Routerlab

Summer semester 2018

Worksheet 2
Group 08

Valentin Franck, 364066
Nikhil Singh, 387694

Pages: 12

Submission Date: May 10, 2018

Question 1

1a

1) One motivation for using VLANs is to split a broadcast domain and thus reduce broadcast traffic. Iff many hosts share a physical network VLANs can logically separate these networks, thus reducing broadcast traffic in both networks and helping with scalability. Especially in large networks of at least a few thousand hosts (where protocols like DHCP and ARP are used for broadcasting messages) this helps reducing networking overhead .
2) VLANs enable support of quality of service. Separate classes of hosts can be assigned different IP prefixes and routers (and switches) can forward traffic with differing priorities depending on the corresponding IP prefixes.

The main motivations of using VLAN's are network segmentation and to reduce network congestion. VLAN's separates the physical network logically into multiple logical networks to allow different departments of the same organization talk to each other without interfering others by forming different broadcast domains.

1b

Port-based VLANs use dedicated physical links, i.e. one network cable is dedicated to one VLAN only. An example is a host connected to an access port of a switch. However, by using VLAN tagging frames can be marked and switches are able to distinguish VLANs, even if these frames are sent over a shared physical link. These shared links are called trunk ports. (They usually connect two switches, that understand the tagging and can forward the frames correctly.)

Port based VLAN's uses access ports which are assigned to a single VLAN to connect to the end host. However, VLAN Trunks uses tags like 802.1q. It receives traffic from several VLANs which means several VLAN's can use the same physical link to connect to different switches and routers. 802.1q is the mechanism used to distinguish between Ethernet frames of different VLAN's on a trunk. It tags the VLAN id in the header and untag it in the receiving side.

1c

The 802.1q protocol uses 12 bits to identify VLANs. Therefore $2^1 2 = 4096$ bits are available. However, the IDs 0 and 4095 are reserved, so the maximum number is 4094 VLANs on a shared physical network. Often switches can only support 300-500 VLANs due to a lack of memory.

1d

If networks are separated, VLAN IDs can be reused. This is an example mentioned in the survey paper in order to deal with a shortage of VLANs. It is crucial, though, that the separation of is preserved as further networking devices are added.
Another possibility is two join different VLANs into a common VLAN. In such case routers will have to deal with the task of differentiating between different groups of hosts, based on their IP addresses.

As per survey on VLAN's provided on ISIS. Yes there are two ways to overcome the limits of 802.1q. By putting many groups in the same VLAN. Reusing the limited VLAN identifiers These are not possible because it makes the configuration quite complex.

1e

One problem in LANs can be ARP spoofing, which hosts use to listen in on other hosts legitimate traffic. This can be mitigated by VLANs, if access control policies are enforced that only allow certain hosts to join critical VLANs. However, it is required to really have a mechanism to authenticate hosts, that try to join the network. This will also make IP address spoofing harder, because certain IP addresses can only exist in certain VLANs.
Access policies are more easily realized, because traffic between different VLANs must pass a router, that can drop traffic from unauthorized sources (acting as a firewall).

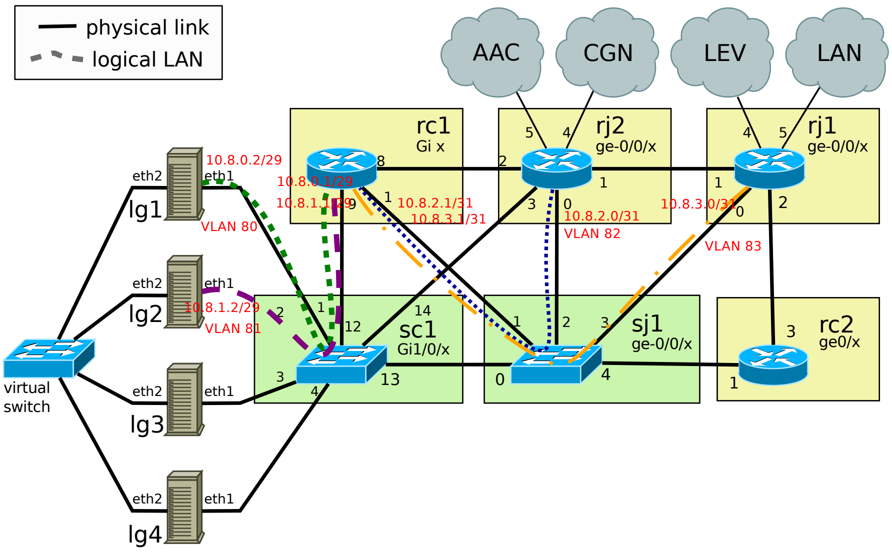

Figure 1: The IPv4 topology.

The use of VLANs reduces broadcast messages in the whole network and therefore makes it harder to carry out DoS-attacks based on broadcasting messages.
Due to VLANs host mobility is enabled. If a host changes its access point it can remain in the same VLAN and thus does not have to obtain a new IP address. However, this depends on the division of VLANs and their respective access policies.

Question 2

2a

In order to separate the VLANs that share a physical link on router rc1, we assigned two IP addresses each to the respective interfaces of that router. The way we assigned IP addresses and prefixes all VLANs are in separate subnets, but the devices within any single VLAN share a common IP prefix (see Figures 1 and 2).

2b

For the commands we used see section 2e.
PINGS

```
root@group08-lg1:~ ping 10.8.0.1
PING 10.8.0.1 (10.8.0.1) 56(84) bytes of data.
64 bytes from 10.8.0.1: icmp_seq=1 ttl=255 time=0.898 ms
64 bytes from 10.8.0.1: icmp_seq=2 ttl=255 time=0.850 ms
64 bytes from 10.8.0.1: icmp_seq=3 ttl=255 time=0.810 ms
64 bytes from 10.8.0.1: icmp_seq=4 ttl=255 time=0.937 ms

root@group08-lg1:~ ping fc00:470:525b:f800::1
```

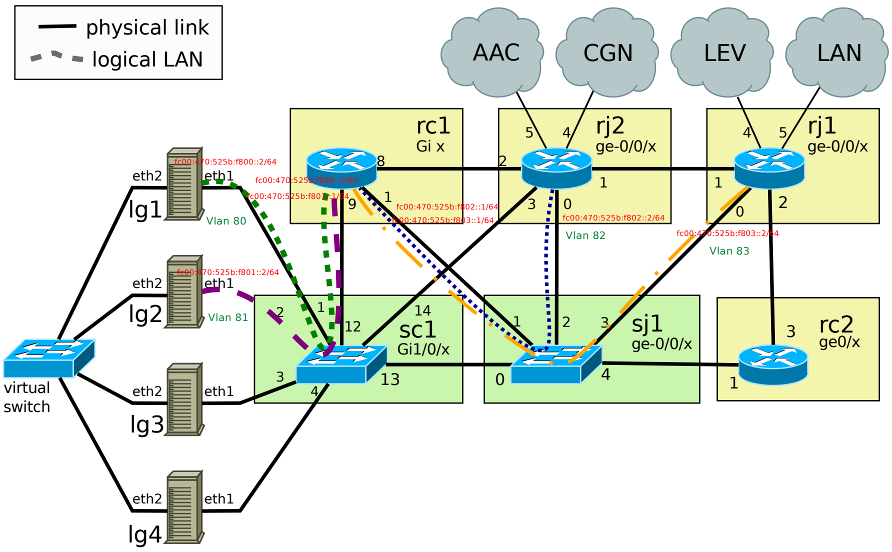

Figure 2: The IPv6 topology.

```
PING fc00:470:525b:f800::1(fc00:470:525b:f800::1) 56 data bytes
64 bytes from fc00:470:525b:f800::1: icmp_seq=1 ttl=64 time=2.24 ms
64 bytes from fc00:470:525b:f800::1: icmp_seq=2 ttl=64 time=0.791 ms
64 bytes from fc00:470:525b:f800::1: icmp_seq=3 ttl=64 time=0.851 ms
64 bytes from fc00:470:525b:f800::1: icmp_seq=4 ttl=64 time=0.701 ms

root@group08-lg2:~# ping6 fc00:470:525b:f801::1
PING fc00:470:525b:f801::1(fc00:470:525b:f801::1) 56 data bytes
64 bytes from fc00:470:525b:f801::1: icmp_seq=1 ttl=64 time=0.682 ms
64 bytes from fc00:470:525b:f801::1: icmp_seq=2 ttl=64 time=0.874 ms
64 bytes from fc00:470:525b:f801::1: icmp_seq=3 ttl=64 time=0.849 ms
64 bytes from fc00:470:525b:f801::1: icmp_seq=4 ttl=64 time=0.930 ms

root@group08-lg2:~# ping 10.8.1.1
PING 10.8.1.1 (10.8.1.1) 56(84) bytes of data.
64 bytes from 10.8.1.1: icmp_seq=1 ttl=255 time=0.737 ms
64 bytes from 10.8.1.1: icmp_seq=2 ttl=255 time=0.996 ms
64 bytes from 10.8.1.1: icmp_seq=3 ttl=255 time=0.945 ms

lev-rc1#ping fc00:470:525b:f800::2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to FC00:470:525B:F800::2, timeout is 2
    seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4 ms

lev-rc1#ping 10.8.0.2
Type escape sequence to abort.
```

```
Sending 5, 100-byte ICMP Echos to 10.8.0.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4 ms

lev-rc1#ping fc00:470:525b:f801::2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to FC00:470:525B:F801::2, timeout is 2
    seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4 ms

lev-rc1#ping 10.8.1.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.8.1.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
```

2c

Without VLANs all routers (rj1 and rj2) will see the frame sent from rc1 to rj2, if it is sent over the interface Gi1 of rc1, because hubs simply forward any incoming frame to all connected ports that are up (except the port from which the frame was received). The receiving hosts will drop the frames, if they are not the intended destination.

2d

2e

For the topology see Figures 1 and 2. SC1

```
lev-sc1(config-if)#vlan 80
lev-sc1(config-vlan)#vlan 81
lev-sc1(config-vlan)#interface Gi1/0/12
lev-sc1(config-if)#switchport mode trunk
lev-sc1(config-if)#switchport trunk allowed vlan add 80
lev-sc1(config-if)#switchport trunk allowed vlan add 81
lev-sc1(config-if)#no shutdown
lev-sc1(config-if)#interface Gi1/0/1
lev-sc1(config-if)#switchport mode access
lev-sc1(config-if)#switchport access vlan 80
lev-sc1(config-if)#no shutdown
lev-sc1(config-if)#interface Gi1/0/2
lev-sc1(config-if)#switchport mode access
lev-sc1(config-if)#switchport access vlan 81
lev-sc1(config-if)#no shutdown
```

RJ2

```
root@lev-rj2# set interfaces ge-0/0/0 vlan-tagging unit 82 vlan-id 82
root@lev-rj2# set interfaces ge-0/0/0 unit 82 family inet6 address fc00
    :470:525b:f802::2/64
root@lev-rj2# set interfaces ge-0/0/0 unit 82 family inet address
    10.8.2.0/31
root@lev-rj2# set security zones security-zone anywhere host-inbound-
    traffic system-services ping
root@lev-rj2# commit
```

```
root@lev-rj2# edit interfaces ge-0/0/0
[edit interfaces ge-0/0/0]
root@lev-rj2# delete unit 0
root@lev-rj2# delete disable
root@lev-rj2# exit
[edit]
root@lev-rj2# commit
commit complete
```

RJ1

```
root@lev-rj1# set interfaces ge-0/0/0 vlan-tagging unit 83 vlan-id 83
root@lev-rj1# set interfaces ge-0/0/0 unit 83 family inet6 address fc00
    :470:525b:f803::2/64
root@lev-rj1# set interfaces ge-0/0/0 unit 83 family inet address
    10.8.3.0/31
root@lev-rj1# set security zones security-zone anywhere host-inbound-
    traffic system-services ping
root@lev-rj1# commit
root@lev-rj1# edit interfaces ge-0/0/0
[edit interfaces ge-0/0/0]
root@lev-rj1# delete unit 0
root@lev-rj1# delete disable
root@lev-rj1# exit
[edit]
root@lev-rj1# commit
commit complete
```

SJ1

```
root@lev-sj1#set vlans vlan82 vlan-id 82
root@lev-sj1#set vlans vlan83 vlan-id 83
root@lev-sj1#edit interfaces ge-0/0/2
[edit interfaces ge-0/0/2]
root@lev-sj1#set unit 0 family ethernet-switching port-mode access
root@lev-sj1#set unit 0 family ethernet-switching vlan members vlan82
root@lev-sj1#exit
[edit]
root@lev-sj1#edit interfaces ge-0/0/3
[edit interfaces ge-0/0/3]
root@lev-sj1#set unit 0 family ethernet-switching port-mode access
root@lev-sj1#set unit 0 family ethernet-switching vlan members vlan83
root@lev-sj1#exit
[edit]
root@lev-sj1#edit interfaces ge-0/0/1
[edit interfaces ge-0/0/1]
root@lev-sj1#set unit 0 family ethernet-switching port-mode trunk
root@lev-sj1#set unit 0 family ethernet-switching vlan members vlan82
root@lev-sj1#set unit 0 family ethernet-switching vlan members vlan83
root@lev-sj1#exit
[edit]
root@lev-sj1#commit
```

RC1

```
lev-rc1(config)#interface GigabitEthernet9.80
lev-rc1(config-subif)#encapsulation dot1Q 80
lev-rc1(config-subif)#ip address 10.8.0.1 255.255.255.248
lev-rc1(config-subif)#ipv6 address FC00:470:525B:F800::1/64
```

```
lev-rc1(config-subif)#no shutdown
lev-rc1(config-subif)#exit
lev-rc1(config)#interface GigabitEthernet9.81
lev-rc1(config-subif)#encapsulation dot1Q 81
lev-rc1(config-subif)#ip address 10.8.1.1 255.255.255.248
lev-rc1(config-subif)#ipv6 address FC00:470:525B:F801::1/64
lev-rc1(config-subif)#no shutdown
lev-rc1(config-subif)#exit
lev-rc1(config)#interface Vlan82
lev-rc1(config-if)#ip address 10.8.2.1 255.255.255.254
lev-rc1(config-if)#ipv6 address FC00:470:525B:F802::1/64
lev-rc1(config-if)#no shutdown
lev-rc1(config-if)#exit
lev-rc1(config)#interface Vlan83
lev-rc1(config-if)#ip address 10.8.3.1 255.255.255.254
lev-rc1(config-if)#ipv6 address FC00:470:525B:F803::1/64
lev-rc1(config-if)#no shutdown
lev-rc1(config-if)#exit
lev-rc1(config)#interface GigabitEthernet1
lev-rc1(config-if)#switchport mode trunk
lev-rc1(config-if)#no shutdown
lev-rc1(config-if)#exit
lev-rc1(config)#exit
```

```
root@group08-lg1:~ ping 10.8.0.1
PING 10.8.0.1 (10.8.0.1) 56(84) bytes of data.
64 bytes from 10.8.0.1: icmp_seq=1 ttl=255 time=0.898 ms
64 bytes from 10.8.0.1: icmp_seq=2 ttl=255 time=0.850 ms
64 bytes from 10.8.0.1: icmp_seq=3 ttl=255 time=0.810 ms
64 bytes from 10.8.0.1: icmp_seq=4 ttl=255 time=0.937 ms
```

```
root@group08-lg1:~ ping fc00:470:525b:f800::1
PING fc00:470:525b:f800::1(fc00:470:525b:f800::1) 56 data bytes
64 bytes from fc00:470:525b:f800::1: icmp_seq=1 ttl=64 time=2.24 ms
64 bytes from fc00:470:525b:f800::1: icmp_seq=2 ttl=64 time=0.791 ms
64 bytes from fc00:470:525b:f800::1: icmp_seq=3 ttl=64 time=0.851 ms
64 bytes from fc00:470:525b:f800::1: icmp_seq=4 ttl=64 time=0.701 ms
```

```
root@group08-lg2:~# ping6 fc00:470:525b:f801::1
PING fc00:470:525b:f801::1(fc00:470:525b:f801::1) 56 data bytes
64 bytes from fc00:470:525b:f801::1: icmp_seq=1 ttl=64 time=0.682 ms
64 bytes from fc00:470:525b:f801::1: icmp_seq=2 ttl=64 time=0.874 ms
64 bytes from fc00:470:525b:f801::1: icmp_seq=3 ttl=64 time=0.849 ms
64 bytes from fc00:470:525b:f801::1: icmp_seq=4 ttl=64 time=0.930 ms
```

```
root@group08-lg2:~# ping 10.8.1.1
PING 10.8.1.1 (10.8.1.1) 56(84) bytes of data.
64 bytes from 10.8.1.1: icmp_seq=1 ttl=255 time=0.737 ms
64 bytes from 10.8.1.1: icmp_seq=2 ttl=255 time=0.996 ms
64 bytes from 10.8.1.1: icmp_seq=3 ttl=255 time=0.945 ms
```

Question 3

3a

To connect different VLANS we have to use the Network Layer (Layer 3) in order to make leverage of the available IP routers.

3b

1) lg1 participates directly in subnet 10.8.0.2/29. lg2 participates directly in subnet 10.8.1.2/29. lg3 participates directly in subnet 10.8.0/29 and 10.8.1/29 via VLAN 80 and VLAN 81 respectively. lg1 and lg2 cannot reach each other, because they are logically separate networks. Howerver, they can reach lg3 via VLAN 80 and VLAN 81 respectively.
2) Router rc1 is always the best next-hop. (We could also have defined it as the default route.)

```
root@group08-lg1:~# ip route add 10.8.1.0/29 via 10.8.0.1 dev eth1
root@group08-lg1:~# ip route add 10.8.2.0/31 via 10.8.0.1 dev eth1
root@group08-lg1:~# ip route add 10.8.3.0/31 via 10.8.0.1 dev eth1
root@group08-lg1:~# ip route add fc00:470:525b:f801::0/64 via fc00
    :470:525b:f800::1 dev eth1
root@group08-lg1:~# ip route add fc00:470:525b:f802::0/64 via fc00
    :470:525b:f800::1 dev eth1
root@group08-lg1:~# ip route add fc00:470:525b:f803::0/64 via fc00
    :470:525b:f800::1 dev eth1

root@group08-lg2:~# ip route add 10.8.0.0/29 via 10.8.1.1 dev eth1
root@group08-lg2:~# ip route add 10.8.2.0/31 via 10.8.1.1 dev eth1
root@group08-lg2:~# ip route add 10.8.3.0/31 via 10.8.1.1 dev eth1
root@group08-lg2:~# ip route add fc00:470:525b:f800::0/64 via fc00
    :470:525b:f801::1 dev eth1
root@group08-lg2:~# ip route add fc00:470:525b:f802::0/64 via fc00
    :470:525b:f801::1 dev eth1
root@group08-lg2:~# ip route add fc00:470:525b:f803::0/64 via fc00
    :470:525b:f801::1 dev eth1
```

3c

All devices that do not share a subnet with the loadgens need a route configured to reach them. In our case this concerns rj1 and rj2. We configured rc1 to be the route for each device to reach devices outside of its own subnet.

3d

```
lev-rc1(config)#ipv6 unicast-routing
lev-rc1(config)#ip routing

root@lev-rj2#set routing-options rib inet6.0 static route fc00:470:525b
    :f800::/64 next-hop fc00:470:525b:f802::1
[edit]
root@lev-rj2# set routing-options rib inet6.0 static route fc00:470:525
    b:f801::/64 next-hop fc00:470:525b:f802::1
[edit]
root@lev-rj2# set routing-options rib inet6.0 static route fc00:470:525
    b:f803::/64 next-hop fc00:470:525b:f802::1
[edit]
root@lev-rj2# set routing-options static route 10.8.0.0/24 next-hop
    10.8.2.1
[edit]
root@lev-rj2# set routing-options static route 10.8.1.0/24 next-hop
    10.8.2.1
[edit]
root@lev-rj2# set routing-options static route 10.8.3.0/24 next-hop
    10.8.2.1
```

```
[edit]
root@lev-rj2# commit
commit complete

root@lev-rj1# set routing-options rib inet6.0 static route fc00:470:525
    b:f800::/64 next-hop fc00:470:525b:f803::1
[edit]
root@lev-rj1# set routing-options rib inet6.0 static route fc00:470:525
    b:f801::/64 next-hop fc00:470:525b:f803::1
[edit]
root@lev-rj1# set routing-options rib inet6.0 static route fc00:470:525
    b:f802::/64 next-hop fc00:470:525b:f803::1
[edit]
root@lev-rj1# set routing-options static route 10.8.0.0/24 next-hop
    10.8.3.1
[edit]
root@lev-rj1# set routing-options static route 10.8.1.0/24 next-hop
    10.8.3.1
[edit]
root@lev-rj1# set routing-options static route 10.8.2.0/24 next-hop
    10.8.3.1
[edit]
root@lev-rj1# commit
commit complete
```

3e

Question 4

4a

See Figures 3 and 4.

4b

```
lev-sc1(config-vlan)#interface Gi1/0/3
lev-sc1(config-if)#switchport mode trunk
lev-sc1(config-if)#no shutdown
lev-sc1(config-if)#switchport trunk allowed vlan add 81
lev-sc1(config-if)#switchport trunk allowed vlan add 80
```

4c

We also assigned IPv6 addresses using ip instead of ifconfig.

```
root@group08-lg3:~# apt install vlan
root@group08-lg3:~# modprobe 8021q
root@group08-lg3:~# vconfig add eth1 80
Added VLAN with VID == 80 to IF -:eth1:-
root@group08-lg3:~# vconfig add eth1 81
Added VLAN with VID == 81 to IF -:eth1:-
root@group08-lg3:~# ip address add 10.8.0.3/29 dev eth1.80
root@group08-lg3:~# ip address add 10.8.1.3/29 dev eth1.81
root@group08-lg3:~# ip address add fc00:470:525b:f800::3/64 dev eth1.80
root@group08-lg3:~# ip address add fc00:470:525b:f801::3/64 dev eth1.81
root@group08-lg3:~# ip link set up dev eth1
root@group08-lg3:~# ip link set up dev eth1.80
root@group08-lg3:~# ip link set up dev eth1.81
```

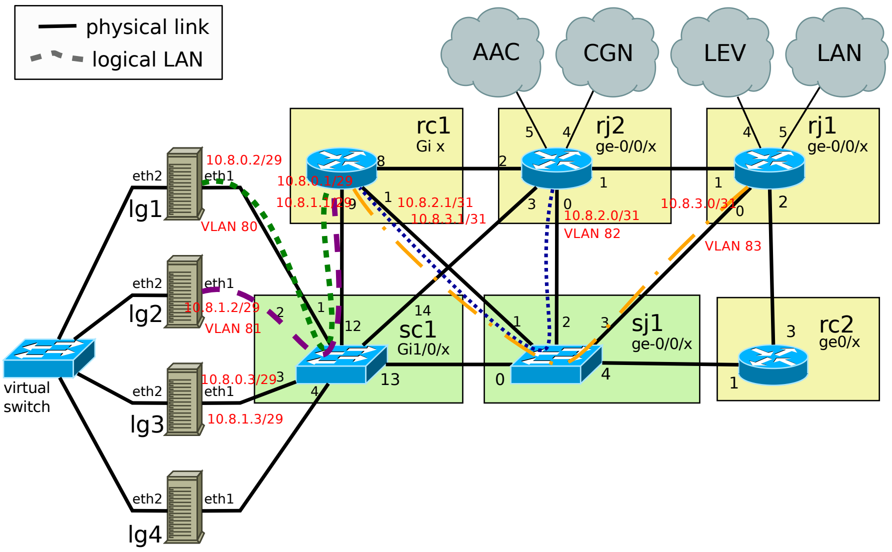

Figure 3: The new IPv4 topology.

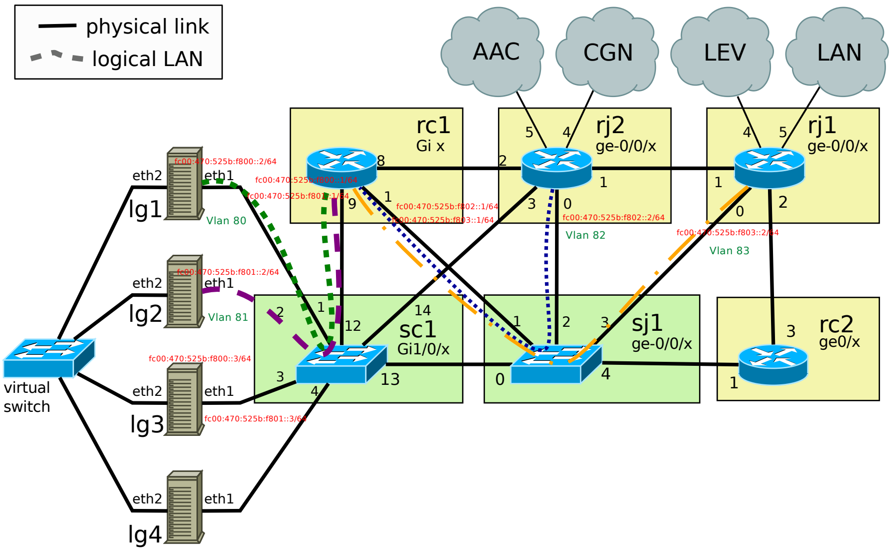

Figure 4: The new IPv6 topology.

Traceroutes:

```
root@group08-lg3:~# traceroute 10.8.1.2
traceroute to 10.8.1.2 (10.8.1.2), 30 hops max, 60 byte packets
 1  10.8.1.2 (10.8.1.2)  0.671 ms  0.620 ms  0.587 ms
root@group08-lg3:~# traceroute 10.8.0.2traceroute to 10.8.0.2
    (10.8.0.2), 30 hops max, 60 byte packets
 1  10.8.0.2 (10.8.0.2)  0.842 ms  0.777 ms  0.733 msroot@group08-lg3
     :~#
root@group08-lg3:~# traceroute fc00:470:525b:f801::2   traceroute to
    fc00:470:525b:f801::2 (fc00:470:525b:f801::2), 30 hops max, 80 byte
     packets
 1  fc00:470:525b:f801::2 (fc00:470:525b:f801::2)  1.388 ms  1.310 ms
     1.264 ms
root@group08-lg3:~# traceroute fc00:470:525b:f800::2
traceroute to fc00:470:525b:f800::2 (fc00:470:525b:f800::2), 30 hops
    max, 80 byte packets
 1  fc00:470:525b:f800::2 (fc00:470:525b:f800::2)  1.300 ms  1.199 ms
     1.165 ms
```

Question 5

5a

Nothing would change, because the ports are down on both switches. If we bring them up, we create a loop:
$lev - sc1 \Rightarrow_{Gi1/0/14 \rightarrow ge-0/0/7} lev - sj1$
$lev - sj1 \Rightarrow_{ge-0/0/0 \rightarrow Gi1/0/13} lev - sc1$
This loop will prevent ARP from working and the network will be flooded with (broadcast) messages, since there is no TTL on layer2.

5b

STP is designed to prevent switching loops. The protocol creates a spanning tree between all nodes, i.e. all network devices, where links are the edges. (Trees by definition do not contain any loops.) The protocol then deactivates all links that are not part of the spanning tree. In that way the protocol ensures that there is always just one path between two nodes. If a link fails the protocol is able to change the spanning tree and bring another link up to replace it. The protocol also adapts the spanning tree, if new devices and links are added.

5c

rSTP would only create a single spanning tree not taking the VLANs into consideration. Therefore one of the two links would be brought down and the traffic of the other VLAN could not be forwarded (or the config would have to be changed to forward all traffic just over one trunk port). So if one wants to use separate access ports for both VLANs, one needs to use a protocol that allows switches to create one spanning tree per VLAN. PVSTP allows network administrators to set different roots from which the spanning trees are created for each VLAN. The advantage of this approach is that none or less links have to be down and the traffic is distributed over more links, which will make the network scale better, while at the same time there are no loops because the traffic of each VLAN is just forwarded in its according spanning tree. MSTP is basically an evolved and standardized form of PVSTP.[1]

[1] https://en.wikipedia.org/wiki/Spanning_Tree_Protocol

Included Files

q02-config-rc1.txt, q02-config-rj1.txt, q02-config-rj2.txt, q02-config-sc1.txt, q02-config-sj1.txt, q04-config-sc1.txt