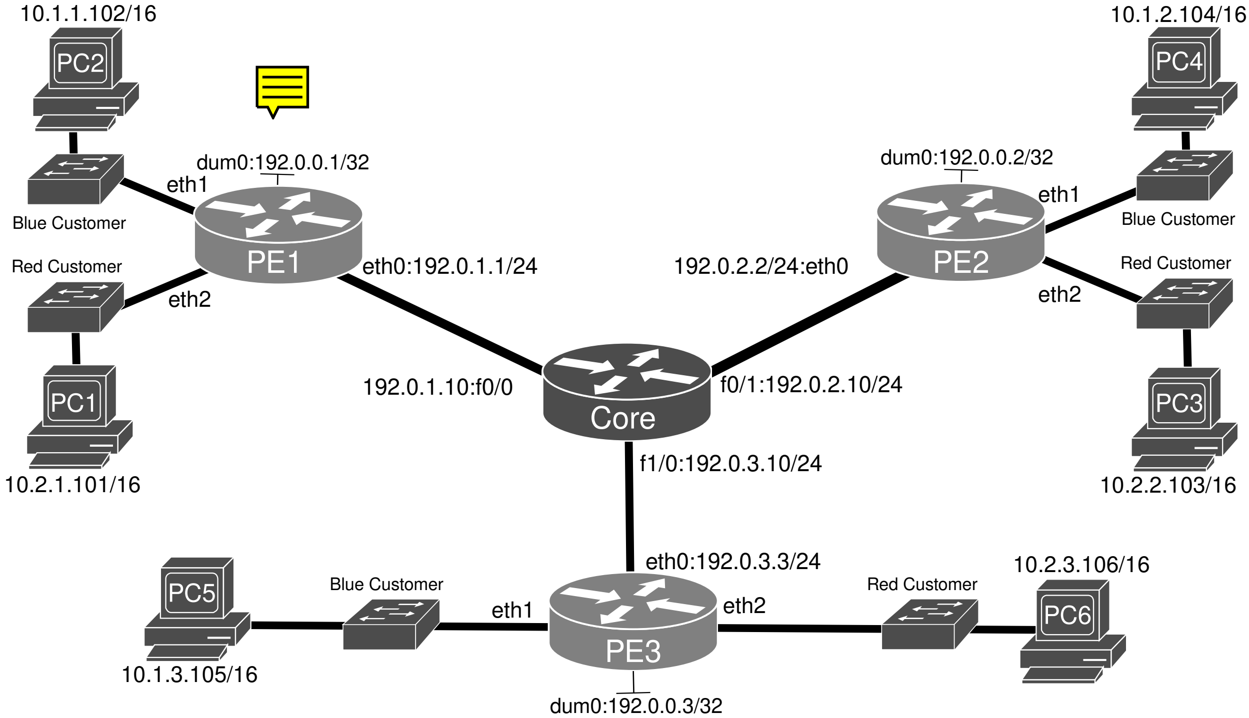
>> Explain how the source VLAN (VXLAN connection) is identified.

>> Discuss the limitations of the VXLAN connection when more than 2 remote sites exist.

Note: the default VXLAN UDP port in VyOS is not the default 4789 but 8479. To make Wireshark decode the VXLAN packets, select one of the UDP packets to/from port 8479, right-click and choose “Decode As…”, change port value to 8479, and change “Current” decode from “(none)” to “VXLAN¨, save and press OK.

Note2: The virtual bridge interfaces can have IPv4 address and act as gateways of the VLAN.

L2VPN/EVPN with VXLAN transport



4. Configure all IPv4 addresses (including the dummy interfaces) and OSPF at the core connections for all devices. To configure a loopback/dummy interface:

```
PE1# set interfaces dummy dum0 address 192.0.0.1/32
```

>> Test the connectivity between all PE routers loopback/dummy addresses.

5. Configure a Layer2 VPN between the customer networks for both customers, using a BGP EVPN with VXLAN transport. Consider a Spine-Leaf relation between the PE routers where PE1 is the Spine. Consider internal BGP relations within the same AS with the usage of Route Reflectors.

For PE1 (Spine - Route Reflector):

```
PE1# set protocols bgp system-as 100
PE1# set protocols bgp address-family l2vpn-evpn advertise-all-vni
PE1# set protocols bgp parameters router-id 192.0.0.1
PE1# set protocols bgp neighbor 192.0.0.2 peer-group evpn
PE1# set protocols bgp neighbor 192.0.0.3 peer-group evpn
PE1# set protocols bgp peer-group evpn update-source dum0
PE1# set protocols bgp peer-group evpn remote-as 100
PE1# set protocols bgp peer-group evpn address-family l2vpn-evpn nexthop-self
PE1# set protocols bgp peer-group evpn address-family l2vpn-evpn route-reflector-client
```

For PE2 (Leaf - Route Reflector client):

```
PE2# set protocols bgp system-as 100
PE2# set protocols bgp address-family l2vpn-evpn advertise-all-vni
PE2# set protocols bgp parameters router-id 192.0.0.2
PE2# set protocols bgp neighbor 192.0.0.1 peer-group evpn
PE2# set protocols bgp peer-group evpn update-source dum0
PE2# set protocols bgp peer-group evpn remote-as 100
PE2# set protocols bgp peer-group evpn address-family l2vpn-evpn nexthop-self
```

Make similar configurations for PE3, just changing the router-id.

Analyze the BGP neighbors status and BGP table of each PE router, and identify the learned EVPN type-3 and type-2 prefixes:

```
$ show bgp neighbors
$ show bgp l2vpn evpn
```

>> Explain the advantages of a BGP Spine-Leaf architecture with Route Reflectors versus a full BGP mesh.

6. Configure the VXLAN and bridge interfaces for each customer. For PE1:

```
PE1# set interfaces vxlan vxlan101 source-address 192.0.0.1
PE1# set interfaces vxlan vxlan101 vni 101
PE1# set interfaces vxlan vxlan101 mtu 1500
PE1# set interfaces vxlan vxlan102 source-address 192.0.0.1
PE1# set interfaces vxlan vxlan102 vni 102
PE1# set interfaces vxlan vxlan102 mtu 1500
PE1# set interfaces bridge br101 address 10.1.1.1/16
PE1# set interfaces bridge br101 description 'customer blue'
PE1# set interfaces bridge br101 member interface eth1
PE1# set interfaces bridge br101 member interface vxlan101
PE1# set interfaces bridge br102 address 10.2.1.1/16
PE1# set interfaces bridge br102 description 'customer red'
PE1# set interfaces bridge br102 member interface eth2
PE1# set interfaces bridge br102 member interface vxlan102
```

Make similar configurations for PE2 and PE3.

>> Explain why now the VXLAN connections do not have a remote address defined.

7. Start packet captures in the three core links. At PE1 router restart the BGP process:

```
PE1$ restart bgp
```

>> Analyze the captured EVPN BGP UPDATE packets.

>> Explain how EVPN type-3 and type-2 prefixes are exchanged.

8. Analyze the BGP table of each PE router, and identify the learned EVPN type-3 and type-2 prefixes:

```
$ show bgp l2vpn evpn
```

In order to each PE router learn the MAC addresses of the terminals in each network, ping from each terminal the respective PE (gateway). Identify, after each ping, in the BGP table of each PE router the learned EVPN type-2 prefixes.

>> Explain how and when EVPN type-2 prefixes, related to the customer terminals, are exchanged.

>> Explain why after some time EVPN type-2 prefixes are withdrawn from the BGP table.

>> Explain why and when ARP packets are transported between sites (based on learned Type-2 routes).

>> Explain how the client data packets are transported and differentiated between the customer networks.

(Optional) 9. Change the Spine-Leaf BGP relations setup from a single AS with Route Reflector to a setup with no Route Reflectors. Consider the usage of private AS within the network: (i) all Leaf devices in a single private AS and (ii) each Leaf device with its individual private AS.