NETWORK SECURITY

Université Libre de Bruxelles

Faculty of Sciences

Lab 2 - VLANs and Layer 2 security

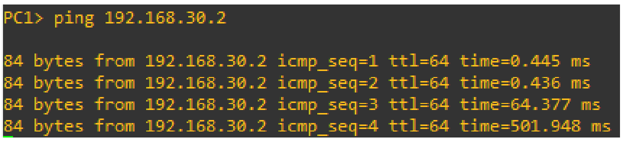
Authors:

Croche Loïc
Hanquin Benjamin
Cochez Benjamin

May 1, 2021

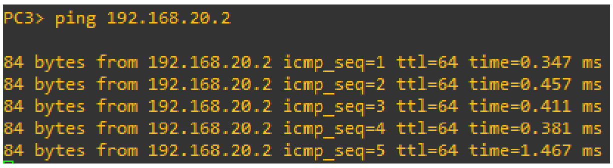
I. Mission 1: VLAN isolation and trunking

First of all, we can show on these two pictures (figure 1 and 2) that the PC1 can communicate with the PC3 when there is no VLAN. To test this communication, we have to change the subnet mask just before.



```
PC1> ping 192.168.30.2

84 bytes from 192.168.30.2 icmp_seq=1 ttl=64 time=0.445 ms
84 bytes from 192.168.30.2 icmp_seq=2 ttl=64 time=0.436 ms
84 bytes from 192.168.30.2 icmp_seq=3 ttl=64 time=64.377 ms
84 bytes from 192.168.30.2 icmp_seq=4 ttl=64 time=501.948 ms
```

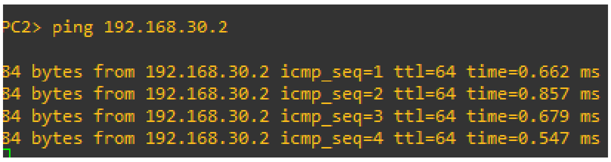
Figure 1: Ping from PC1 to PC3.



```
PC3> ping 192.168.20.2

84 bytes from 192.168.20.2 icmp_seq=1 ttl=64 time=0.347 ms
84 bytes from 192.168.20.2 icmp_seq=2 ttl=64 time=0.457 ms
84 bytes from 192.168.20.2 icmp_seq=3 ttl=64 time=0.411 ms
84 bytes from 192.168.20.2 icmp_seq=4 ttl=64 time=0.381 ms
84 bytes from 192.168.20.2 icmp_seq=5 ttl=64 time=1.467 ms
```

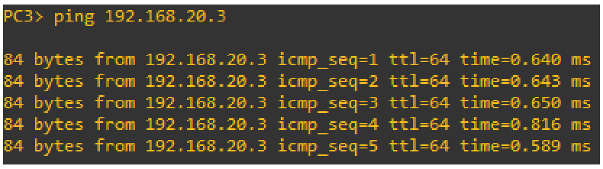
Figure 2: Ping from PC3 to PC1.

We also can show the same observation with the PC2 and PC3 (figure 3 and 4).



```
PC2> ping 192.168.30.2

84 bytes from 192.168.30.2 icmp_seq=1 ttl=64 time=0.662 ms
84 bytes from 192.168.30.2 icmp_seq=2 ttl=64 time=0.857 ms
84 bytes from 192.168.30.2 icmp_seq=3 ttl=64 time=0.679 ms
84 bytes from 192.168.30.2 icmp_seq=4 ttl=64 time=0.547 ms
```

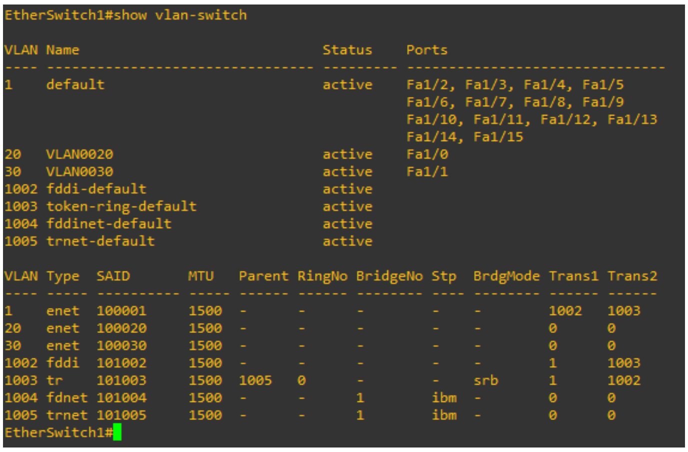
Figure 3: Ping from PC2 to PC3.



```
PC3> ping 192.168.20.3

84 bytes from 192.168.20.3 icmp_seq=1 ttl=64 time=0.640 ms
84 bytes from 192.168.20.3 icmp_seq=2 ttl=64 time=0.643 ms
84 bytes from 192.168.20.3 icmp_seq=3 ttl=64 time=0.650 ms
84 bytes from 192.168.20.3 icmp_seq=4 ttl=64 time=0.816 ms
84 bytes from 192.168.20.3 icmp_seq=5 ttl=64 time=0.589 ms
```

Figure 4: Ping from PC3 to PC2.

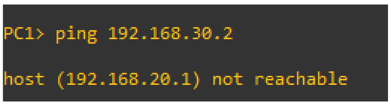
Then, we configured VLANs on switches and we can see that VLANs are correctly configured for EtherSwitch1 on the figure 5. We don't show the result for EtherSwitch2 because the configuration is the same.



```
EtherSwitch1#show vlan-switch

VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
1    default                          active    Fa1/2, Fa1/3, Fa1/4, Fa1/5
                                                Fa1/6, Fa1/7, Fa1/8, Fa1/9
                                                Fa1/10, Fa1/11, Fa1/12, Fa1/13
                                                Fa1/14, Fa1/15
20   VLAN0020                         active    Fa1/0
30   VLAN0030                         active    Fa1/1
1002 fddi-default                     active
1003 token-ring-default               active
1004 fddinet-default                  active
1005 trnet-default                    active

VLAN Type  SAID       MTU   Parent RingNo BridgeNo Stp  BrdgMode Trans1 Trans2
---- ----- ---------- ----- ------ ------ -------- ---- -------- ------ ------
1    enet  100001     1500  -      -      -        -    -        1002   1003
20   enet  100020     1500  -      -      -        -    -        0      0
30   enet  100030     1500  -      -      -        -    -        0      0
1002 fddi  101002     1500  -      -      -        -    -        1      1003
1003 tr    101003     1500  1005   0      -        -    srb      1      1002
1004 fdnet 101004     1500  -      -      1        ibm  -        0      0
1005 trnet 101005     1500  -      -      1        ibm  -        0      0
EtherSwitch1#
```

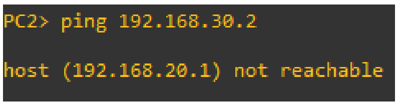
Figure 5: VLAN configuration on EtherSwitch1.

Once VLAN are configured on switches, we can observe on figures 6 and 7 that the PC1 and PC2 cannot communicate with the PC3.



```
PC1> ping 192.168.30.2

host (192.168.20.1) not reachable
```

Figure 6: Ping from PC1 to PC3.

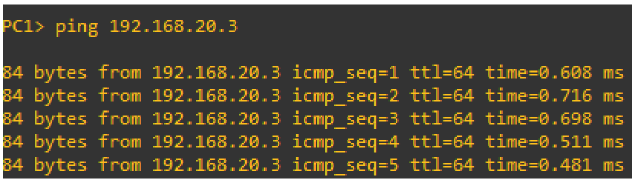


```
PC2> ping 192.168.30.2

host (192.168.20.1) not reachable
```

Figure 7: Ping from PC2 to PC3.

II. Mission 2: Trunking

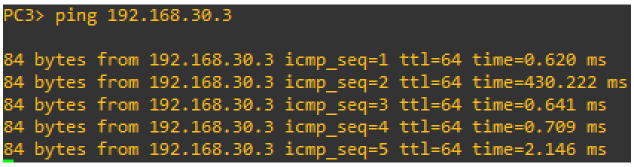
Once the trunk has been configured in both switches, we can show on the figure 8 that PC1 and PC2, which are on the same VLAN but connected to a different switch, can communicate each other.



```
PC1> ping 192.168.20.3

84 bytes from 192.168.20.3 icmp_seq=1 ttl=64 time=0.608 ms
84 bytes from 192.168.20.3 icmp_seq=2 ttl=64 time=0.716 ms
84 bytes from 192.168.20.3 icmp_seq=3 ttl=64 time=0.698 ms
84 bytes from 192.168.20.3 icmp_seq=4 ttl=64 time=0.511 ms
84 bytes from 192.168.20.3 icmp_seq=5 ttl=64 time=0.481 ms
```

Figure 8: Communication from PC1 to PC2.

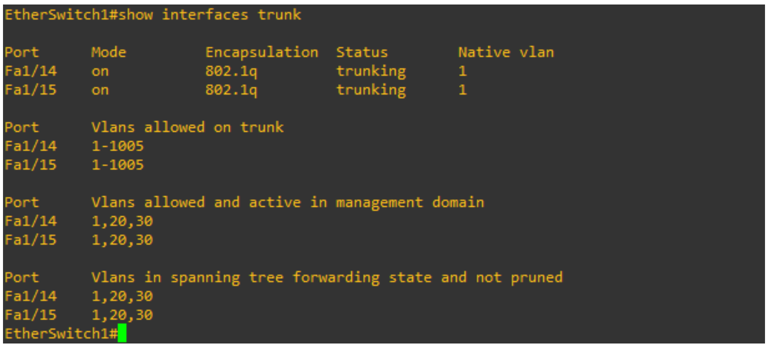
Same observation for the communication between PC3 and PC4 on the figure 9.



```
PC3> ping 192.168.30.3

84 bytes from 192.168.30.3 icmp_seq=1 ttl=64 time=0.620 ms
84 bytes from 192.168.30.3 icmp_seq=2 ttl=64 time=430.222 ms
84 bytes from 192.168.30.3 icmp_seq=3 ttl=64 time=0.641 ms
84 bytes from 192.168.30.3 icmp_seq=4 ttl=64 time=0.709 ms
84 bytes from 192.168.30.3 icmp_seq=5 ttl=64 time=2.146 ms
```

Figure 9: Communication from PC3 to PC4.

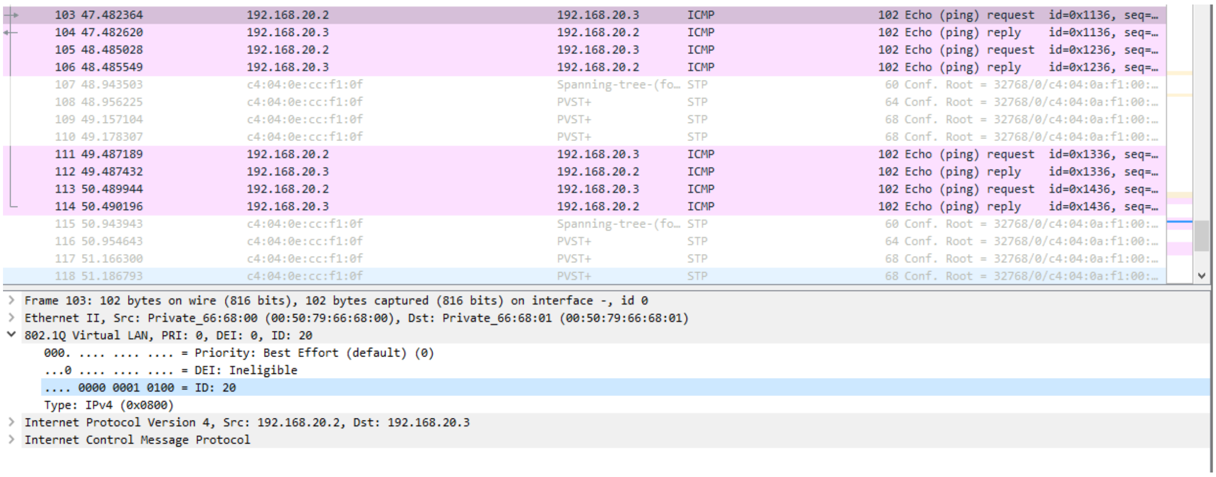
We can also show on the EtherSwitch1 the configuration of the trunk with the command “show interfaces trunk”. We can show on the figure 10 that it displays interfaces which accept the trunk and which VLAN is allowed through them (interface 1/15 is used after for the routing).



```
EtherSwitch1#show interfaces trunk

Port      Mode         Encapsulation  Status        Native vlan
Fa1/14    on           802.1q         trunking      1
Fa1/15    on           802.1q         trunking      1

Port      Vlans allowed on trunk
Fa1/14    1-1005
Fa1/15    1-1005

Port      Vlans allowed and active in management domain
Fa1/14    1,20,30
Fa1/15    1,20,30

Port      Vlans in spanning tree forwarding state and not pruned
Fa1/14    1,20,30
Fa1/15    1,20,30
EtherSwitch1#
```

Figure 10: Trunk configuration on the switch1.

When we capture the traffic on the trunk line (figure 11), we can intercept the ping from the PC1 to the PC2 and notice that the 802.1Q header contains the VLAN ID 20.



```
103 47.482364   192.168.20.2        192.168.20.3        ICMP   102 Echo (ping) request  id=0x1136, seq=…
104 47.482620   192.168.20.3        192.168.20.2        ICMP   102 Echo (ping) reply    id=0x1136, seq=…
105 48.485028   192.168.20.2        192.168.20.3        ICMP   102 Echo (ping) request  id=0x1236, seq=…
106 48.485549   192.168.20.3        192.168.20.2        ICMP   102 Echo (ping) reply    id=0x1236, seq=…
107 48.943503   c4:04:0e:cc:f1:0f   Spanning-tree-(fo…  STP     60 Conf. Root = 32768/0/c4:04:0a:f1:00:…
108 48.956225   c4:04:0e:cc:f1:0f   PVST+               STP     64 Conf. Root = 32768/0/c4:04:0a:f1:00:…
109 49.157104   c4:04:0e:cc:f1:0f   PVST+               STP     68 Conf. Root = 32768/0/c4:04:0a:f1:00:…
110 49.178307   c4:04:0e:cc:f1:0f   PVST+               STP     68 Conf. Root = 32768/0/c4:04:0a:f1:00:…
111 49.487189   192.168.20.2        192.168.20.3        ICMP   102 Echo (ping) request  id=0x1336, seq=…
112 49.487432   192.168.20.3        192.168.20.2        ICMP   102 Echo (ping) reply    id=0x1336, seq=…
113 50.489944   192.168.20.2        192.168.20.3        ICMP   102 Echo (ping) request  id=0x1436, seq=…
114 50.490196   192.168.20.3        192.168.20.2        ICMP   102 Echo (ping) reply    id=0x1436, seq=…
115 50.943943   c4:04:0e:cc:f1:0f   Spanning-tree-(fo…  STP     60 Conf. Root = 32768/0/c4:04:0a:f1:00:…
116 50.954643   c4:04:0e:cc:f1:0f   PVST+               STP     64 Conf. Root = 32768/0/c4:04:0a:f1:00:…
117 51.166300   c4:04:0e:cc:f1:0f   PVST+               STP     68 Conf. Root = 32768/0/c4:04:0a:f1:00:…
118 51.186793   c4:04:0e:cc:f1:0f   PVST+               STP     68 Conf. Root = 32768/0/c4:04:0a:f1:00:…

> Frame 103: 102 bytes on wire (816 bits), 102 bytes captured (816 bits) on interface -, id 0
> Ethernet II, Src: Private_66:68:00 (00:50:79:66:68:00), Dst: Private_66:68:01 (00:50:79:66:68:01)
v 802.1Q Virtual LAN, PRI: 0, DEI: 0, ID: 20
    000. .... .... .... = Priority: Best Effort (default) (0)
    ...0 .... .... .... = DEI: Ineligible
    .... 0000 0001 0100 = ID: 20
    Type: IPv4 (0x0800)
> Internet Protocol Version 4, Src: 192.168.20.2, Dst: 192.168.20.3
> Internet Control Message Protocol
```

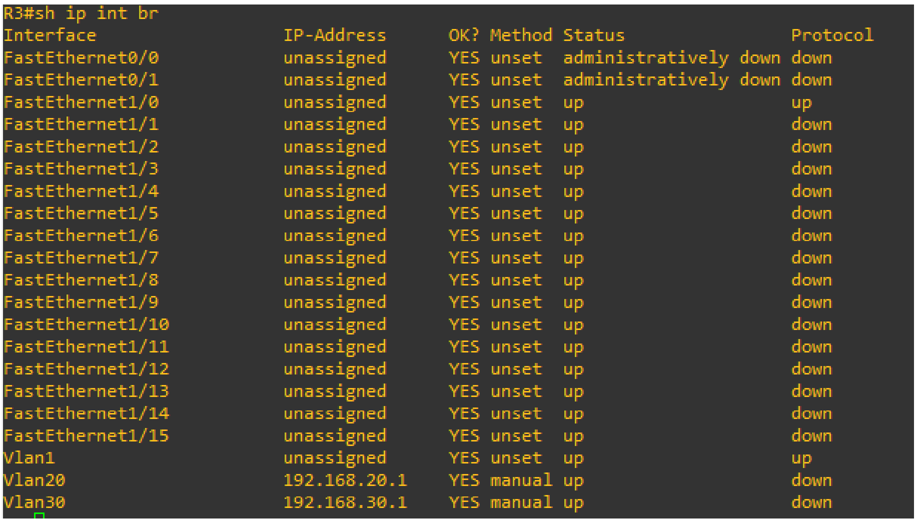
Figure 11: Analyze of ICMP packet in Wireshark.

It’s also possible to filter packet in Wireshark to be faster targeting specifically a VLAN with the command “vlan.id==20” and even be more precisely indicating the desired protocol. In our case we want to analyze ICMP packets, so we can use this command “vlan.id == 30 && icmp”.

III. Mission 3: Inter-VLAN routing

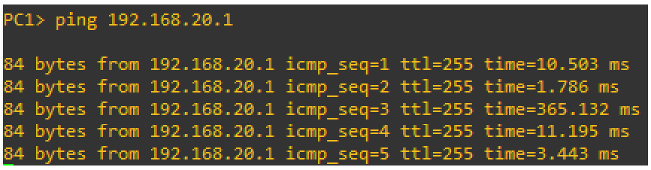
The topology is called router on a stick because it’s an isolated router which uses only one physical Ethernet connection with a switch to forward packets from one VLAN to another one (inter-VLAN routing). It allows computers on different VLANs to discuss together. The physical connection is defined as a Trunk line following the IEEE 802.1Q network standard. This method is interesting, but we must keep in mind that it can creates a bottleneck on the Trunk line if there are a lot of VLAN communications. To avoid this problem, we can use a L3 switch which optimizes the performance thanks to the possibility to directly route the network traffic.

1. To enable the inter-VLAN routing, we must configure the router adding a trunk between the EtherSwitch1 and the Router1 and configuring VLANs. Once the router configured, we can see on the figure 11 that VLAN interfaces are correctly added. To try these interfaces, we send a ping from the PC1 to the router interface (figure 12).



```
R3#sh ip int br
Interface                  IP-Address      OK? Method Status                Protocol
FastEthernet0/0            unassigned      YES unset  administratively down down
FastEthernet0/1            unassigned      YES unset  administratively down down
FastEthernet1/0            unassigned      YES unset  up                    up
FastEthernet1/1            unassigned      YES unset  up                    down
FastEthernet1/2            unassigned      YES unset  up                    down
FastEthernet1/3            unassigned      YES unset  up                    down
FastEthernet1/4            unassigned      YES unset  up                    down
FastEthernet1/5            unassigned      YES unset  up                    down
FastEthernet1/6            unassigned      YES unset  up                    down
FastEthernet1/7            unassigned      YES unset  up                    down
FastEthernet1/8            unassigned      YES unset  up                    down
FastEthernet1/9            unassigned      YES unset  up                    down
FastEthernet1/10           unassigned      YES unset  up                    down
FastEthernet1/11           unassigned      YES unset  up                    down
FastEthernet1/12           unassigned      YES unset  up                    down
FastEthernet1/13           unassigned      YES unset  up                    down
FastEthernet1/14           unassigned      YES unset  up                    down
FastEthernet1/15           unassigned      YES unset  up                    down
Vlan1                      unassigned      YES unset  up                    up
Vlan20                     192.168.20.1    YES manual up                    down
Vlan30                     192.168.30.1    YES manual up                    down
```

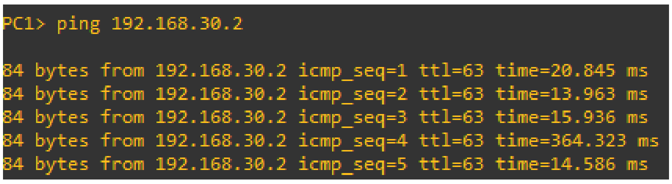
Figure 12: Interface's configuration of Router1.



```
PC1> ping 192.168.20.1

84 bytes from 192.168.20.1 icmp_seq=1 ttl=255 time=10.503 ms
84 bytes from 192.168.20.1 icmp_seq=2 ttl=255 time=1.786 ms
84 bytes from 192.168.20.1 icmp_seq=3 ttl=255 time=365.132 ms
84 bytes from 192.168.20.1 icmp_seq=4 ttl=255 time=11.195 ms
84 bytes from 192.168.20.1 icmp_seq=5 ttl=255 time=3.443 ms
```

Figure 13: Communication from PC1 to Router1.

2. We also can see on the figure 13 that the PC1 can communicate with the PC3 thanks to the inter-VLAN routing.



```
PC1> ping 192.168.30.2

84 bytes from 192.168.30.2 icmp_seq=1 ttl=63 time=20.845 ms
84 bytes from 192.168.30.2 icmp_seq=2 ttl=63 time=13.963 ms
84 bytes from 192.168.30.2 icmp_seq=3 ttl=63 time=15.936 ms
84 bytes from 192.168.30.2 icmp_seq=4 ttl=63 time=364.323 ms
84 bytes from 192.168.30.2 icmp_seq=5 ttl=63 time=14.586 ms
```

Figure 14: Communication from the PC1 to the PC3.

The following figure (figure 15) shows the path taken by the ICMP request packets during the ping. For the response phase, it's the same path but reversed.

We can say that the bandwidth is negatively impacted by this configuration because all inter-VLAN packets are using this stick in BOTH ways, meaning that is is a real bottleneck if the network traffic increase. If this link is overloaded, the whole inter-vlan traffic will be slowed down. As explained before, this problem can be mitigated using a L3 switch.

In terms of security, it's a good practice because it allows to segment the network and ensure that only devices on the same VLAN or which are trunked can communicate between each other. It allows to restrict some communication between some devices like a device from a financial department which wants to communicate with a device from a human resources department.

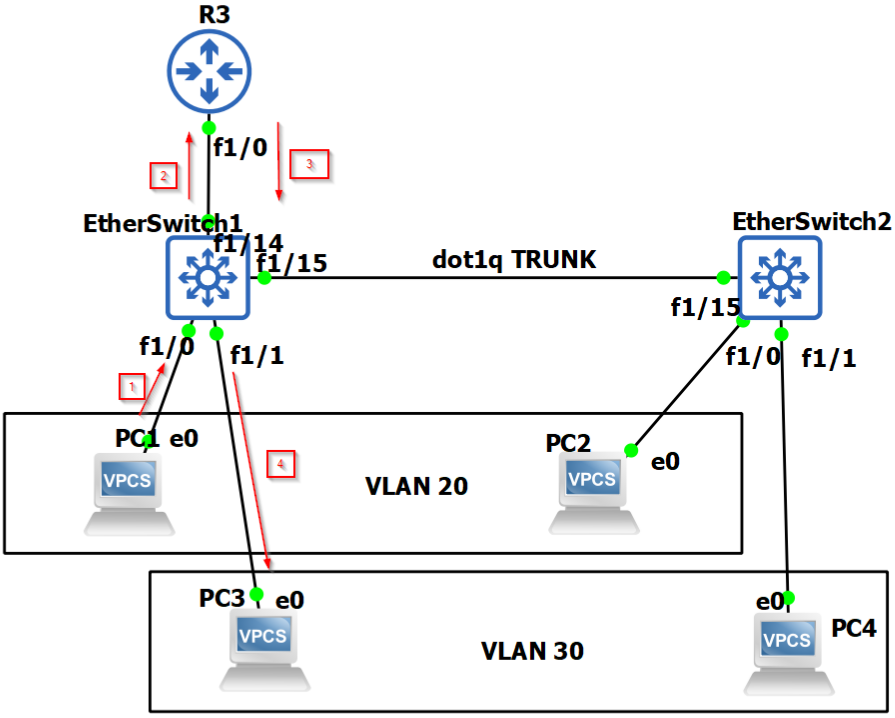


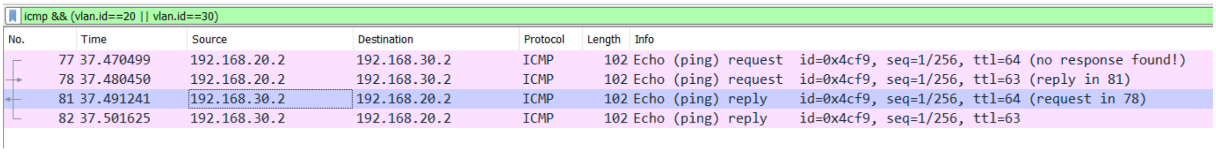
Figure 15: Path taken by packets.

3. When we monitor the stick line, we can show that all ICMP packets goes through the router, so all packets are re-encapsulated to change the VLAN ID and then communicate in layer 3 (IP) on the good VLAN. Since they are virtually separated LAN, ethernet (layer 2 can't be used). Using this way, PC1 and PC3 can communicate each other.

   The 4 packets are:

   - 77: Ping request on VLAN 20 from ES1 to R3 (2 on Figure 15)
   - 78: Ping request on VLAN 30 from R3 to ES1 (3 on Figure 15)
   - 81: Ping reply on VLAN 30 from ES1 to R3 (Inverted 3 on Figure 15)
   - 82: Ping reply on VLAN 20 from R3 to ES1 (Inverted 2 on Figure 15)

   The filter used in wireshark is *icmp && (vlad.id ==20* || *vlan.id==30)*. It filter only ICMP packets (ping in this case) having as vlan tag id 20 or 30.



icmp && (vlan.id==20 || vlan.id==30)

No.	Time	Source	Destination	Protocol	Length	Info
77	37.470499	192.168.20.2	192.168.30.2	ICMP	102	Echo (ping) request id=0x4cf9, seq=1/256, ttl=64 (no response found!)
78	37.480450	192.168.20.2	192.168.30.2	ICMP	102	Echo (ping) request id=0x4cf9, seq=1/256, ttl=63 (reply in 81)
81	37.491241	192.168.30.2	192.168.20.2	ICMP	102	Echo (ping) reply id=0x4cf9, seq=1/256, ttl=64 (request in 78)
82	37.501625	192.168.30.2	192.168.20.2	ICMP	102	Echo (ping) reply id=0x4cf9, seq=1/256, ttl=63

Figure 16: Filtered Wireshark analysis of PC1 to PC3 ping

IV. To go further

Management VLAN

As mentioned in the lab, a management VLAN is used in case someone screw the configuration and cut the network in parts. It is a common practice to put it as VLAN 99.

Scalability

In a wide network, configurate a lot of L2 switches can take a really long time. That is why Cisco created VTP (proprietary protocol) meaning VLAN Trunk Protocol, which can propagate the configuration of one well configured switch to other switches. Obviously if a rogue switch share his configuration all shared switch are going to have their configuration screwed ! That is why cisco implemented a password protected VTP domain. (need password to "connect" to the vtp domain).

Switched Spoofing

Avoid switched spoofing, rogue switch trunking with a configured switch, allowing the attacker to access the whole traffic of all VLANs. This can be mitigated by disallowing a switch to autonegociate a trunk. When configuring a switch interface, the *"switchport nonegotiate"* command should be used to disallow this autonegociation.

V. Conclusion

VLAN are useful to virtually separate and segment the physical network infrastructure. It allows to make harder attackers to exploit some protocols weakness in the network. It is quite easy to configure, but hardly scalable if the number of L2 switches increases. The usage of VLANs is obviously also used in other network security applications like in Network Access Controller (PacketFence) to ensure a better and easily manageable network security.