



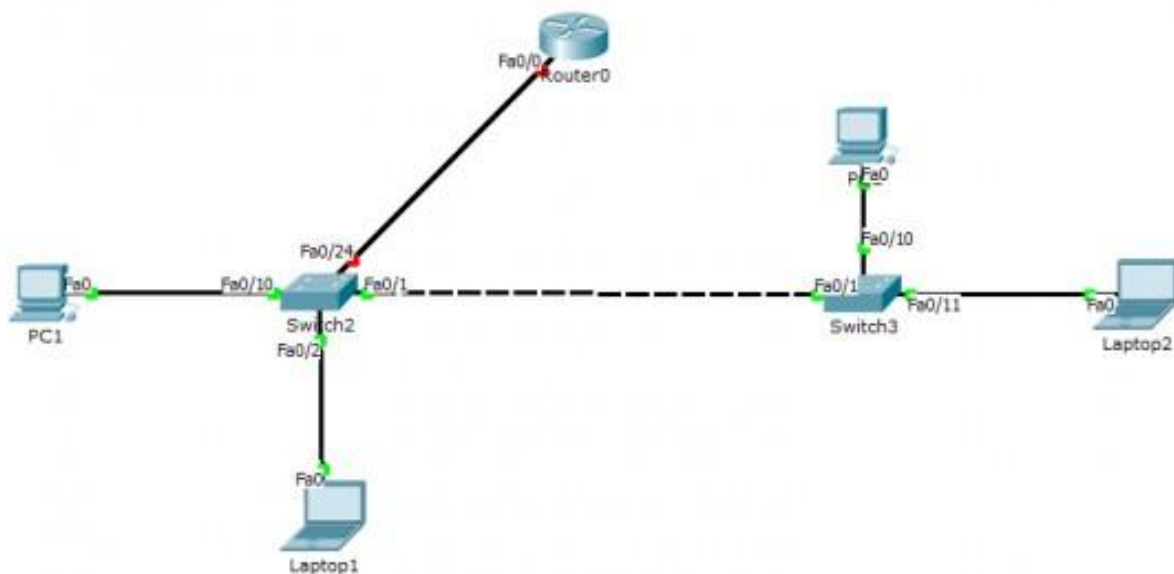
# MISE EN PLACE DE VLANS ET DE ROUTAGE INTER-VLANS

## ✚ Présentation

Ce réseau simple est composé de 4 postes de travail, deux **switchs** et un **routeur**. Les deux switchs partageront des VLANs et le routeur se chargera des tâches de routage inter-VLANs.

Un VLAN est un réseau local virtuel. Les VLANs permettent de créer plusieurs LAN sur une même infrastructure physique. Sur un LAN, toutes les machines appartiennent à un même domaine de diffusion, elles peuvent toutes communiquer entre elles. Le mot « virtuel » traduit la capacité à créer plusieurs LAN c'est-à-dire plusieurs domaines de diffusion sur un même switch. Nous ne pouvons les interconnecter **que par l'intermédiaire de routeur** (d'élément gérant la couche 3 - réseau plus spécifiquement). Nous avons maintenant besoin de router nos différents VLANs entre elles pour qu'elles puissent communiquer.

### ➤ Schéma du réseau



## Configuration des vlans

La première étape à suivre une fois que le câblage est en place est de créer les deux VLANS sur nos deux switchs. Pour faire simple, nous allons supposer que nous aurons deux VLANS (10 et 20) avec une liaison par port trunk entre le switch 2 et le switch 3.

```
Switch>enable
Switch#conf t
```

Nous allons ensuite créer les VLANS et les nommer :

```
Switch(config)#vlan 10
Switch(config-vlan)#name vlan_10
Switch(config-vlan)#vlan 20
Switch(config-vlan)#name vlan_20
Switch(config-vlan)#vlan 99
Switch(config-vlan)#name Native
Switch(config-vlan)#exit
```

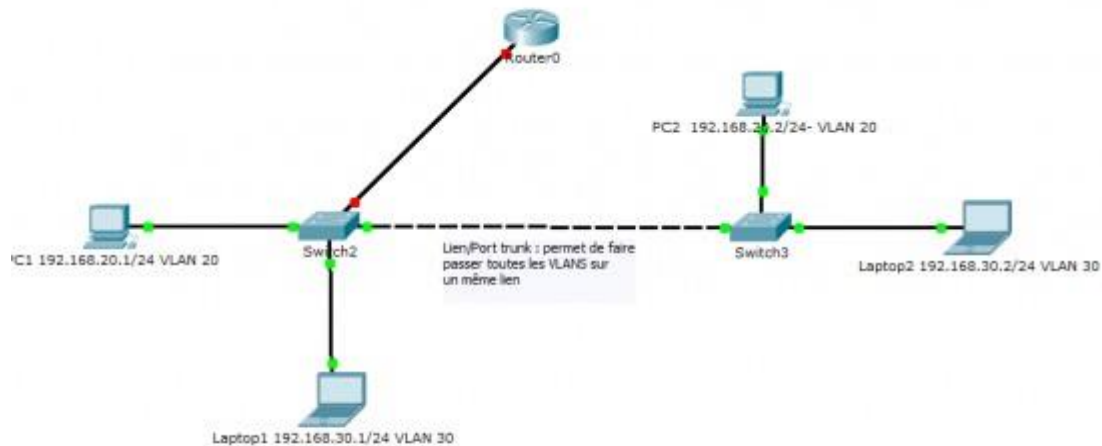
Nous allons maintenant créer nos ports trunk sur les interfaces Fa0/1 de nos deux switchs. Le port trunk va permettre, au travers des trames 802.1q de faire transiter des trames tagguées (ou étiquetées) selon un Vlan ou un autre afin que tous les Vlan autorisés puissent passer au travers d'un même lien. Plus clairement, c'est un port qui peut faire passer plusieurs VLAN vers un autre élément actif. Cela permet, dans notre cas, de faire communiquer les VLANS 10 et 20 entre des éléments connectés à deux switchs différents. Sans port trunk, il faudrait une liaison entre les switchs par VLANS.

```
Switch(config)#interface fa0/1
Switch(config-if)#switchport mode trunk
Switch(config-if)#switchport trunk allowed vlan 20,30,99
Switch(config-if)# switchport trunk native vlan 99
Switch(config-if)#no shutdown
Switch(config-if)#exit
```

On spécifie également les VLANS que nous souhaitons laisser passer sur notre trunk à savoir les trames étiquetées sur les VLANs 20,30 et 99. Par défaut, toutes les VLANS peuvent passer sur un port trunk. Si nous spécifions l'autorisation de certaines VLANs, les autres ne seront pas acceptés à transiter. Nous allons maintenant affecter les ports voulus à nos différentes VLANS. On présume que nous souhaitons affecter les ports Fa0/10 des deux switches sur le VLAN 20 et les ports Fa0/11 sur le VLAN 30, on exécute donc ces commandes sur nos deux switches :

---

```
Switch(config)#interface fa0/10
Switch(config-if)#switchport access vlan 20
Switch(config-if)#no shutdown
Switch(config-if)#exit
Switch(config)#interface fa0/11
Switch(config-if)#switchport access vlan 30
Switch(config-if)#no shutdown
Switch(config-if)#exit
```



Ajouter des adresses IP à mes postes. Ceux sur le VLAN 20 appartient au réseau "**192.168.20.0/24**" et ceux sur le VLAN 30 appartiennent au réseau "**192.168.30.0/24**"

## Test Vlan

Nous allons maintenant tester la connectivité des postes situées sur la même VLAN. On ne prend pas exemple le poste "Laptop2" sur le VLAN 30 et avec l'IP 192.168.30.2 pour pinger le poste "**Laptop1**" située sur la VLAN 30 de l'autre switch et avec l'IP 192.168.30.1 :

```

Laptop2 192.168.30.2/24 VLAN 30
Physical Config Desktop Software/Services
Command Prompt
PC>ping 192.168.30.1

Pinging 192.168.30.1 with 32 bytes of data:

Reply from 192.168.30.1: bytes=32 time=0ms TTL=128
Reply from 192.168.30.1: bytes=32 time=0ms TTL=128
Reply from 192.168.30.1: bytes=32 time=0ms TTL=128
Reply from 192.168.30.1: bytes=32 time=0ms TTL=128

Ping statistics for 192.168.30.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
  
```

On peut faire la même chose en faisant un ping sur le poste "**PC1**" au poste "**PC2**" qui sont également tous deux sur le même **VLAN** (20 cette fois ci). Les plus curieux auront remarqué que le "**PC1**" ou "**PC2**" ne peuvent pinguer "**Laptop1**" et "Laptop2" qui sont sur des **VLANS** différentes (20 et 30). C'est justement la problématique que nous venons de nous créer est que les VLANS isolent correctement les groupes de postes/d'utilisateur mais bloquent complètement la communication entre elles.

## Mise en place de vlans et de routage inter-vlans

### ➤ Présentation

Nous allons aujourd'hui voir comment mettre en place un réseau simple composé de 4 postes de travail, deux switchs et un routeur. Les deux switchs partageront des VLANS et le routeur se chargera des tâches de routage inter-VLANs. Nous aborderons divers fonction et manipulation sous des éléments de marques **Cisco**. Ainsi, nous régulerons plus facilement le flux (Les vlans bloquent les adresses de diffusions), nous pourrons créer des espaces de travail indépendants et la sécurité sera accrue car les flux réseau seront isolés.

## Configuration des VLANs

La première étape à suivre une fois que le câblage est en place est de créer les deux VLANs sur nos deux switchs. Pour faire simple, nous allons supposer que nous aurons deux VLANs (10 et 20) avec une liaison par port trunk entre le switch 2 et le switch 3. Le reste de la configuration sera détaillée et expliquée plus tard.

- **Les lignes de configuration suivantes sont à exécuter sur les deux switches**

```
Switch>enable
```

```
Switch#conf t
```

**Nous allons ensuite créer les VLANS et les nommer :**

```
Switch(config)#vlan 10
```

```
Switch(config-vlan)#name vlan_10
```

```
Switch(config-vlan)#vlan 20
```

```
Switch(config-vlan)#name vlan_20
```

```
Switch(config-vlan)#vlan 99
```

```
Switch(config-vlan)#name Native
```

```
Switch(config-vlan)#exit
```

**Nous allons maintenant créer nos ports trunk sur les interfaces Fa0/1 de nos deux switches. Le port trunk va permettre, au travers des trames 802.1q de faire transiter des trames tagguées (ou étiquetées) selon un Vlan ou un autre afin que tous les Vlan autorisés puissent passer au travers d'un même lien. Plus**

**Clairement, c'est un port qui peut faire passer plusieurs VLAN vers un autre élément actif. Cela permet, dans notre cas, de faire communiquer les VLANS 10 et 20 entres des éléments connectés à deux switches différents. Sans port trunk, il faudrait une liaison entre les switches par VLANs.**



```
Switch(config)#interface fa0/1
```

```
Switch(config-if)#switchport mode trunk
```

```
Switch(config-if)#switchport trunk allowed vlan 20,30,99
```

```
Switch(config-if)# switchport trunk native vlan 99
```

```
Switch(config-if)#no shutdown
```

```
Switch(config-if)#exit
```

**On spécifie également les VLANS que nous souhaitons laisser passer sur notre trunk à savoir les trames étiquetées sur les VLANs 20,30 et 99. Par défaut, toutes les VLANs peuvent passer sur un port trunk. Si nous spécifions l'autorisation de certaines VLANs, les autres ne seront pas acceptés à transiter. Nous allons maintenant affecter les ports voulus à nos différentes VLANs. On présume que nous souhaitons affecter les ports Fa0/10 des deux switchs sur le VLAN 20 et les ports Fa0/11 sur le VLAN 30, on exécute donc ces commandes sur nos deux switchs :**

```
Switch(config)#interface fa0/10
```

```
Switch(config-if)#switchport access vlan 20
```

```
Switch(config-if)#no shutdown
```

```
Switch(config-if)#exit
```

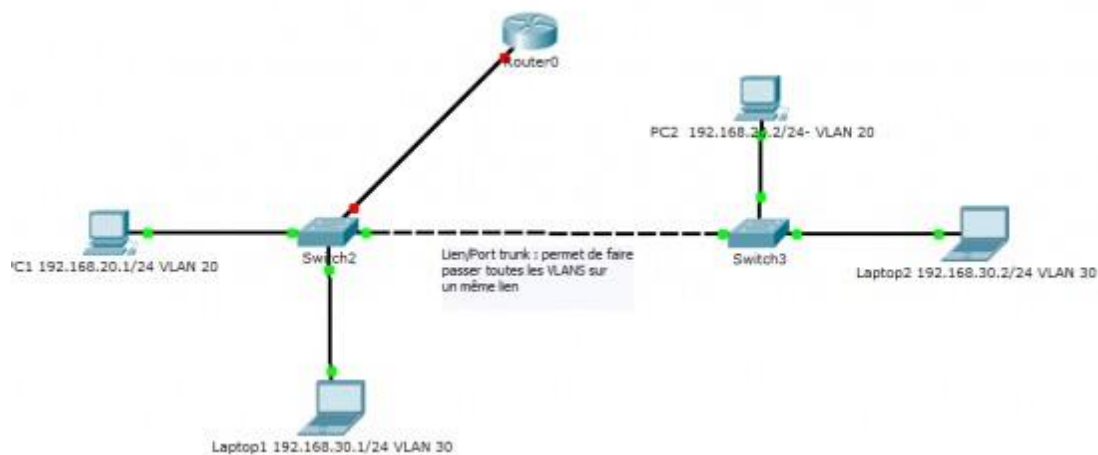
```
Switch(config)#interface fa0/11
```

```
Switch(config-if)#switchport access vlan 30
```

```
Switch(config-if)#no shutdown
```

```
Switch(config-if)#exit
```

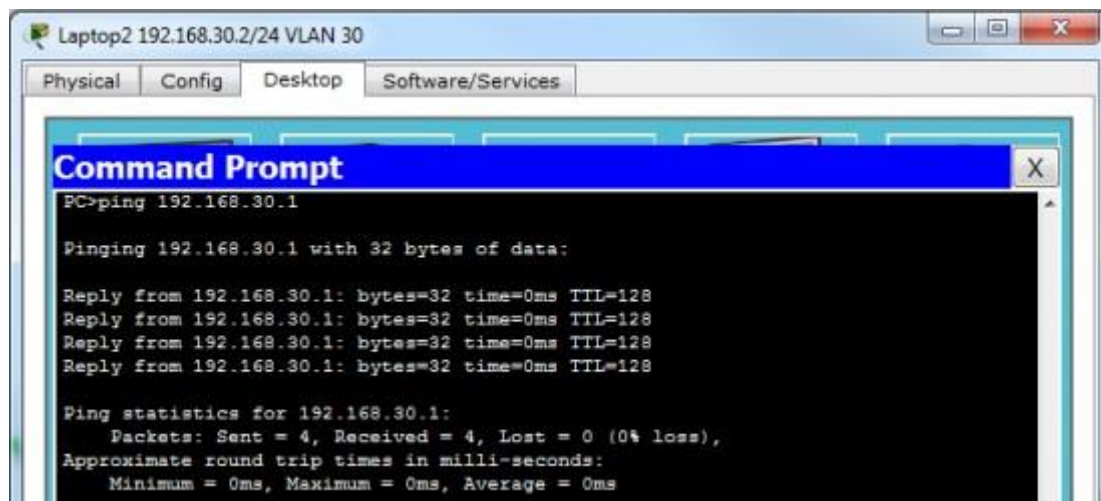
Voici à présent le schéma de notre réseau, j'ai juste ajouté des détails sur les postes pour savoir sur quel VLAN ils sont :



J'ai également ajouté des IP à mes postes. Ceux sur le VLAN 20 appartiennent au réseau "192.168.20.0/24" et ceux sur le VLAN 30 appartiennent au réseau "192.168.30.0/24"

## 🚦 Test Vlans

Nous allons maintenant tester la connectivité des postes situées sur la même VLAN. On ne prend pas exemple le poste "Laptop2" sur le VLAN 30 et avec l'IP 192.168.30.2 pour pinguer le poste "Laptop1" située sur le VLAN 30 de l'autre switch et avec l'IP 192.168.30.1 :



On peut faire la même chose en pingant du poste "PC1" au poste "PC2" qui sont également tout deux sur le même VLAN (20 cette fois ci). Les plus curieux auront remarqué que le "PC1" ou "PC2" ne peuvent pinguer "Laptop1" et "Laptop2" qui sont sur des VLANS différentes (20 et 30). C'est justement la problématique que nous venons de nous créer est que les VLANS isolent correctement les groupes de postes/d'utilisateur mais bloquent complètement la communication entre elles.

## Routage inter-vlan

Il se peut qu'un besoin de communication se fasse entre les deux groupes de travail. Il est alors possible de faire communiquer deux Vlan sans pour autant compromettre leur sécurité.

Pour cela nous utilisons un routeur relié à un des deux switches. Nous appelons ce type de routage inter-vlan un Router-on-stick. Cela signifie que le router va, par intermédiaire d'un seul lien physique router et faire transiter un ensemble de VLAN. On aurait également pu mettre en place un switch de niveau trois qui aurait été capable d'effectuer les tâches de routage inter-vlan.

Plusieurs Vlans peuvent avoir pour passerelle un même port physique du routeur qui sera "découpé" en plusieurs interfaces virtuelles. Nous pouvons en effet diviser un port du routeur selon les Vlans à router et ainsi créer une multitude de passerelles virtuelles avec des adresses IP différentes.

## Configuration du routeur

Nous allons donc créer nos interfaces virtuelles sur le port Fa0/0 de notre routeur. Il faut tout d'abord absolument activer l'interface physique pour que les interfaces virtuelles soient opérationnelles :

```
Router>enable
Router#configuration terminal
Router(config)#interface fa0/0
Router(config-if)#no shutdown
Router(config-if)#exit
```

Nous allons ensuite créer l'interface **fa0/0.1** (interface virtuelle 1 de l'interface physique fa0/0), nous dirons que ce port virtuel sur la passerelle des postes du VLAN20 :

```
Router(config)#interface fa0/0.1
Router(config-subif)#encapsulation dot1Q 20
Router(config-subif)#ip address 192.168.20.254 255.255.255.0
Router(config-subif)#no shutdown
Router(config-subif)#exit
```

Nous faisons pareil pour l'interface fa0/0.2 et les postes du réseau du vlan 30

```
Router(config)#interface fa0/0.2
Router(config-subif)#encapsulation dot1Q 30
Router(config-subif)#ip address 192.168.30.254 255.255.255.0
Router(config-subif)#no shutdown
Router(config-subif)#exit
```

La commande "**encapsulation dot1q**". La norme de trame **802.1q** indique que les trames sont étiquetées pour contenir le numéro de vlan à laquelle elles sont destinées/attribuées.

La commande "**encapsulation dot1q 30**" permet donc d'encapsuler une trame pour transiter sur le vlan 30 si elle est destinée à celui-ci. Le routeur a besoin de cette information par exemple quand il voit une trame venant du vlan 20 (étiquetée vlan 20) qui souhaite se diriger sur le vlan 30. Il change donc à ce moment-là son étiquetage 802.1q pour que les switchs puissent correctement acheminer la trame vers le ou les postes du vlan 30. N'oublions pas notre switch ! Il faut également mettre le port **fa0/24** de notre "**Switch2**" (qui fait la liaison avec le routeur) en mode trunk pour que lui aussi puisse acheminer toutes les VLANs vers et depuis le routeur

```
Switch(config)#interface fa0/24
Switch(config-if)#switchport mode trunk
Switch(config-if)#switchport trunk allowed vlan 20,30,99
Switch(config-if)# switchport trunk native vlan 99
Switch(config-if)#no shutdown
Switch(config-if)#exit
```

