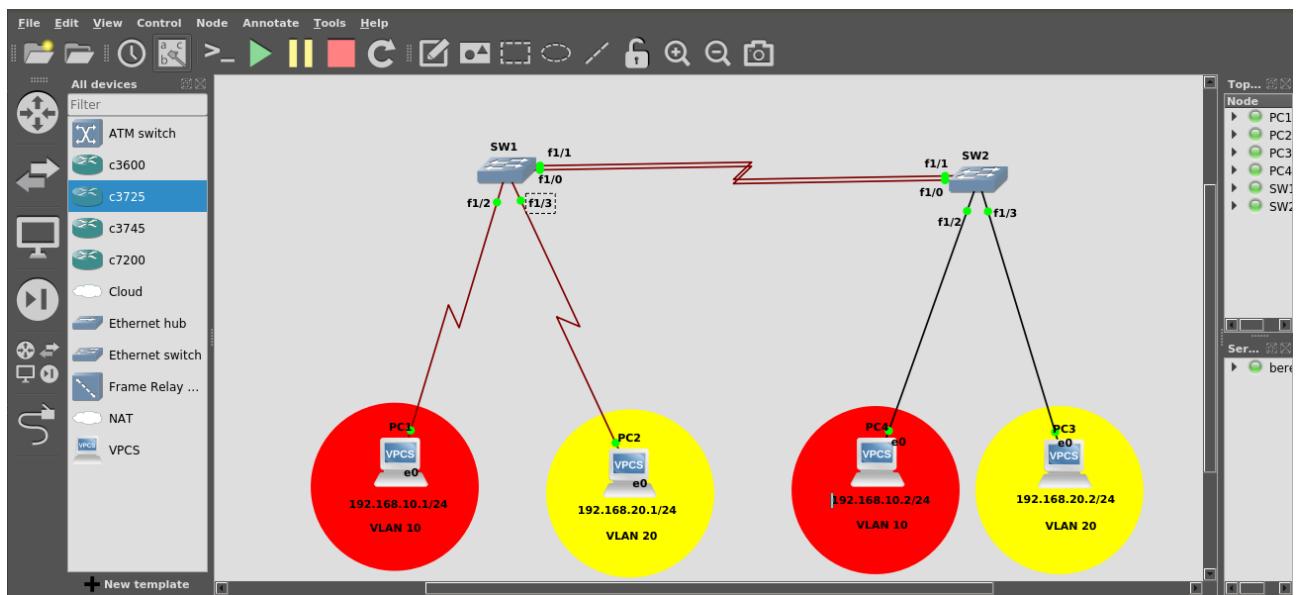


# ARCHITECTURE DES ÉQUIPEMENTS CISCO

Présenté par : BENAM BERENGER  
 Licence 3 TR  
 2021 / 2022

sous la direction de : Mr OUYA SAMUEL

**TP1 :** ce tp consiste à utiliser quelques ports des deux **switchs** pour créer des réseaux virtuels appelé VLAN. Les flux de chaque réseau virtuel empruntent un chemin différent donc au tend de réseaux virtuels que de liens entre les deux switchs. Voici l'architecture utilisé :



Architecture de déploiement d'un VLAN d'une liaison **ACCESS** Tout d'abord, nous allons créer les vlans sur les deux switchs.

- Sur l'interface de switch1

```
SW1#vlan database
% Warning: It is recommended to configure VLAN from config mode,
as VLAN database mode is being deprecated. Please consult user
documentation for configuring VTP/VLAN in config mode.

SW1(vlan)#vlan 10 name bangui
VLAN 10 added:
  Name: bangui
SW1(vlan)#vlan 20 name centrafrique
VLAN 20 added:
  Name: centrafrique
SW1(vlan)#app
SW1(vlan)#apply
APPLY completed.
SW1(vlan)#[
```

- Sur l'interface de switch2

```
SW2#vlan database
% Warning: It is recommended to configure VLAN from config mode,
as VLAN database mode is being deprecated. Please consult user
documentation for configuring VTP/VLAN in config mode.

SW2(vlan)#vlan 10 name bangui
VLAN 10 added:
  Name: bangui
SW2(vlan)#vlan 20 name centrafrique
VLAN 20 added:
  Name: centrafrique
SW2(vlan)#app
SW2(vlan)#apply
APPLY completed.
SW2(vlan)#[
```

Ensuite, nous allons affecter les ports aux vlans

- Sur l'interface de switch1

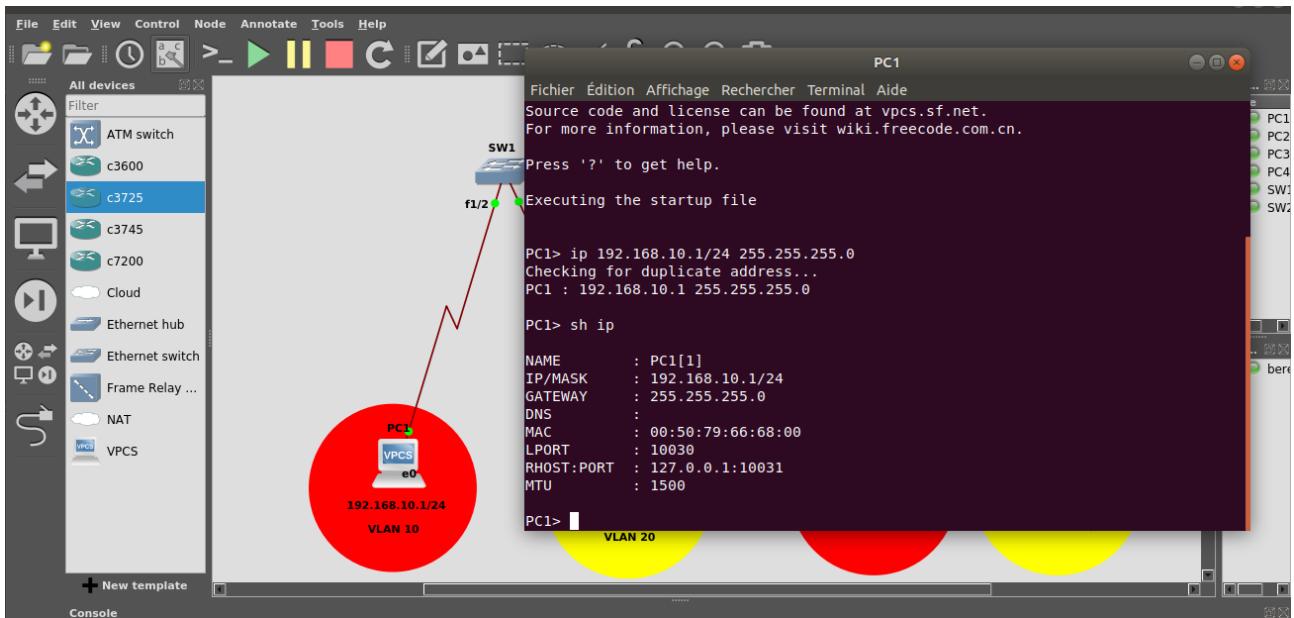
```
SW1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
SW1(config)#int f1/1
SW1(config-if)#switchport mode access
SW1(config-if)#switchport access vlan 10
SW1(config-if)#exit
SW1(config)#int f1/2
SW1(config-if)#switchport mode access
SW1(config-if)#switchport access vlan 10
SW1(config-if)#exit
SW1(config)#int f1/0
SW1(config-if)#switchport mode access
SW1(config-if)#switchport access vlan 20
SW1(config-if)#exit
*Mar 1 00:07:07.503: %CDP-4-NATIVE_VLAN_MISMATCH: Native VLAN mismatch discovered on FastEthernet1/0 (20), with SW2 FastEthernet1/0 (1).
*Mar 1 00:07:07.515: %CDP-4-NATIVE_VLAN_MISMATCH: Native VLAN mismatch discovered on FastEthernet1/1 (10), with SW2 FastEthernet1/1 (1).
SW1(config-if)#exit
SW1(config)#int f1/3
SW1(config-if)#switchport mode access
SW1(config-if)#switchport access vlan 20
SW1(config-if)#
*Mar 1 00:07:26.931: %LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan1, changed state to down
SW1(config-if)#[
```

- Sur l'interface de switch2

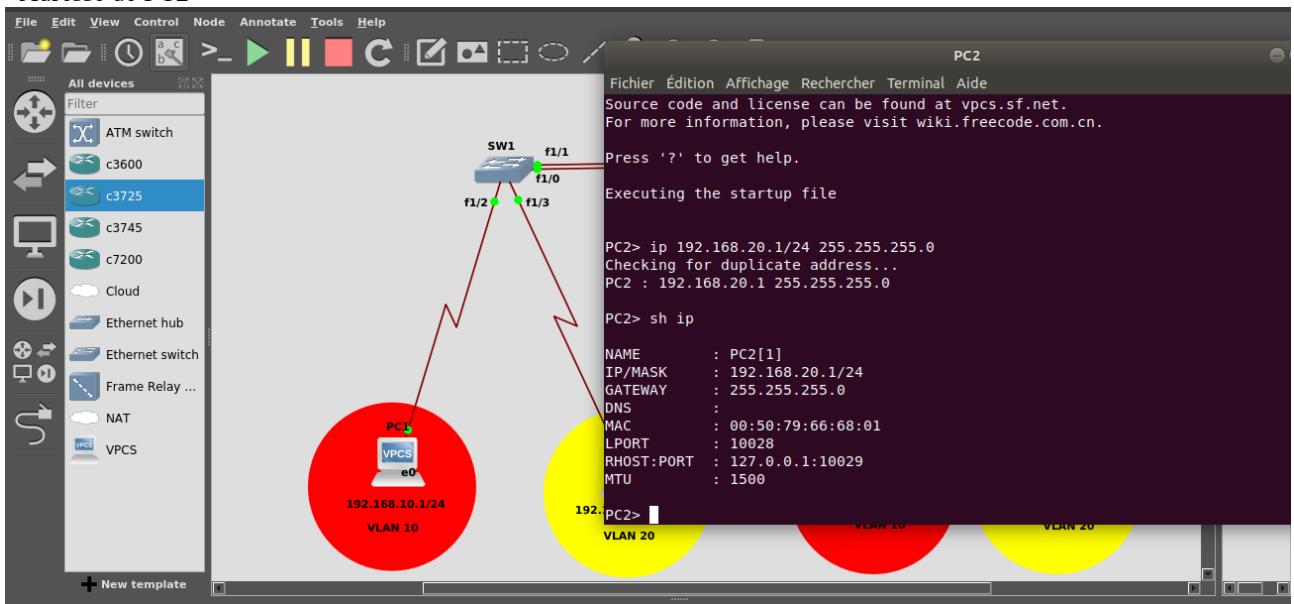
```
SW2#conf t
Enter configuration commands, one per line. End with CNTL/Z.
SW2(config)#int f1/1
SW2(config-if)#switchport mode access
SW2(config-if)#switchport access vlan 10
SW2(config-if)#
*Mar 1 00:09:07.487: %CDP-4-NATIVE_VLAN_MISMATCH: Native VLAN mismatch discovered on FastEthernet1/0 (1), with SW1 FastEthernet1/0 (20).
*Mar 1 00:09:07.495: %CDP-4-NATIVE_VLAN_MISMATCH: Native VLAN mismatch discovered on FastEthernet1/1 (1), with SW1 FastEthernet1/1 (10).
SW2(config-if)#exit
SW2(config)#int f1/2
SW2(config-if)#switchport mode access
SW2(config-if)#switchport access vlan 10
SW2(config-if)#exit
SW2(config)#int f1/0
SW2(config-if)#switchport mode access
SW2(config-if)#switchport access vlan 20
SW2(config-if)#
*Mar 1 00:10:02.099: %LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan1, changed state to down
SW2(config-if)#[
```

### Adressage des machines :

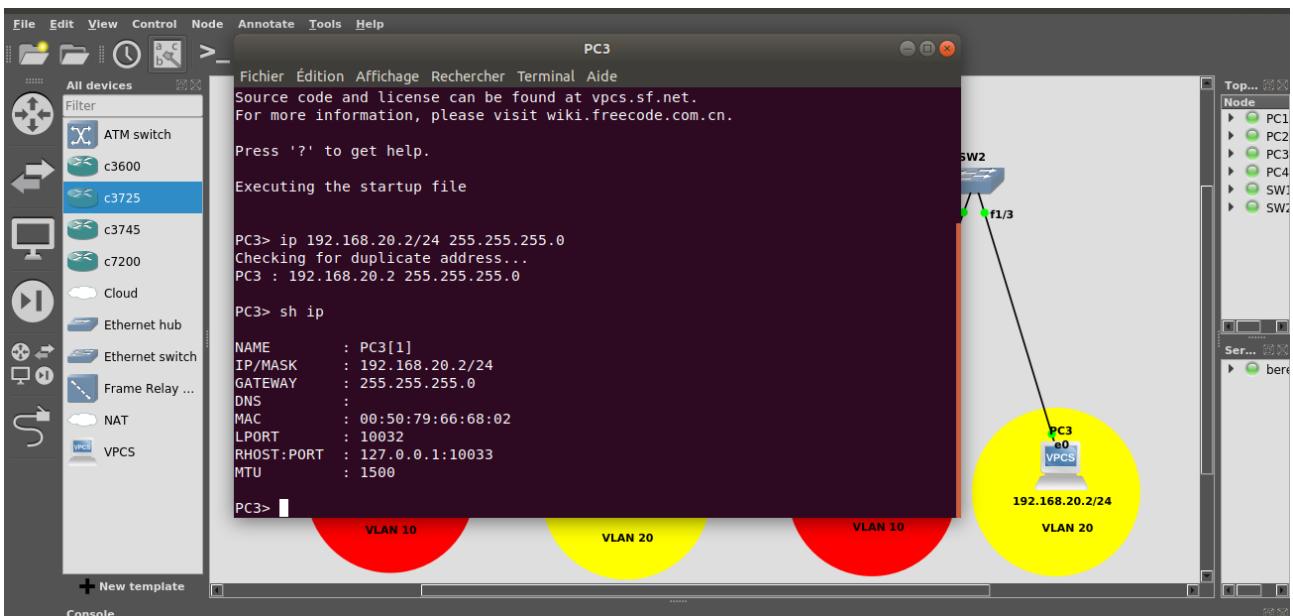
- Adresse de Pc1



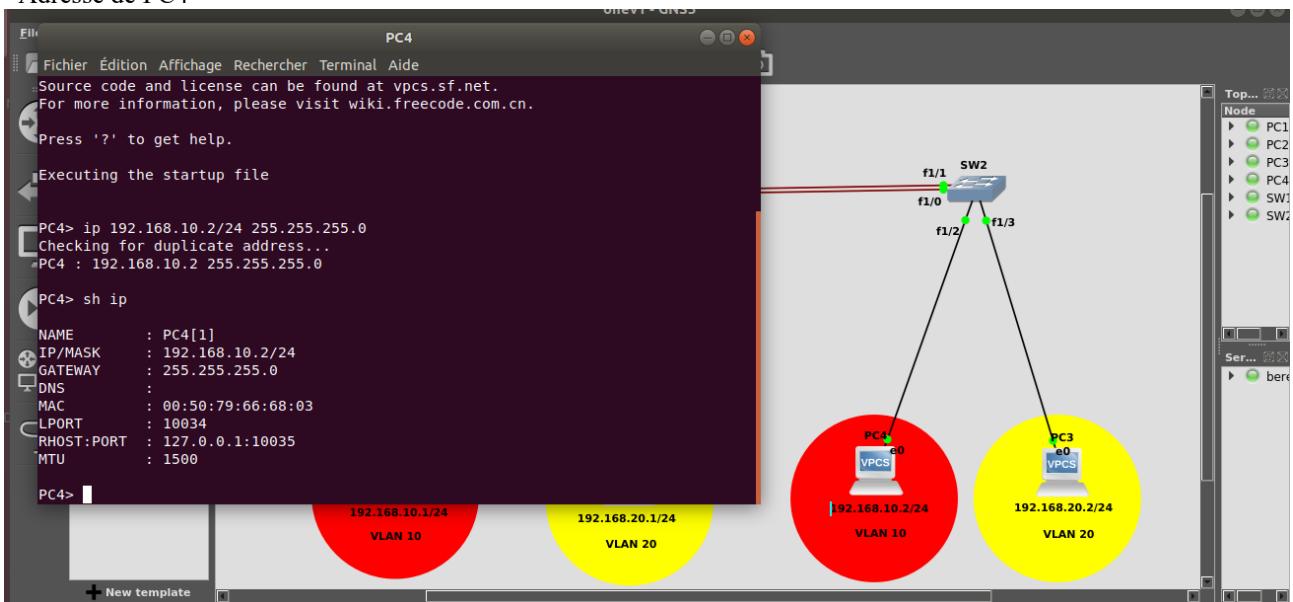
• Adresse de PC2



• Adresse de PC3



- Adresse de PC4



### Test de connectivité entre d'un même VLAN

- PC1 vers PC3

```
PC1> ping 192.168.10.2

84 bytes from 192.168.10.2 icmp_seq=1 ttl=64 time=0.223 ms
84 bytes from 192.168.10.2 icmp_seq=2 ttl=64 time=0.287 ms
84 bytes from 192.168.10.2 icmp_seq=3 ttl=64 time=0.300 ms
84 bytes from 192.168.10.2 icmp_seq=4 ttl=64 time=0.311 ms
84 bytes from 192.168.10.2 icmp_seq=5 ttl=64 time=0.324 ms

PC1> 
```

- PC2 vers PC4

```
PC2> ping 192.168.20.2

84 bytes from 192.168.20.2 icmp_seq=1 ttl=64 time=0.231 ms
84 bytes from 192.168.20.2 icmp_seq=2 ttl=64 time=0.305 ms
84 bytes from 192.168.20.2 icmp_seq=3 ttl=64 time=0.289 ms
84 bytes from 192.168.20.2 icmp_seq=4 ttl=64 time=0.304 ms
84 bytes from 192.168.20.2 icmp_seq=5 ttl=64 time=0.306 ms

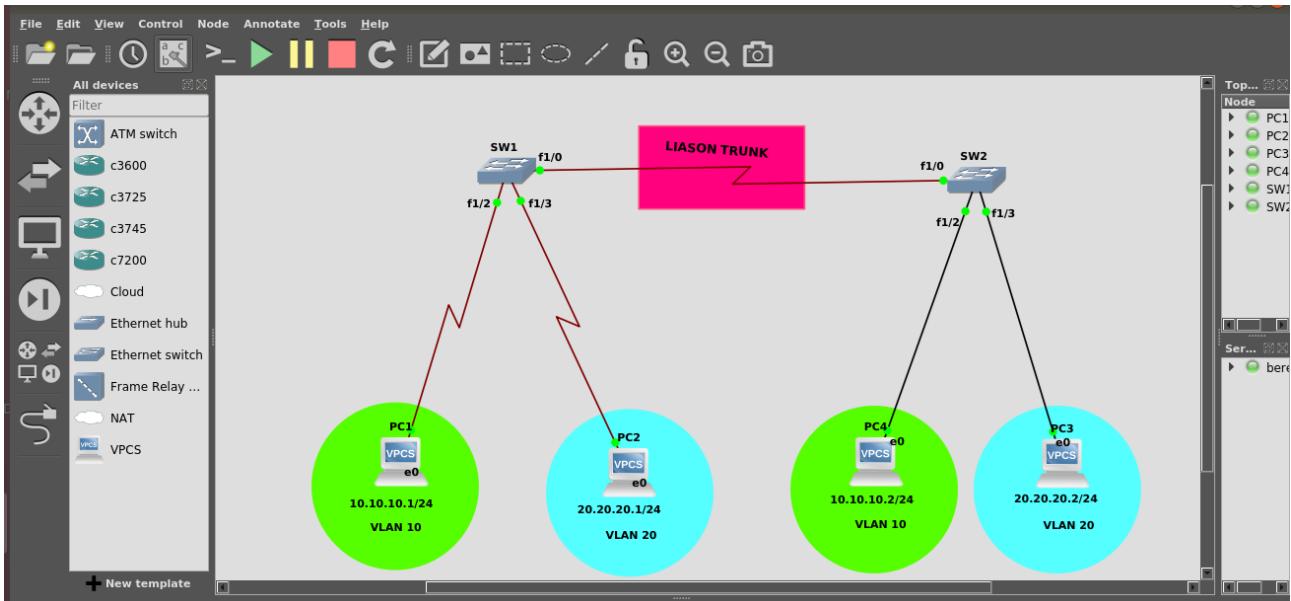
PC2> 
```

idem pour PC4 vers PC2

## 2. VLAN : Une seule liaison TRUNK pour relier les deux switchs

### Objectif :

L'objectif de ce TP est d'utiliser un lien TRUNK entre les deux switchs pour transporter les flux des différents réseaux virtuels. L'avantage d'un lien TRUNK est de pouvoir véhiculer les données des différents vlans. Il permet aussi d'économiser le câble et des ports au niveau des deux switchs. Voici notre l'architecture de déploiement :



### Architecture de déploiement Une seule liaison TRUNK

Nous allons créer les vlans sur les deux switchs

- sur l'interface de switch1 nous avons créé deux VLAN à savoir 10 et 20

```
SW1(vlan)#vlan 10 name reseaux
VLAN 10 added:
  Name: reseaux
SW1(vlan)#vlan 20 name telecoms
VLAN 20 added:
  Name: telecoms
SW1(vlan)#apply
APPLY completed.
SW1(vlan)#[
```

- sur l'interface de switch2 nous avons aussi créé deux VLAN à savoir 10 et 20

```
SW2#vlan database
% Warning: It is recommended to configure VLAN from config mode,
as VLAN database mode is being deprecated. Please consult user
documentation for configuring VTP/VLAN in config mode.

SW2(vlan)#vlan 10 name reseaux
VLAN 10 added:
  Name: reseaux
SW2(vlan)#vlan 20 name telecoms
VLAN 20 added:
  Name: telecoms
SW2(vlan)#[
```

Après la création des VLAN nous allons affecter des ports aux vlans

- Sur le switch1 :

```
SW1(config)#int f1/1
SW1(config-if)#switchport mode access
SW1(config-if)#switchport access vlan 10
SW1(config-if)#exit
SW1(config)#int f1/2
SW1(config-if)#switchport mode access
SW1(config-if)#switchport access vlan 20
SW1(config-if)#exit
SW1(config)#int f1/0
SW1(config-if)#switchport mode trunk
SW1(config-if)#[
```

- Sur le switch2 :

```

SW2#conf t
Enter configuration commands, one per line. End with CNTL/Z.
SW2(config)#int f1/1
SW2(config-if)#switchport mode access
SW2(config-if)#switchport access vlan 10
SW2(config-if)#exit
SW2(config)#int f1/2
SW2(config-if)#switchport mode access
SW2(config-if)#switchport access vlan 20
SW2(config-if)#exit
SW2(config)#int f1/0
SW2(config-if)#switchport mode trunk
SW2(config-if)#

```

Ensuite, nous allons afficher les détails du lient trunk avec la commande **show interface 0/0 switchport**

- Sur le switch1

```

SW1#show interfaces f1/0 switchport
Name: Fa1/0
Switchport: Enabled
Administrative Mode: trunk
Operational Mode: trunk
Administrative Trunking Encapsulation: dot1q
Operational Trunking Encapsulation: dot1q
Negotiation of Trunking: Disabled
Access Mode VLAN: 0 ((Inactive))
Trunking Native Mode VLAN: 1 (default)
Trunking VLANs Enabled: ALL
Trunking VLANs Active: 1,10,20
Protected: false
Priority for untagged frames: 0
Override vlan tag priority: FALSE
Voice VLAN: none
Appliance trust: none
SW1#

```

- Sur le switch2

```

SW2#show interfaces f1/0 switchport
Name: Fa1/0
Switchport: Enabled
Administrative Mode: trunk
Operational Mode: trunk
Administrative Trunking Encapsulation: dot1q
Operational Trunking Encapsulation: dot1q
Negotiation of Trunking: Disabled
Access Mode VLAN: 0 ((Inactive))
Trunking Native Mode VLAN: 1 (default)
Trunking VLANs Enabled: ALL
Trunking VLANs Active: 1,10,20
Protected: false
Priority for untagged frames: 0
Override vlan tag priority: FALSE
Voice VLAN: none
Appliance trust: none
SW2#

```

Après avoir configuré les différents **switch**, on passe à la configuration des machines

- Configuration de PC1 et PC2

```

PC1> ip 10.10.10.1/24 255.255.255.0
Checking for duplicate address...
PC1 : 10.10.10.1 255.255.255.0

PC1> sh ip

NAME      : PC1[1]
IP/MASK   : 10.10.10.1/24
GATEWAY   : 255.255.255.0
DNS       :
MAC       : 00:50:79:66:68:00
LPORT     : 10030
RHOST:PORT : 127.0.0.1:10031
MTU       : 1500

PC1>

```

```

PC2> ip 20.20.20.1/24 255.255.255.0
Checking for duplicate address...
PC2 : 20.20.20.1 255.255.255.0

PC2> sh ip

NAME      : PC2[1]
IP/MASK   : 20.20.20.1/24
GATEWAY   : 255.255.255.0
DNS       :
MAC       : 00:50:79:66:68:01
LPORT     : 10028
RHOST:PORT : 127.0.0.1:10029
MTU       : 1500

PC2>

```

- Configuration de PC3 et PC4

```

PC3> sh ip

NAME      : PC3[1]
IP/MASK   : 20.20.20.2/24
GATEWAY   : 255.255.255.0
DNS       :
MAC       : 00:50:79:66:68:02
LPORT     : 10032
RHOST:PORT : 127.0.0.1:10033
MTU       : 1500

PC3>

```

```

PC4> sh ip

NAME      : PC4[1]
IP/MASK   : 10.10.10.2/24
GATEWAY   : 255.255.255.0
DNS       :
MAC       : 00:50:79:66:68:03
LPORT     : 10034
RHOST:PORT : 127.0.0.1:10035
MTU       : 1500

PC4>

```

## Test de connectivité entre les différentes machines de chaque réseau virtuel :

- PC1 vers PC3

```
PC1> ping 10.10.10.2

84 bytes from 10.10.10.2 icmp_seq=1 ttl=64 time=0.225 ms
84 bytes from 10.10.10.2 icmp_seq=2 ttl=64 time=0.297 ms
84 bytes from 10.10.10.2 icmp_seq=3 ttl=64 time=0.295 ms
84 bytes from 10.10.10.2 icmp_seq=4 ttl=64 time=0.303 ms
84 bytes from 10.10.10.2 icmp_seq=5 ttl=64 time=0.323 ms

PC1> █
```

- PC2 vers PC4

```
PC2> ping 20.20.20.2

84 bytes from 20.20.20.2 icmp_seq=1 ttl=64 time=0.237 ms
84 bytes from 20.20.20.2 icmp_seq=2 ttl=64 time=0.309 ms
84 bytes from 20.20.20.2 icmp_seq=3 ttl=64 time=0.312 ms
84 bytes from 20.20.20.2 icmp_seq=4 ttl=64 time=0.308 ms
84 bytes from 20.20.20.2 icmp_seq=5 ttl=64 time=0.314 ms

PC2> █
```

- PC4 vers PC1

```
PC4> ping 10.10.10.1

84 bytes from 10.10.10.1 icmp_seq=1 ttl=64 time=0.311 ms
84 bytes from 10.10.10.1 icmp_seq=2 ttl=64 time=0.291 ms
84 bytes from 10.10.10.1 icmp_seq=3 ttl=64 time=0.339 ms
84 bytes from 10.10.10.1 icmp_seq=4 ttl=64 time=0.316 ms
84 bytes from 10.10.10.1 icmp_seq=5 ttl=64 time=0.305 ms

PC4> █
```

pc3 vers pc2

```
PC3> ping 20.20.20.1

84 bytes from 20.20.20.1 icmp_seq=1 ttl=64 time=0.240 ms
84 bytes from 20.20.20.1 icmp_seq=2 ttl=64 time=0.322 ms
84 bytes from 20.20.20.1 icmp_seq=3 ttl=64 time=0.288 ms
84 bytes from 20.20.20.1 icmp_seq=4 ttl=64 time=0.296 ms
84 bytes from 20.20.20.1 icmp_seq=5 ttl=64 time=0.300 ms

PC3> █
```

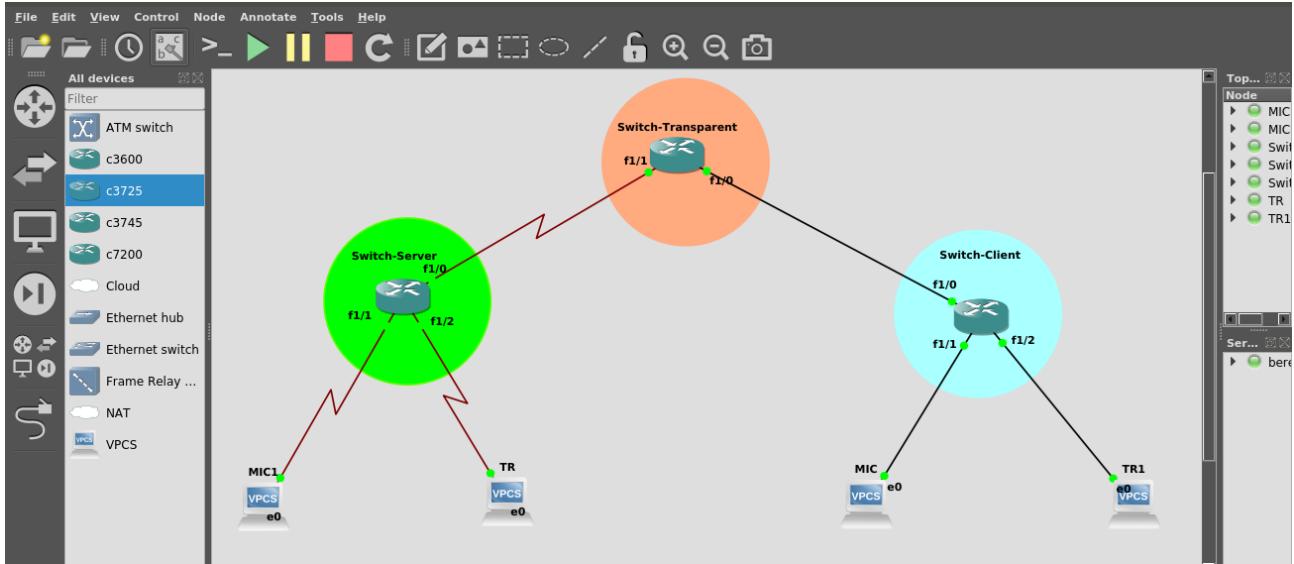
## 3. Mise en œuvre des VLAN avec VTP

### Objectif :

Ce TP consiste à mettre en place un VTP (VLAN Trunking Protocol), protocole propriétaire Cisco permet, aux commutateurs qui l'implémentent, d'échanger des informations de configuration des VLAN. Le trunking sert dans l'extension d'un domaine VLAN sur d'autres switch. Un lien qui relie deux interfaces configurées en mode Trunk permet

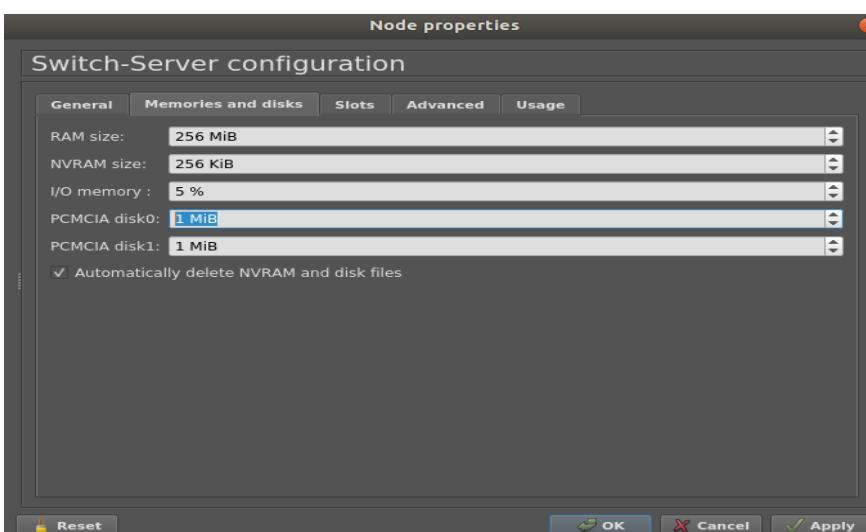
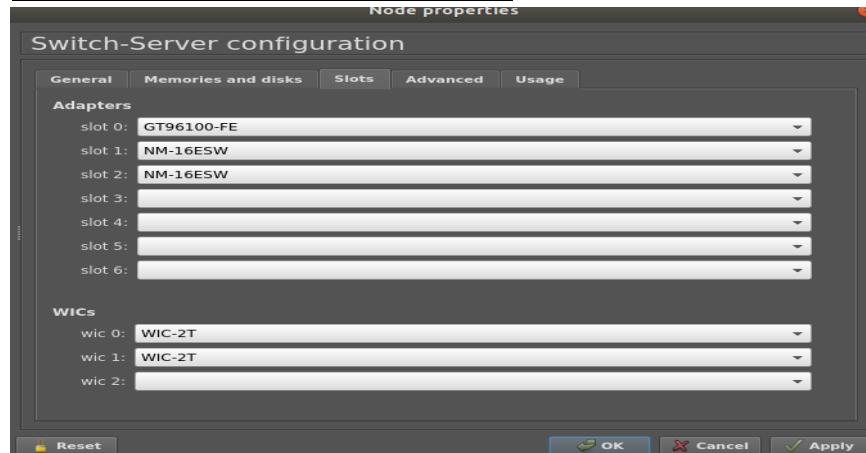
d'étendre les VLANs entre deux switches (transporte le trafic de plusieurs VLANs) et aussi permet de transporter les messages VTP véhiculés.

automatiquement entre les switches pour signaler la mise à jour de la liste des VLANs. Voici l'architecture :



## Architecture de déploiement des VLAN avec VTP

- Configuration du switch-server**



Après avoir augmenté les mémoires disk et flash se trouvant dans la configuration du **switch** à 1 Mo et formater le disque grâce à la commande **erase flash** :

```
switch-server#erase flash:  
Erasing the flash filesystem will remove all files! Continue? [confirm]  
Current DOS File System flash card in flash: will be formatted into Low End File  
System flash card! Continue? [confirm]  
Erasing device... eeeeeeeeeeeeeeee ...erased  
Erase of flash: complete  
switch-server#
```

Ensuite, nous allons passer à la configuration de celui-ci en définissant le domaine **VTP** et son mot de passe comme suit :

```
Switch-Server#conf t  
Enter configuration commands, one per line. End with CNTL/Z.  
Switch-Server(config)#vtp mode server  
Device mode already VTP SERVER.  
Switch-Server(config)#vtp domain telecoms  
Changing VTP domain name from NULL to telecoms  
Switch-Server(config)#vtp password passer  
Setting device VLAN database password to passer  
Switch-Server(config)#
```

Puis, nous allons afficher la configuration VTP

```
Switch-Server#show vtp status  
VTP Version : 2  
Configuration Revision : 0  
Maximum VLANs supported locally : 36  
Number of existing VLANs : 5  
VTP Operating Mode : Server  
VTP Domain Name : telecoms  
VTP Pruning Mode : Disabled  
VTP V2 Mode : Disabled  
VTP Traps Generation : Disabled  
MD5 digest : 0x26 0xBB 0x6D 0xA4 0xEE 0x59 0x6E 0x78  
Configuration last modified by 0.0.0.0 at 0-0-00 00:00:00  
Local updater ID is 0.0.0.0 (no valid interface found)  
Switch-Server#
```

On voit tous les informations y compris le nom de domain telecoms qu'on a crée, on va créer nos vlans au niveau du switch.

- création des VLAN 10 et 20

```
Switch-Server#vlan database  
% Warning: It is recommended to configure VLAN from config mode,  
as VLAN database mode is being deprecated. Please consult user  
documentation for configuring VTP/VLAN in config mode.  
  
Switch-Server(vlan)#vlan 10  
VLAN 10 added:  
    Name: VLAN0010  
Switch-Server(vlan)#vlan 20  
VLAN 20 added:  
    Name: VLAN0020  
Switch-Server(vlan)#apply  
APPLY completed.  
Switch-Server(vlan)#
```

Cependant, nous allons affecter chaque interface à un VLAN

- L'interface f1/1 affectée au VLAN 10

```
Switch-Server#conf t  
Enter configuration commands, one per line. End with CNTL/Z.  
Switch-Server(config)#int f1/1  
Switch-Server(config-if)#switchport mode access  
Switch-Server(config-if)#switchport access vlan 10  
Switch-Server(config-if)#
```

- L'interface f1/2 affectée au VLAN 20

```
Switch-Server#conf t  
Enter configuration commands, one per line. End with CNTL/Z.  
Switch-Server(config)#int f1/1  
Switch-Server(config-if)#switchport mode access  
Switch-Server(config-if)#switchport access vlan 10  
Switch-Server(config-if)#int f1/2  
Switch-Server(config-if)#switchport mode access  
Switch-Server(config-if)#switchport access vlan 20  
Switch-Server(config-if)#
```

- L'interface f1/0 est celle qui relie les deux switchs et est définie en mode trunk

```
Switch-Server(config-if)#
Switch-Server(config-if)#exit
Switch-Server(config)#int f1/0
Switch-Server(config-if)#switchport mode trunk
Switch-Server(config-if)#
*Mar 1 00:17:38.083: %DTP-5-TRUNKPORTON: Port Fa1/0 has become dot1q trunk
Switch-Server(config-if)#switchport trunk allowed vlan 10,20
Command rejected: Bad VLAN allowed list. You have to include all default vlans, e.g. 1-2,1002-1005.
Switch-Server(config-if)#[
```

affichage des différents VLAN avec leurs interfaces

```
Switch-Server#show vlan-switch
VLAN Name          Status    Ports
----- -----
1   default        active    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
10  VLAN0010       active    Fa1/1
20  VLAN0020       active    Fa1/2
1002 fddi-default  act/unsup
1003 token-ring-default  act/unsup
1004 fddinet-default  act/unsup
1005 trnet-default   act/unsup
VLAN Type  SAID      MTU    Parent RingNo BridgeNo Stp  BrdgMode Trans1 Trans2
----- -----
1   enet  100001    1500   -     -     -     -     1002  1003
10  enet  100010    1500   -     -     -     -     0     0
20  enet  100020    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
Switch-Server#[
```

Affichage du status de VTP pour voir la révision

```
Switch-Server#show vtp status
VTP Version          : 2
Configuration Revision : 1
Maximum VLANs supported locally : 36
Number of existing VLANs      : 7
VTP Operating Mode       : Server
VTP Domain Name         : telecoms
VTP Pruning Mode        : Disabled
VTP V2 Mode             : Disabled
VTP Traps Generation    : Disabled
MD5 digest              : 0x44 0xA2 0x98 0xD3 0xE9 0xF6 0x4F 0xB4
Configuration last modified by 0.0.0.0 at 3-1-02 00:10:31
Local updater ID is 0.0.0.0 (no valid interface found)
Switch-Server#[
```

On voit bien qu'il est passé à 1

## ● Configuration du switch-transparent

Tout d'abord, nous allons formater le disque et mettre le **switch-transparent** en mode transparent comme suit.

```
Switch-Transparent#erase flash:
Erasing the flash filesystem will remove all files! Continue? [confirm]
Current DOS File System flash card in flash: will be formatted into Low End File System flash card! Continue? [confirm]
Erase device... eeeeeeeeeeeeeeee ...erased
Erase of flash: complete
Switch-Transparent#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Switch-Transparent(config)#vtp mode transparent
Setting device to VTP TRANSPARENT mode.
Switch-Transparent(config)#[
```

On crée ensuite le même domaine créer sur le **switch-server** et on lui attribue le même mot de passe comme suit :

```
Switch-Transparent(config)#vtp domain telecoms
Changing VTP domain name from NULL to telecoms
Switch-Transparent(config)#vtp password passer
Setting device VLAN database password to passer
Switch-Transparent(config)#[
```

Puis, on met les deux interface (fa 1/1 et fa 1/0) en mode trunk

```
Switch-Transparent(config)#int fa1/1
Switch-Transparent(config-if)#switchport mode trunk
Switch-Transparent(config-if)#
*Mar 1 00:37:20.071: %DTP-5-TRUNKPORTON: Port Fa1/1 has become dot1q trunk
Switch-Transparent(config-if)#switchport trunk allowed vlan 10,20
Command rejected: Bad VLAN allowed list. You have to include all default vlans, e.g. 1-2,1002-1005.
Switch-Transparent(config-if)#[
```

```

Switch-Transparent(config)#int f1/0
Switch-Transparent(config-if)#switchport mode trunk
Switch-Transparent(config-if)#
*Mar 1 00:38:48.803: %DTP-5-TRUNKPORTON: Port Fa1/0 has become dot1q trunk
Switch-Transparent(config-if)#switchport trunk allowed vlan 10,20
Command rejected: Bad VLAN allowed list. You have to include all default vlans, e.g. 1-2,1002-1005.
Switch-Transparent(config-if)#

```

## ● Configuration du switch-client

Nous allons regarder le status du vtp

```

Switch-Client#show vtp status
VTP Version           : 2
Configuration Revision : 0
Maximum VLANs supported locally : 36
Number of existing VLANs      : 5
VTP Operating Mode        : Server
VTP Domain Name          :
VTP Pruning Mode         : Disabled
VTP V2 Mode               : Disabled
VTP Traps Generation     : Disabled
MD5 digest              : 0xBF 0x86 0x94 0x45 0xFC 0xDF 0xB5 0x70
Configuration last modified by 0.0.0.0 at 0-0-00 00:00:00
Local updater ID is 0.0.0.0 (no valid interface found)
Switch-Client#

```

Ensuite, nous allons le configurer en mode client et dans le même domaine que switch-server et donnons également le même mot de passe.

```

Switch-Client#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Switch-Client(config)#vtp mode client
Setting device to VTP CLIENT mode.
Switch-Client(config)#vtp domain telecoms
Changing VTP domain name from NULL to telecoms
Switch-Client(config)#vtp password passer
Setting device VLAN database password to passer
Switch-Client(config)#

```

Nous avons mis le port fa 1/0 du switch-client en mode trunk et attendu quelques secondes pour visualiser les vlans créés sur le switch-server

```

switch-client#show vlan-switch

VLAN Name                Status    Ports
---- -----
1   default               active    Fa1/1, 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
10  VLAN0010              active
20  VLAN0020              active
1002 fddi-default         act/unsup
1003 token-ring-default   act/unsup
1004 fddinet-default      act/unsup
1005 trnet-default        act/unsup

VLAN Type    SAID      MTU    Parent RingNo BridgeNo Stp  BrdgMode Trans1 Trans2
---- -----
1   enet     100001    1500   -     -     -     -     -     1002  1003
10  enet     100010    1500   -     -     -     -     -     0     0
20  enet     100020    1500   -     -     -     -     -     0     0
1002 fddi    101002    1500   -     0     -     -     -     1     1003
1003 tr      101003    1500   1005  0     -     -     srb   1     1002
1004 fdnet   101004    1500   -     -     1     ibm  -     0     0
1005 trnet   101005    1500   -     -     -     ibm  -     0     0
switch-client#

```

Nous constatons que le switch-client a les mêmes configurations que celui du switch-server. De ce fait, nous allons nous rendre à nouveau sur le switch-server et attribuer à chaque vlan une adresse IP.

```
Switch-Server#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Switch-Server(config)#int vlan 10
Switch-Server(config-if)#ip address 192.168.10.254 255.255.255.0
Switch-Server(config-if)#no sh
Switch-Server(config-if)#[
```

```
Switch-Server(config-if)#exit
Switch-Server(config)#int vlan 20
Switch-Server(config-if)#ip
*Mar 1 00:56:05.815: %LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan20, changed state to up
Switch-Server(config-if)#ip address 192.168.20.254 255.255.255.0
Switch-Server(config-if)#no sh
Switch-Server(config-if)#[
```

Ensuite, on créer un VLAN nommé vlan 30

```
Switch-Server#vlan database
% Warning: It is recommended to configure VLAN from config mode,
as VLAN database mode is being deprecated. Please consult user
documentation for configuring VTP/VLAN in config mode.

Switch-Server(vlan)#vlan 30
VLAN 30 added:
Name: VLAN0030
Switch-Server(vlan)#apply
APPLY completed.
Switch-Server(vlan)#[
```

On lui affecte le port f1/3 du switch-server au vlan 30

```
Switch-Server#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Switch-Server(config)#int f1/3
Switch-Server(config-if)# switchport mode access
Switch-Server(config-if)# switchport access vlan 30
Switch-Server(config-if)#[
```

Puis, on visualise le status du vtp

```
Switch-Server#show vtp status
VTP Version : 2
Configuration Revision : 2
Maximum VLANs supported locally : 36
Number of existing VLANs : 8
VTP Operating Mode : Server
VTP Domain Name : telecoms
VTP Pruning Mode : Disabled
VTP V2 Mode : Disabled
VTP Traps Generation : Disabled
MD5 digest : 0x81 0xB4 0xAE 0x2A 0x6B 0xA4 0x62 0x73
Configuration last modified by 192.168.10.254 at 3-1-02 00:58:00
Local updater ID is 192.168.10.254 on interface Vl10 (lowest numbered VLAN interface found)
Switch-Server#[
```

On constate que la valeur du RN est passée à 2 ; ceci est dû à la modification faite sur les vlans.  
Nous allons visualiser le status du vtp sur le switch-client

```
Switch-Client#show vtp status
VTP Version : 2
Configuration Revision : 0
Maximum VLANs supported locally : 36
Number of existing VLANs : 5
VTP Operating Mode : Client
VTP Domain Name : telecoms
VTP Pruning Mode : Disabled
VTP V2 Mode : Disabled
VTP Traps Generation : Disabled
MD5 digest : 0x26 0xBB 0x6D 0xA4 0xEE 0x59 0x6E 0x78
Configuration last modified by 0.0.0.0 at 0-0-00 00:00:00
Switch-Client#[
```

Nous allons afficher les vlans pour voir

```

switch-client#show vlan-switch

VLAN Name                               Status    Ports
---- -----
1   default                             active    Fa1/1, 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
10  VLAN0010                           active
20  VLAN0020                           active
30  VLAN0030                           active
1002 fddi-default                      act/unsup
1003 token-ring-default                act/unsup
1004 fddinet-default                  act/unsup
1005 trnet-default                    act/unsup

VLAN Type     SAID      MTU    Parent RingNo BridgeNo Stp  BrdgMode Trans1 Trans2
---- -----
1   enet      100001   1500   -       -       -       -       -       1002   1003
10  enet      100010   1500   -       -       -       -       -       0      0
20  enet      100020   1500   -       -       -       -       -       0      0
30  enet      100030   1500   -       -       -       -       -       0      0
1002 fddi     101002   1500   -       0       -       -       -       1      1003
1003 tr      101003   1500   1005   0       -       -       srb    1      1002
--More-- 

```

On constate qu'il y a les mêmes configurations

Visualisons également les vlans et le status vtp sur le switch-transparent :

```

Switch-Transparent#show vtp status
VTP Version          : 2
Configuration Revision : 0
Maximum VLANs supported locally : 36
Number of existing VLANs      : 5
VTP Operating Mode       : Transparent
VTP Domain Name         : telecoms
VTP Pruning Mode        : Disabled
VTP V2 Mode             : Disabled
VTP Traps Generation    : Disabled
MD5 digest              : 0x26 0xBB 0x6D 0xA4 0xEE 0x59 0x6E 0x78
Configuration last modified by 0.0.0.0 at 0-0-00 00:00:00
Switch-Transparent#

```

```

Switch-Transparent#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
1002 fddi-default                      act/unsup
1003 token-ring-default                act/unsup
1004 fddinet-default                  act/unsup
1005 trnet-default                    act/unsup

VLAN Type     SAID      MTU    Parent RingNo BridgeNo Stp  BrdgMode Trans1 Trans2
---- -----
1   enet      100001   1500   -       -       -       1002   1003
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
Switch-Transparent#

```

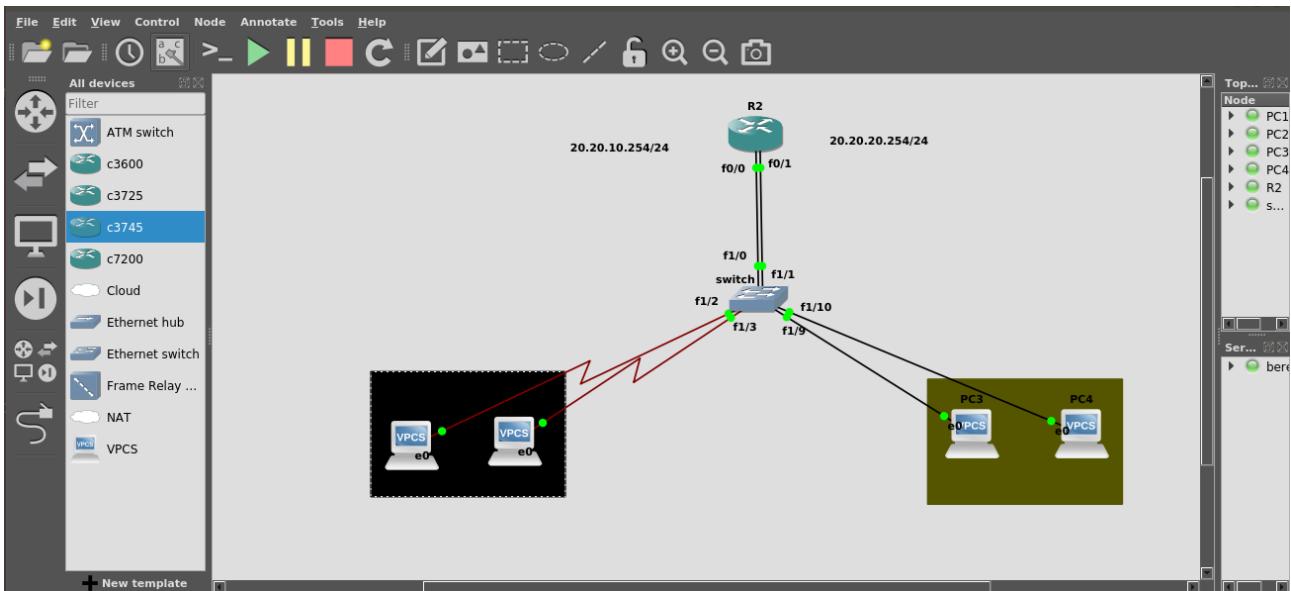
On constate aussi les mêmes configurations comme celles du switch-server

## 4. Routage inter-vlan classique

### Objectif :

Ce TP consiste à mettre en place du routage inter-vlan pour faire communiquer les deux réseaux.

Dans ce TP nous utilisons un équipement de niveau 3 c'est-à-dire un routeur qui a deux interfaces et chaque interface sera affecter à un vlan comme le montre l'architecture ci-dessous :



### Architecture de déploiement d'un routage inter-vlan

Cependant, nous allons créer les Vlans 10 et 20

```
switch#erase fl
switch#erase flash:
Erasing the flash filesystem will remove all files! Continue? [confirm]
Current DOS File System flash card in flash: will be formatted into Low End File System flash card! Continue? [confirm]
Erasing device... eeeeeeeeeeeeeeee ...erased
Erase of flash: complete
switch#vlan database
% Warning: It is recommended to configure VLAN from config mode,
as VLAN database mode is being deprecated. Please consult user
documentation for configuring VTP/VLAN in config mode.

switch(vlan)#vlan 10 name berenger
VLAN 10 added:
  Name: berenger
switch(vlan)#vlan 20 name deranger
VLAN 20 added:
  Name: deranger
switch(vlan)#apply
APPLY completed.
switch(vlan)#exit
APPLY completed.
Exiting...
switch#
```

Nous allons affecter les interfaces de f1/2 à f1/8 au vlan 10 et les interfaces de f1/9 à f1/15 au vlan 20

```
switch#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
switch(config)#int range f1/2 - 8
switch(config-if-range)#switchport mode access
switch(config-if-range)#switchport access vlan 10
switch(config-if-range)#exit
switch(config)#int range f1/9 - 15
switch(config-if-range)#switchport mode access
switch(config-if-range)#switchport access vlan 20
switch(config-if-range)#exit
switch(config)#
```

Ensuite, on affecter et l'interface f1/0 et f1/1 au vlan 10 et 20

```

switch(config)#int f1/0
switch(config-if)#switchport mode access
switch(config-if)#switchport access vlan 10
switch(config-if)#exit
switch(config)#int f1/1
switch(config-if)#switchport mode access
switch(config-if)#switchport access vlan 20
switch(config-if)#exit
switch(config)#
*Mar 1 00:06:07.671: %LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan1, changed state to down
switch(config)#

```

## Affichage des VLANs

```

switch#show vlan-switch

VLAN Name          Status    Ports
-----  -----
1   default         active    Fa1/0, Fa1/2, Fa1/3, Fa1/4
10  berenger        active    Fa1/5, Fa1/6, Fa1/7, Fa1/8
20  deranger        active    Fa1/1, Fa1/9, Fa1/10, Fa1/11
                                Fa1/12, Fa1/13, Fa1/14, Fa1/15
1002 fddi-default  act/unsup
1003 token-ring-default  act/unsup
1004 fdnet-default   act/unsup
1005 trnet-default   act/unsup

VLAN Type   SAID      MTU    Parent RingNo BridgeNo Stp   BrdgMode Trans1 Trans2
-----  -----
1   enet     100001   1500   -       -       -       -       1002   1003
10  enet     100010   1500   -       -       -       -       0       0
20  enet     100020   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
switch#

```

Puis, nous allons donner une adresse IP à l'interface f0/0 qui constitue une passerelle par défaut du vlan 10 ensuite activer l'interface.

```

R2#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R2(config)#int f0/0
R2(config-if)#ip add 20.20.10.254 255.255.255.0
R2(config-if)#no sh
R2(config-if)#
*Mar 1 00:09:02.083: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state to up
*Mar 1 00:09:03.083: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
R2(config-if)#exit
R2(config)#

```

Puis, on donne une adresse IP à l'interface f0/1 qui constitue une passerelle par défaut du vlan 20 ensuite activer l'interface.

```

R2(config-if)#exit
R2(config)#int f0/1
R2(config-if)#ip add 20.20.20.254 255.255.255.0
R2(config-if)#no sh
R2(config-if)#
*Mar 1 00:10:28.503: %LINK-3-UPDOWN: Interface FastEthernet0/1, changed state to up
*Mar 1 00:10:29.503: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1, changed state to up
R2(config-if)#

```

Nous allons par la suite donner des adresses IP aux différentes machines des VLANs 10 et 20

- Adressage de PC1 du VLAN 10

```

PC1> ip 20.20.10.1/24 20.20.10.254
Checking for duplicate address...
PC1 : 20.20.10.1 255.255.255.0 gateway 20.20.10.254

```

PC1>

Adressage de PC2 du VLAN 10

```

PC2> ip 20.20.10.2/24 20.20.10.254
Checking for duplicate address...
PC2 : 20.20.10.2 255.255.255.0 gateway 20.20.10.254

```

PC2>

Adressage de PC3 du VLAN 20

```
PC3> ip 20.20.20.1/24 20.20.20.254
Checking for duplicate address...
PC3 : 20.20.20.1 255.255.255.0 gateway 20.20.20.254

PC3> █
```

- Adressage de PC4 du VLAN 20

```
PC4> ip 20.20.20.2/24 20.20.20.254
Checking for duplicate address...
PC4 : 20.20.20.2 255.255.255.0 gateway 20.20.20.254

PC4> █
```

Test de connectivité entre PC2 et PC3 ; PC2 et PC4

```
PC2> ping 20.20.20.2

20.20.20.2 icmp_seq=1 timeout
84 bytes from 20.20.20.2 icmp_seq=2 ttl=63 time=19.959 ms
84 bytes from 20.20.20.2 icmp_seq=3 ttl=63 time=15.390 ms
84 bytes from 20.20.20.2 icmp_seq=4 ttl=63 time=14.922 ms
84 bytes from 20.20.20.2 icmp_seq=5 ttl=63 time=14.578 ms

PC2> ping 20.20.20.1

20.20.20.1 icmp_seq=1 timeout
84 bytes from 20.20.20.1 icmp_seq=2 ttl=63 time=19.093 ms
84 bytes from 20.20.20.1 icmp_seq=3 ttl=63 time=15.672 ms
84 bytes from 20.20.20.1 icmp_seq=4 ttl=63 time=17.076 ms
84 bytes from 20.20.20.1 icmp_seq=5 ttl=63 time=17.284 ms

PC2> █
```

Test de connectivité entre PC4 et PC1 ; PC4 et PC2

```
PC4> ping 20.20.10.2

84 bytes from 20.20.10.2 icmp_seq=1 ttl=63 time=15.491 ms
84 bytes from 20.20.10.2 icmp_seq=2 ttl=63 time=15.307 ms
84 bytes from 20.20.10.2 icmp_seq=3 ttl=63 time=14.914 ms
84 bytes from 20.20.10.2 icmp_seq=4 ttl=63 time=14.830 ms
84 bytes from 20.20.10.2 icmp_seq=5 ttl=63 time=14.625 ms

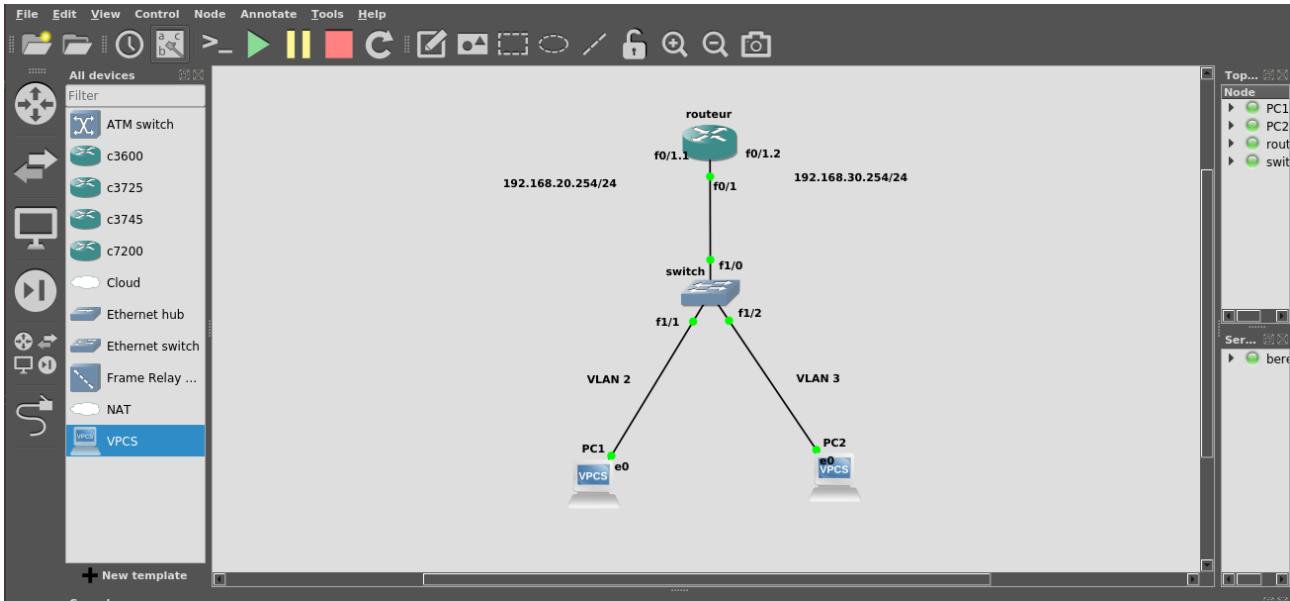
PC4> ping 20.20.10.1

20.20.10.1 icmp_seq=1 timeout
84 bytes from 20.20.10.1 icmp_seq=2 ttl=63 time=18.450 ms
84 bytes from 20.20.10.1 icmp_seq=3 ttl=63 time=19.445 ms
84 bytes from 20.20.10.1 icmp_seq=4 ttl=63 time=17.224 ms
84 bytes from 20.20.10.1 icmp_seq=5 ttl=63 time=18.347 ms

PC4> █
```

## 5. Routage Inter-vlan avec un Router On a Stick

Ce TP consiste à faire appel à un équipement de niveau 3 pour faire communiquer les différents réseaux virtuels. Vue que le switch ne peut être connecté qu'à une seule interface du routeur donc il faut des interfaces virtuelles comme le montre l'architecture ci-dessous :



## Architecture de déploiement routage Inter-vlan avec un Router On a Stick

### • Configuration de switch

Tout d'abord, nous allons créer les VLANs 2 et 3 après avoir formaté le disque

```
switch#erase fl
switch#erase flash:
Erasing the flash filesystem will remove all files! Continue? [confirm]
Current DOS File System flash card in flash: will be formatted into Low End File System flash card! Continue? [confirm]
Erasing device... eeeeeeeeeeeeeeee ...erased
Erase of flash: complete
switch#
switch#vlan database
switch(vlan)#vlan 2
VLAN 2 added:
  Name: VLAN0002
switch(vlan)#vlan 3
VLAN 3 added:
  Name: VLAN0003
switch(vlan)#apply
APPLY completed.
switch(vlan)#[
```

Ensuite, on attribue respectivement les ports f1/1 et f1/2 aux vlans 2 et 3

```
switch#conf t
Enter configuration commands, one per line. End with CNTL/Z.
switch(config)#int f1/1
switch(config-if)#switchport mode access
switch(config-if)#switchport access vlan 2
switch(config-if)#exit
switch(config)#int f1/2
switch(config-if)#switchport mode access
switch(config-if)#switchport access vlan 3
switch(config-if)#[
```

Puis, nous allons activer le trunk sur le port f1/0 qui relie le routeur au switch puis enregistrer

```
switch(config-if)#exit
switch(config)#int f1/0
switch(config-if)#switchport mode trunk
switch(config-if)#
*Mar 1 00:04:16.963: %DTP-5-TRUNKPORTON: Port Fa1/0 has become dot1q trunk
switch(config-if)#switchport trunk allowed vlan 2,3
Command rejected: Bad VLAN allowed list. You have to include all default vlans, e.g. 1-2,1002-1005.
switch(config-if)#[
```

### • Configuration du routeur

Nous allons activer l'interface f0/1 du routeur relié au switch

```
routeur#conf t
Enter configuration commands, one per line. End with CNTL/Z.
routeur(config)#int f0/1
routeur(config-if)#no sh
routeur(config-if)#
*Mar 1 00:06:46.299: %LINK-3-UPDOWN: Interface FastEthernet0/1, changed state to up
*Mar 1 00:06:47.299: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1, changed state to up
routeur(config-if)#exit
routeur(config)#[
```

Ensuite, on crée la sous interface f0/1.1 sur le routeur et nous allons lui attribuer le vlan 2 et une adresse ip qui est 192.168.20.254/24

```
routeur(config)#int f0/1.1
routeur(config-subif)#encapsulation dot1
routeur(config-subif)#encapsulation dot1Q 2
routeur(config-subif)#
*Mar 1 00:08:13.367: %LINK-3-UPDOWN: Interface FastEthernet0/1, changed state to up
routeur(config-subif)#ip add 192.168.20.254 255.255.255.0
routeur(config-subif)#no sh
routeur(config-subif)#[
```

Puis, on crée la sous interface f0/1.2 sur le routeur et on lui attribue le vlan 3 et une adresse ip qui est 192.168.30.254/24

```
routeur(config)#int f0/1.2
routeur(config-subif)#encapsulation dot1Q 3
routeur(config-subif)#ip add 192.168.30.254 255.255.255.0
routeur(config-subif)#no sh
routeur(config-subif)#exit
routeur(config)#[
```

Maintenant, nous allons enregistrer les configurations par la commande write memory

```
routeur#write memory
Building configuration...
[OK]
routeur#[
```

## • Configuration des VPCs et Test de connectivité

Nous allons donner les adresses aux vpcs en indiquant leur passerelle

→ Configuration de PC1

```
PC1> ip 192.168.20.1/24 192.168.20.254
Checking for duplicate address...
PC1 : 192.168.20.1 255.255.255.0 gateway 192.168.20.254

PC1> [
```

→ Test de connectivité entre PC1 et sa passerelle

```
PC1> ping 192.168.20.254

84 bytes from 192.168.20.254 icmp_seq=1 ttl=255 time=9.542 ms
84 bytes from 192.168.20.254 icmp_seq=2 ttl=255 time=4.421 ms
84 bytes from 192.168.20.254 icmp_seq=3 ttl=255 time=5.029 ms
84 bytes from 192.168.20.254 icmp_seq=4 ttl=255 time=6.783 ms
84 bytes from 192.168.20.254 icmp_seq=5 ttl=255 time=5.549 ms

PC1> [
```

→ Configuration de PC2

```
PC2> ip 192.168.30.1/24 192.168.30.254
Checking for duplicate address...
PC2 : 192.168.30.1 255.255.255.0 gateway 192.168.30.254

PC2> [
```

→ Test de connectivité entre PC2 et sa passerelle

```
PC2> ping 192.168.30.254

84 bytes from 192.168.30.254 icmp_seq=1 ttl=255 time=9.186 ms
84 bytes from 192.168.30.254 icmp_seq=2 ttl=255 time=6.554 ms
84 bytes from 192.168.30.254 icmp_seq=3 ttl=255 time=7.313 ms
84 bytes from 192.168.30.254 icmp_seq=4 ttl=255 time=6.381 ms
84 bytes from 192.168.30.254 icmp_seq=5 ttl=255 time=8.063 ms
```

PC2> █

→ Test de connectivité entre le pc1 et la passerelle du pc2

```
PC1> ping 192.168.30.254

84 bytes from 192.168.30.254 icmp_seq=1 ttl=255 time=9.356 ms
84 bytes from 192.168.30.254 icmp_seq=2 ttl=255 time=5.739 ms
84 bytes from 192.168.30.254 icmp_seq=3 ttl=255 time=5.308 ms
84 bytes from 192.168.30.254 icmp_seq=4 ttl=255 time=7.531 ms
84 bytes from 192.168.30.254 icmp_seq=5 ttl=255 time=5.378 ms
```

PC1> █

→ Test de connectivité entre PC2 et PC1

```
PC2> ping 192.168.20.1

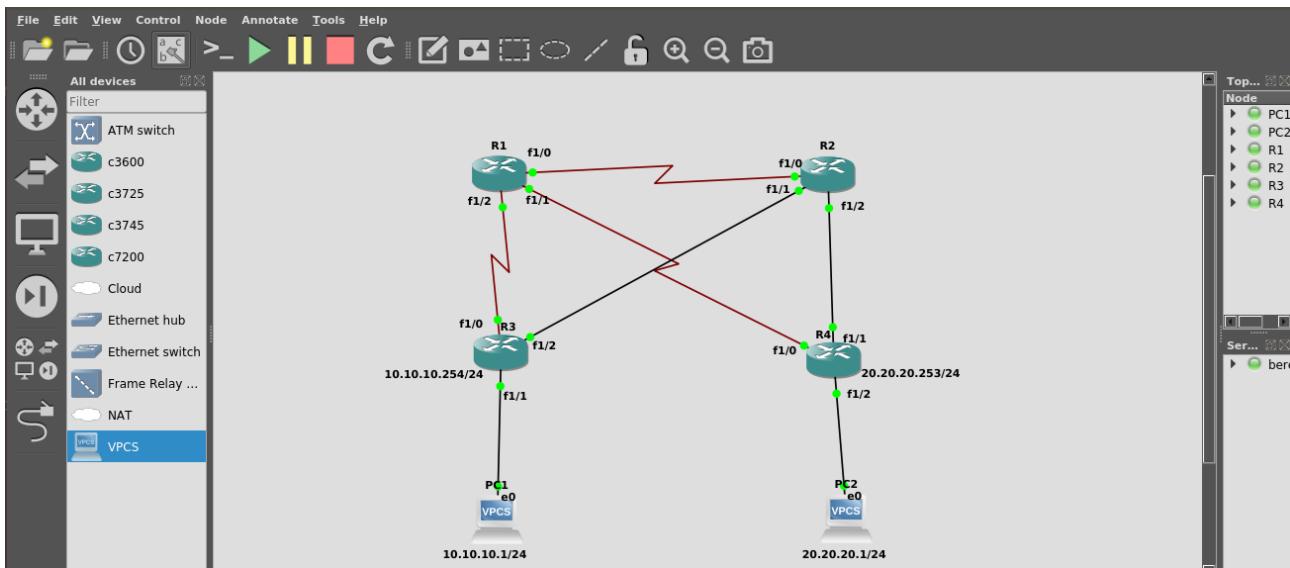
84 bytes from 192.168.20.1 icmp_seq=1 ttl=63 time=22.005 ms
84 bytes from 192.168.20.1 icmp_seq=2 ttl=63 time=16.215 ms
84 bytes from 192.168.20.1 icmp_seq=3 ttl=63 time=15.189 ms
84 bytes from 192.168.20.1 icmp_seq=4 ttl=63 time=16.729 ms
84 bytes from 192.168.20.1 icmp_seq=5 ttl=63 time=16.147 ms
```

PC2> █

ipInterfaces	Switch	Vlan	Sous interface	Adresses
pc1	e0 → f1/1	Vlan 2	f0/1.1	192.168.20.254/24
pc2	e0 → f1/2	Vlan 3	f0/1.2	192.168.30.254/24
Routeur	e0 → f0/1			
Sous interface f0/1.1				192.168.20.1/24
Sous interface 0/1.1				192.168.30.1/24 Tester la connectivité de pc3 et pc1

## 6. Routage Inter-vlan avec un switch de niveau 3

Ce TP consiste à faire appel à un équipement de niveau 3 pour faire communiquer les différents réseaux virtuels et chaque interface sera affecter à un vlan comme le montre l'architecture ci-dessous :



### Architecture de déploiement d'un routage Inter-vlan avec un switch de niveau 3

#### → Configuration de R1 :

Nous allons configurer les liens Trunk entre les switchs d'accès et les switchs de distribution

```
R1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#int range f1/0 -2
R1(config-if-range)#switchport mode trunk
R1(config-if-range)#
*Mar 1 00:02:06.179: %DTP-5-TRUNKPORTON: Port Fa1/0-2 has become dot1q trunk
R1(config-if-range)#
R1#
```

Ensuite, nous allons afficher les interfaces f1/0, f1/1, f1/2 comme suit :

```
R1#show interfaces f1/0 switchport
Name: Fa1/0
Switchport: Enabled
Administrative Mode: trunk
Operational Mode: trunk
Administrative Trunking Encapsulation: dot1q
Operational Trunking Encapsulation: dot1q
Negotiation of Trunking: Disabled
Access Mode VLAN: 0 ((Inactive))
Trunking Native Mode VLAN: 1 (default)
Trunking VLANs Enabled: ALL
Trunking VLANs Active: 1
Protected: false
Priority for untagged frames: 0
Override vlan tag priority: FALSE
Voice VLAN: none
Appliance trust: none
R1#
```

```
R1#show interfaces f1/1 switchport
Name: Fa1/1
Switchport: Enabled
Administrative Mode: trunk
Operational Mode: trunk
Administrative Trunking Encapsulation: dot1q
Operational Trunking Encapsulation: dot1q
Negotiation of Trunking: Disabled
Access Mode VLAN: 0 ((Inactive))
Trunking Native Mode VLAN: 1 (default)
Trunking VLANs Enabled: ALL
Trunking VLANs Active: 1
Protected: false
Priority for untagged frames: 0
Override vlan tag priority: FALSE
Voice VLAN: none
Appliance trust: none
R1#
```

```
R1#show interfaces f1/2 switchport
Name: Fa1/2
Switchport: Enabled
Administrative Mode: trunk
Operational Mode: trunk
Administrative Trunking Encapsulation: dot1q
Operational Trunking Encapsulation: dot1q
Negotiation of Trunking: Disabled
Access Mode VLAN: 0 ((Inactive))
Trunking Native Mode VLAN: 1 (default)
Trunking VLANs Enabled: ALL
Trunking VLANs Active: 1
Protected: false
Priority for untagged frames: 0
Override vlan tag priority: FALSE
Voice VLAN: none
Appliance trust: none
R1#
```

## →Configuration du R2

Nous allons configurer les liens Trunk entre les switchs d'accès et les switchs de distribution

```
R2#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R2(config)#int range f1/0 -2
R2(config-if-range)#switchport mode trunk
R2(config-if-range)#
*Mar 1 00:24:48.359: %DTP-5-TRUNKPORTON: Port Fa1/0-2 has become dot1q trunk
R2(config-if-range)#
R2#
```

Ensuite, nous allons afficher les interfaces f1/0, f1/1, f1/2 comme suit :

```
R2#show int f1/0 switchport
Name: Fa1/0
Switchport: Enabled
Administrative Mode: trunk
Operational Mode: trunk
Administrative Trunking Encapsulation: dot1q
Operational Trunking Encapsulation: dot1q
Negotiation of Trunking: Disabled
Access Mode VLAN: 0 ((Inactive))
Trunking Native Mode VLAN: 1 (default)
Trunking VLANs Enabled: ALL
Trunking VLANs Active: 1
Protected: false
Priority for untagged frames: 0
Override vlan tag priority: FALSE
Voice VLAN: none
Appliance trust: none
R2#
```

```

R2#show int f1/2 switchport
Name: Fa1/2
Switchport: Enabled
Administrative Mode: trunk
Operational Mode: trunk
Administrative Trunking Encapsulation: dot1q
Operational Trunking Encapsulation: dot1q
Negotiation of Trunking: Disabled
Access Mode VLAN: 0 ((Inactive))
Trunking Native Mode VLAN: 1 (default)
Trunking VLANs Enabled: ALL
Trunking VLANs Active: 1
Protected: false
Priority for untagged frames: 0
Override vlan tag priority: FALSE
Voice VLAN: none
Appliance trust: none
R2#show int f1/1 switchport
Name: Fa1/1
Switchport: Enabled
Administrative Mode: trunk
Operational Mode: trunk
Administrative Trunking Encapsulation: dot1q
Operational Trunking Encapsulation: dot1q
Negotiation of Trunking: Disabled
Access Mode VLAN: 0 ((Inactive))
Trunking Native Mode VLAN: 1 (default)
Trunking VLANs Enabled: ALL
Trunking VLANs Active: 1
Protected: false
Priority for untagged frames: 0
Override vlan tag priority: FALSE
Voice VLAN: none
Appliance trust: none
R2#

```

## → Configuration de R3

Nous allons mettre en liaison trunk les interfaces f1/0 et f1/2

```

R3#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R3(config)#int f1/0
R3(config-if)#switchport mode trunk
R3(config-if)#
*Mar 1 00:29:42.267: %DTP-5-TRUNKPORTON: Port Fa1/0 has become dot1q trunk
R3(config-if)#exit
R3(config)#int f1/2
R3(config-if)#switchport mode trunk
R3(config-if)#
*Mar 1 00:30:04.395: %DTP-5-TRUNKPORTON: Port Fa1/2 has become dot1q trunk
R3(config-if)#

```

Ensuite, nous allons afficher ces interfaces pour voir

```

R3#show interfaces f1/0 switchport
Name: Fa1/0
Switchport: Enabled
Administrative Mode: trunk
Operational Mode: trunk
Administrative Trunking Encapsulation: dot1q
Operational Trunking Encapsulation: dot1q
Negotiation of Trunking: Disabled
Access Mode VLAN: 0 ((Inactive))
Trunking Native Mode VLAN: 1 (default)
Trunking VLANs Enabled: ALL
Trunking VLANs Active: 1
Protected: false
Priority for untagged frames: 0
Override vlan tag priority: FALSE
Voice VLAN: none
Appliance trust: none
R3#
R3#show interfaces f1/2 switchport
Name: Fa1/2
Switchport: Enabled
Administrative Mode: trunk
Operational Mode: trunk
Administrative Trunking Encapsulation: dot1q
Operational Trunking Encapsulation: dot1q
Negotiation of Trunking: Disabled
Access Mode VLAN: 0 ((Inactive))
Trunking Native Mode VLAN: 1 (default)
Trunking VLANs Enabled: ALL
Trunking VLANs Active: 1
Protected: false
Priority for untagged frames: 0
Override vlan tag priority: FALSE
Voice VLAN: none
Appliance trust: none
R3#

```

## → Configuration de R4

Nous allons tout d'abord mettre en liaison trunk les interfaces f1/0 et f1/1

```
R4#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R4(config)#int range f1/0 -1
R4(config-if-range)#switchport mode trunk
R4(config-if-range)#
*Mar 1 00:33:19.403: %DTP-5-TRUNKPORTON: Port Fa1/0-1 has become dot1q trunk
R4(config-if-range)#[
```

Ensuite, nous allons visualiser les configurations faites :

```
R4#show interfaces f1/0 switchport
Name: Fa1/0
Switchport: Enabled
Administrative Mode: trunk
Operational Mode: trunk
Administrative Trunking Encapsulation: dot1q
Operational Trunking Encapsulation: dot1q
Negotiation of Trunking: Disabled
Access Mode VLAN: 0 ((Inactive))
Trunking Native Mode VLAN: 1 (default)
Trunking VLANs Enabled: ALL
Trunking VLANs Active: 1
Protected: false
Priority for untagged frames: 0
Override vlan tag priority: FALSE
Voice VLAN: none
Appliance trust: none
R4#show interfaces f1/1 switchport
Name: Fa1/1
Switchport: Enabled
Administrative Mode: trunk
Operational Mode: trunk
Administrative Trunking Encapsulation: dot1q
Operational Trunking Encapsulation: dot1q
Negotiation of Trunking: Disabled
Access Mode VLAN: 0 ((Inactive))
Trunking Native Mode VLAN: 1 (default)
Trunking VLANs Enabled: ALL
Trunking VLANs Active: 1
Protected: false
Priority for untagged frames: 0
Override vlan tag priority: FALSE
Voice VLAN: none
Appliance trust: none
R4#[
```

## →Création des vlans

Sur R1, R2, R3 et R4 nous allons créer les vlans 10 et 20 comme suit :

```
R1#vlan database
% Warning: It is recommended to configure VLAN from config mode,
as VLAN database mode is being deprecated. Please consult user
documentation for configuring VTP/VLAN in config mode.

R1(vlan)#vlan 10
VLAN 10 added:
  Name: VLAN0010
R1(vlan)#vlan 20
VLAN 20 added:
  Name: VLAN0020
R1(vlan)#apply
APPLY completed.
R1(vlan)#[
```

```
R2#vlan database
% Warning: It is recommended to configure VLAN from config mode,
as VLAN database mode is being deprecated. Please consult user
documentation for configuring VTP/VLAN in config mode.

R2(vlan)#vlan 10
VLAN 10 added:
  Name: VLAN0010
R2(vlan)#vlan 20
VLAN 20 added:
  Name: VLAN0020
R2(vlan)#apply
APPLY completed.
R2(vlan)#[
```

```
R3#vlan database
% Warning: It is recommended to configure VLAN from config mode,
as VLAN database mode is being deprecated. Please consult user
documentation for configuring VTP/VLAN in config mode.

R3(vlan)#vlan 10
VLAN 10 added:
  Name: VLAN0010
R3(vlan)#vlan 20
VLAN 20 added:
  Name: VLAN0020
R3(vlan)#apply
APPLY completed.
R3(vlan)#[
```

```
R4(vlan)#vlan 10
VLAN 10 added:
  Name: VLAN0010
R4(vlan)#vlan 20
VLAN 20 added:
  Name: VLAN0020
R4(vlan)#apply
APPLY completed.
R4(vlan)#[
```

## →Affectation des interfaces de switchs d'accès aux vlans

- On attribue l'interface f1/1 de R3 au vlan 10

```
R3#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R3(config)#int f1/1
R3(config-if)#switchport mode access
R3(config-if)#switchport access vlan 10
R3(config-if)#[
```

- On attribue l'interface f1/2 de R4 au vlan 20

```
R4(config)#int f1/2
R4(config-if)#switchport mode access
R4(config-if)#switchport access vlan 20
R4(config-if)#[
```

## →Création des interfaces virtuelles sur les deux switchs de distribution (R1 et R2)

- Configuration du R1

```
R1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#int vlan 20
R1(config-if)#
*Mar 1 00:46:57.551: %LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan20, changed state to up
R1(config-if)#ip add 20.20.20.254 255.255.255.0
R1(config-if)#no sh
R1(config-if)#exit
R1(config)#
R1(config)#int vlan 10
R1(config-if)#exit
*Mar 1 00:47:33.811: %LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan10, changed state to up
R1(config-if)#ip add 10.10.10.254 255.255.255.0
R1(config-if)#no sh
R1(config-if)#exit
R1(config)#[
```

- Configuration du R2

```
R2#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R2(config)#int vlan 20
R2(config-if)#
*Mar 1 00:49:04.399: %LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan20, changed state to up
R2(config-if)#ip add 20.20.20.253 255.255.255.0
R2(config-if)#no sh
R2(config-if)#exit
R2(config)#
R2(config)#int vlan 10
R2(config-if)#
*Mar 1 00:49:35.359: %LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan10, changed state to up
R2(config-if)#ip add 10.10.10.0 255.255.255.0
Bad mask /24 for address 10.10.10.0
R2(config-if)#exit
R2(config)#ip routing
R2(config)#[
```

## →Configuration des machines et test de bon fonctionnement

Nous donnerons une adresse à chaque machine :

- PC1

```
PC1> ip 10.10.10.1/24 10.10.10.254
Checking for duplicate address...
PC1 : 10.10.10.1 255.255.255.0 gateway 10.10.10.254
PC1> █
```

- PC2

```
PC2> ip 20.20.20.1/24 20.20.20.253
Checking for duplicate address...
PC2 : 20.20.20.1 255.255.255.0 gateway 20.20.20.253
PC2> █
```

Nous allons depuis PC-1 faire un ping sur PC-2 comme le montre la figure ci-après :

```
PC2> ping 10.10.10.1
*20.20.20.253 icmp_seq=1 ttl=255 time=19.094 ms (ICMP type:3, code:1, Destination host unreachable)
*20.20.20.253 icmp_seq=2 ttl=255 time=8.282 ms (ICMP type:3, code:1, Destination host unreachable)
*20.20.20.253 icmp_seq=3 ttl=255 time=8.921 ms (ICMP type:3, code:1, Destination host unreachable)
*20.20.20.253 icmp_seq=4 ttl=255 time=6.389 ms (ICMP type:3, code:1, Destination host unreachable)
*20.20.20.253 icmp_seq=5 ttl=255 time=16.459 ms (ICMP type:3, code:1, Destination host unreachable)
PC2> █
```

```
PC1> ping 20.20.20.1
20.20.20.1 icmp_seq=1 timeout
84 bytes from 20.20.20.1 icmp_seq=2 ttl=63 time=10.956 ms
20.20.20.1 icmp_seq=3 timeout
20.20.20.1 icmp_seq=4 timeout
20.20.20.1 icmp_seq=5 timeout
PC1> █
```

On voit bien qu'elles communiquent !!

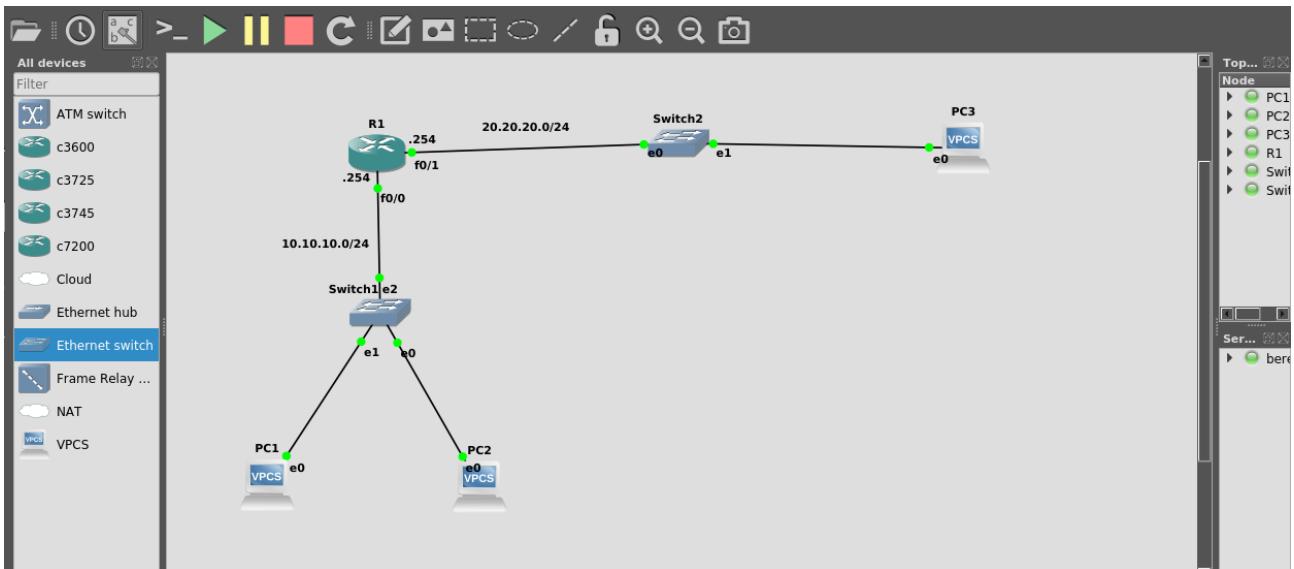
## TP3 : Mise en place d'ACL

### 1. NAT statique

En réseau informatique, on dit qu'un routeur fait du network address translation (NAT) lorsqu'il fait correspondre des adresses IP à d'autres adresses IP. En particulier, un cas courant est de permettre à des machines disposant d'adresses privées qui font partie d'un intranet et ne sont ni uniques ni routables à l'échelle d'Internet, de communiquer avec le reste d'Internet en utilisant vers l'extérieur des adresses externes publiques, uniques et routables. Ainsi, il est possible de faire correspondre une seule adresse externe publique visible sur Internet à toutes les adresses d'un réseau privé, afin de pallier l'épuisement des adresses IPv4. La fonction NAT dans un routeur de service intégré (ISR) traduit une adresse IP source interne en adresse IP globale. Il est également utilisé de façon similaire dans des réseaux privés virtuels.

### Objectif :

Ce TP consiste à placer un routeur à qui ses interfaces dont l'une a un réseau public et l'autre un réseau privé. Nous allons faire de telle sorte que les machines qui se trouvent dans chaque réseau communiquent grâce à la nat statique. Voici notre architecture de déploiement :



## Architecture de déploiement d'une nat statique

Nous allons passer à la configuration des interfaces du routeur :

- L'interface f1/0 a pour adresse 10.10.10.254/24 qui est l'adresse privée

```
R1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#int f0/0
R1(config-if)#ip add 10.10.10.254 255.255.255.0
R1(config-if)#no sh
R1(config-if)#ip n
*Mar 1 00:01:24.879: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state to up
*Mar 1 00:01:25.879: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
R1(config-if)#ip nat inside
R1(config-if)#
*Mar 1 00:01:37.739: %LINEPROTO-5-UPDOWN: Line protocol on Interface NVI0, changed state to up
R1(config-if)#[
```

- L'interface f1/1 a pour adresse 20.20.20.254/24 qui est une adresse publique du routeur

```
R1(config)#int f0/1
R1(config-if)#ip add 20.20.20.254 255.255.255.0
R1(config-if)#no sh
R1(config-if)#
*Mar 1 00:04:01.091: %LINK-3-UPDOWN: Interface FastEthernet0/1, changed state to up
*Mar 1 00:04:02.091: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1, changed state to up
R1(config-if)#ip nat outside
R1(config-if)#[
```

Ensuite, nous allons passer à la création du mappage entre adresse privée et publique

```
R1(config)#ip nat inside source static 10.10.10.1 20.20.20.60
R1(config)#ip nat inside source static 10.10.10.2 20.20.20.70
R1(config)#[
```

Nous allons par la suite afficher la correspondance des adresses par la commande suivante :

```
R1#show ip nat translations
Pro Inside global      Inside local      Outside local      Outside global
--- 20.20.20.60          10.10.10.1           ---             ---
--- 20.20.20.70          10.10.10.2           ---             ---
R1#[
```

- On fixe l'adresse ip aux pc

```
PC1> ip 10.10.10.1/24 10.10.10.254
Checking for duplicate address...
PC1 : 10.10.10.1 255.255.255.0 gateway 10.10.10.254
```

```
PC2> ip 10.10.10.2/24 10.10.10.254
Checking for duplicate address...
PC2 : 10.10.10.2 255.255.255.0 gateway 10.10.10.254
```

PC2> █

```
PC3> ip 20.20.20.1/24 20.20.20.254
Checking for duplicate address...
PC3 : 20.20.20.1 255.255.255.0 gateway 20.20.20.254
```

PC3> █

Cependant, nous allons tester la communication des machines :

Pc1 va faire un ping à pc3 et nous allons superviser les paquets via wireshark

```
PC1> ping 20.20.20.1
```

```
20.20.20.1 icmp_seq=1 timeout
84 bytes from 20.20.20.1 icmp_seq=2 ttl=63 time=21.378 ms
84 bytes from 20.20.20.1 icmp_seq=3 ttl=63 time=25.775 ms
84 bytes from 20.20.20.1 icmp_seq=4 ttl=63 time=16.149 ms
84 bytes from 20.20.20.1 icmp_seq=5 ttl=63 time=16.284 ms
```

PC1> █

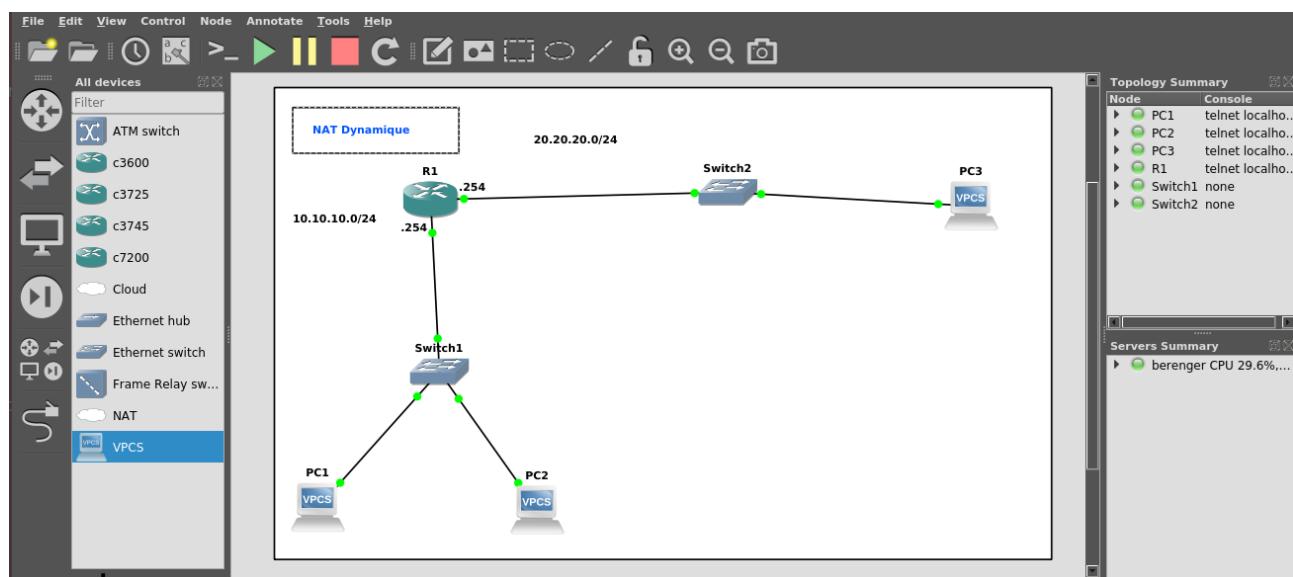
On voit bien que Pc1 du réseau privé parvient à communiquer avec PC3 du réseau public

## 2. NAT dynamique

La NAT dynamique fonctionne sur le même principe que la NAT statique mis à part que l'adresse publique qui sera attribuée à la station A (station en adressage privé) ne sera pas toujours la même. Cette adresse sera choisie dans un "pool d'adresses" à la disposition du routeur et donnée pour un temps défini. L'adresse obtenue par A sera potentiellement à chaque fois différente.

### Objectif :

Ce TP consiste à placer un routeur à deux interfaces dont l'une est un réseau public et l'autre un réseau privé. Nous allons faire de telle sorte que les machines qui se trouvent dans chaque réseau communiquent grâce à la translation d'adresse. Voici notre architecture de déploiement :



# Architecture de déploiement d'une nat dynamique

Nous allons passer à la configuration des interfaces du routeur :

- L'interface f1/0 a pour adresse 10.10.10.254/24 qui est l'adresse privée

```
R1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#int f0/0
R1(config-if)#ip add 10.10.10.254 255.255.255.0
R1(config-if)#no sh
R1(config-if)#ip n
*Mar 1 00:03:18.351: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state to up
*Mar 1 00:03:19.351: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
R1(config-if)#ip nat inside

*Mar 1 00:03:25.427: %LINEPROTO-5-UPDOWN: Line protocol on Interface NVI0, changed state to up
R1(config-if)#[
```

- L'interface f1/1 a pour adresse 20.20.20.254/24 qui est une adresse publique du routeur

```
R1(config-if)#exit
R1(config)#int f0/1
R1(config-if)#ip add 20.20.20.254 255.255.255.0
R1(config-if)#no sh
R1(config-if)#int f0/1
*Mar 1 00:04:41.619: %LINK-3-UPDOWN: Interface FastEthernet0/1, changed state to up
*Mar 1 00:04:42.619: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1, changed state to up
R1(config-if)#ip nat outside
R1(config-if)#[
```

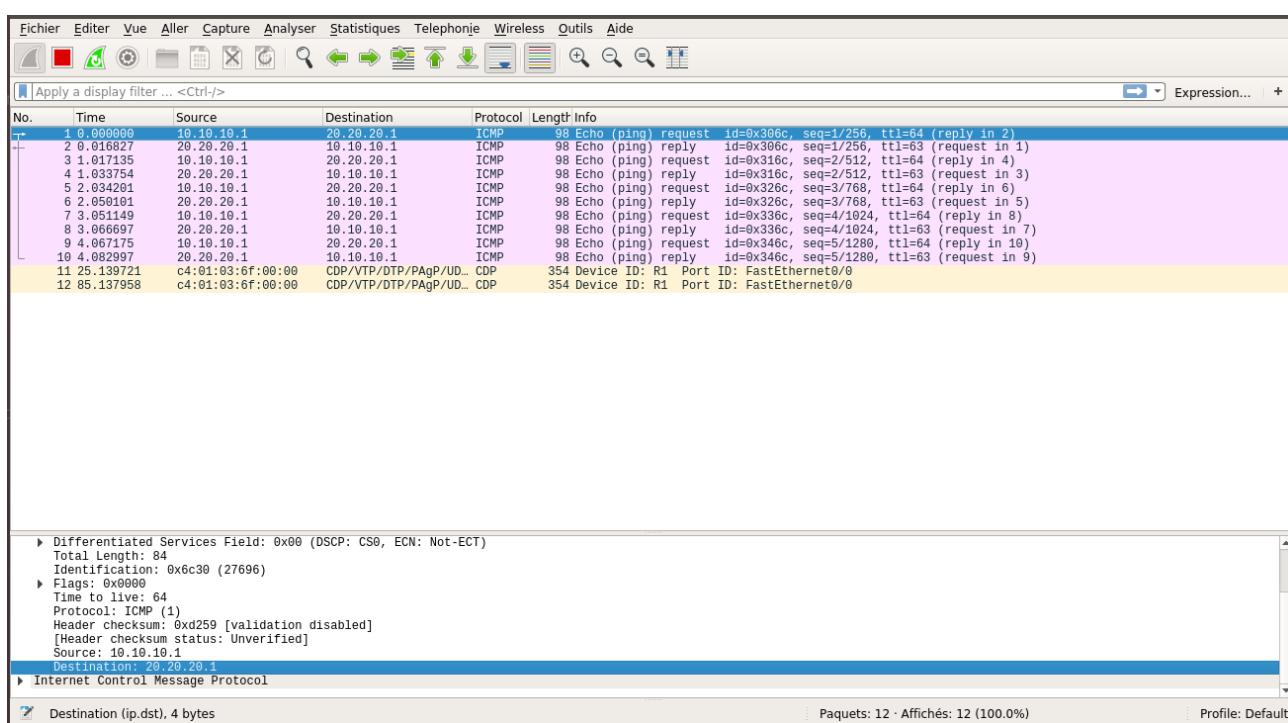
Ensuite, nous allons définir le pool d'adresses IP publiques 192.168.42.0/24 dans le réseau public par la commande suivante.

```
R1(config)#ip nat pool telecoms 20.20.20.30 20.20.20.120 netmask 255.255.255.0
R1(config)#[
```

Puis, on définit une liste d'accès standard autorisant les adresses qui doivent être traduites

```
R1(config)#access-list 1 permit 10.10.10.0 0.0.0.255
R1(config)#[
```

Nous allons tester la connectivité des machines et visualiser les paquets sur wireshark



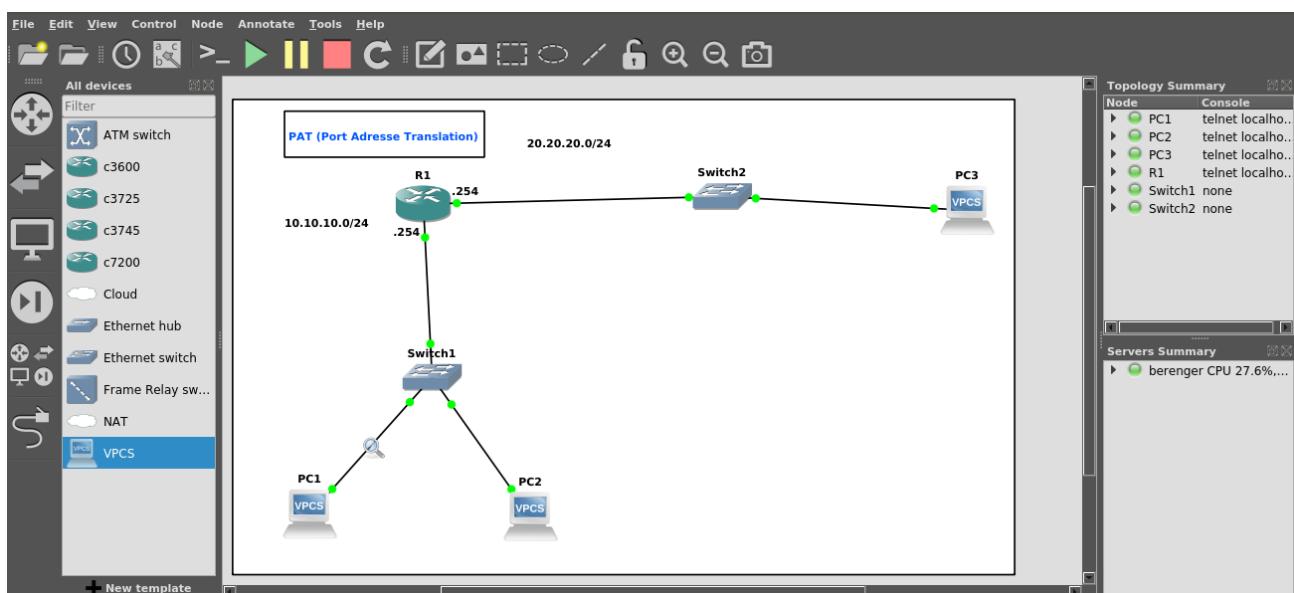
### 3. PAT

Ce que l'on appelle communément NAT dynamique est en fait la mise en place d'une autre translation : la translation de port ou PAT (Port Adresse Translation).

Objectif de ce tp permet de mettre en place le protocole **VRRP** c'est ce qui nous a permis que notre machine webterm ait deux passerelles c est pour cela on a coupé le lien vers R1 le Master et le webterm continue d 'avoir la connexion en passant par l'autre lien. Donc le routeur2 est considéré comme le master protocole s'appuyant sur UDP est lui aussi implémenté sur les routeurs en complément de la translation d'adresse.

#### Objectif

Ce TP consiste à place un routeur à qui ses interfaces dont l'une a un réseau public et l'autre un réseau privé. Nous allons faire de telle sorte que les machines qui se trouvent dans chaque réseau communiquent grâce à la translation d'adresse par port. Voici notre architecture de déploiement :



#### Architecture de déploiement d'une PAT

Nous allons passer à la configuration des interfaces du routeur :

- L'interface f1/0 a pour adresse 10.10.10.254/24 qui est l'adresse privée Notre Objectif est atteint on parvient à mettre en place le protocole VRRP c'est ce qui nous a permis que notre machine webterm ait deux passerelles c est pour cela on a coupé le lien vers R1,le Master et le webterm continue d 'avoir la connexion en passant par l'autre lien. Donc le routeur2 est considéré comme le master.

```
R1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#int f0/0
R1(config-if)#ip add 10.10.10.254 255.255.255.0
R1(config-if)#no sh
R1(config-if)#ip n
*Mar 1 00:03:34.243: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state to up
*Mar 1 00:03:35.243: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
R1(config-if)#ip nat inside
R1(config-if)#
*Mar 1 00:03:42.031: %LINEPROTO-5-UPDOWN: Line protocol on Interface NVI0, changed state to up
R1(config-if)#[
```

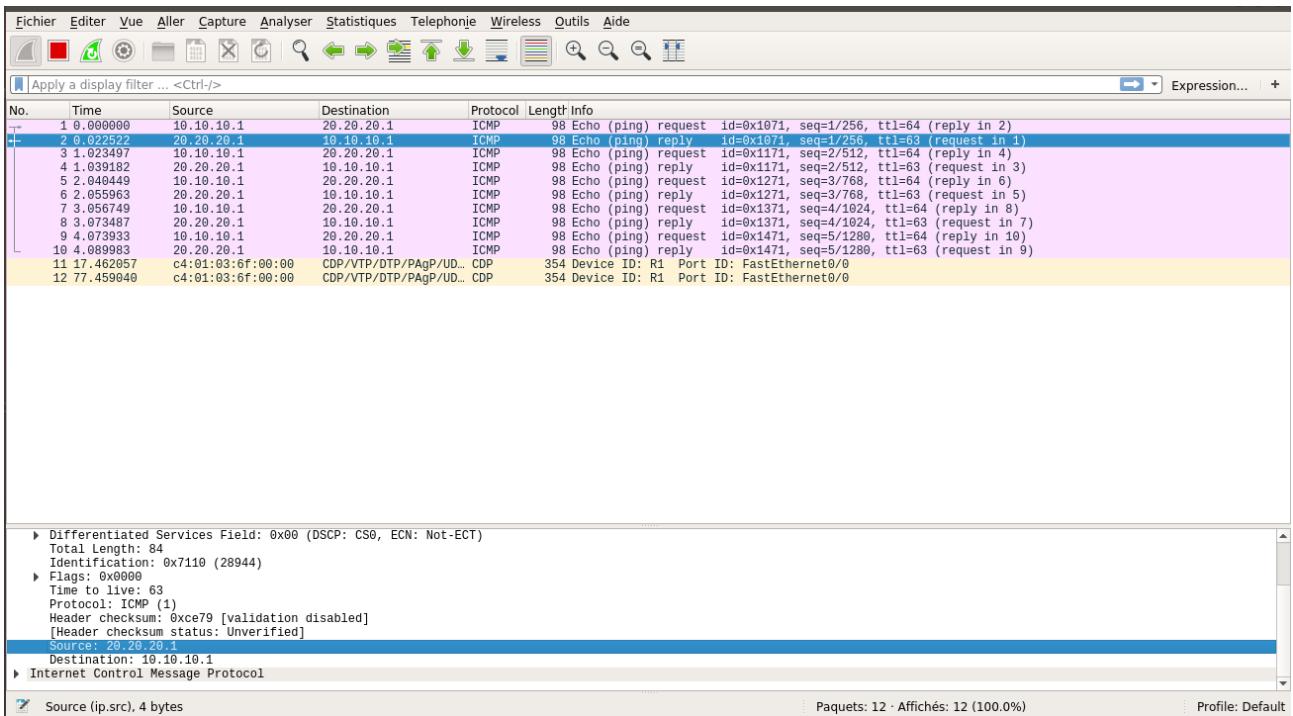
- L'interface f1/1 a pour adresse 20.20.20.254/24 qui est une adresse publique du routeur

```
R1(config)#  
R1(config)#int f0/1  
R1(config-if)#ip add 20.20.20.254 255.255.255.0  
R1(config-if)#no sh  
R1(config-if)#1  
*Mar 1 00:06:22.031: %LINK-3-UPDOWN: Interface FastEthernet0/1, changed state to up  
*Mar 1 00:06:23.031: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1, changed state to up  
R1(config-if)#ip nat outside  
R1(config-if)#
```

Ensuite, configurer l'interface f0/1 comme interface de PAT et mettre en place des règles access-list

```
R1(config)#ip nat inside source list 1 interface f0/1 overload  
R1(config)#access-list 1 permit 10.10.10.0 0.0.0.255  
R1(config)#
```

Maintenant, nous allons tester la connectivité entre les deux réseaux et visualiser sur wireshark :



On voit bien que pc1 communique avec pc3

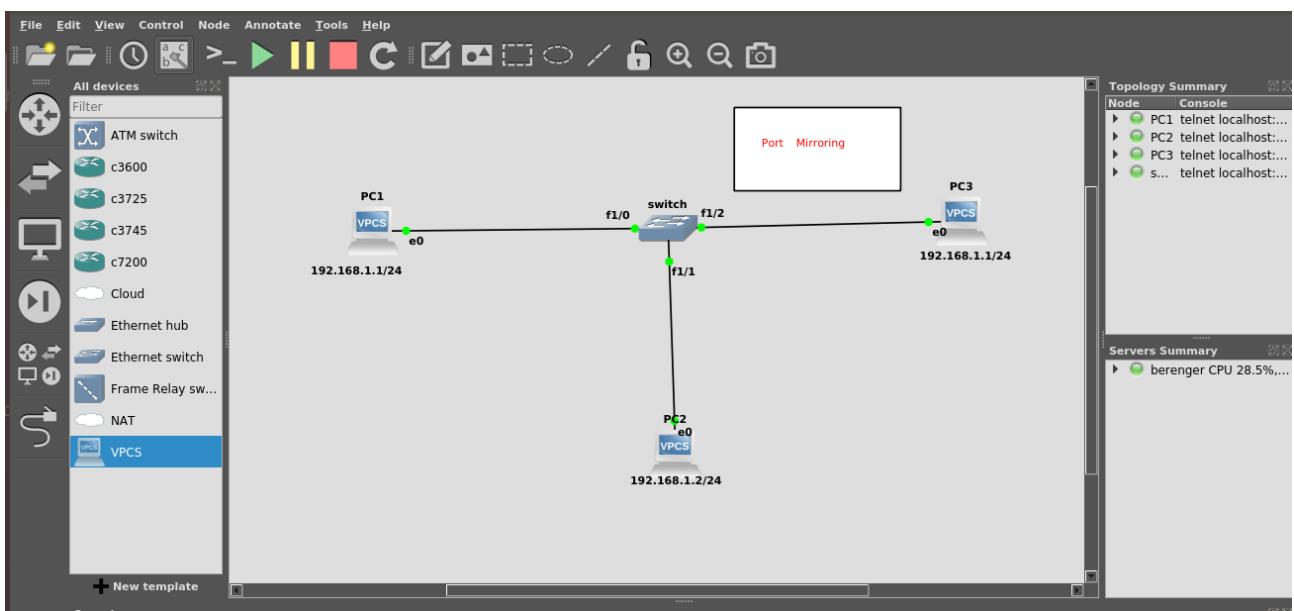
## TP4 : Mise en œuvre des ports mirroring et forwarding

### 1. Port mirroring

La mise en miroir des ports est utilisée sur un commutateur réseau pour envoyer une copie des paquets réseau vus sur un port de commutateur (ou un VLAN entier) vers une connexion de surveillance réseau sur un autre port de commutateur. Ceci est couramment utilisé pour les appliances réseau qui nécessitent une surveillance du trafic réseau, comme un système de détection d'intrusion, une sonde passive ou une technologie de surveillance des utilisateurs réels (RUM) utilisée pour prendre en charge la gestion des performances des applications (APM). La mise en miroir de ports sur un commutateur Cisco Systems est généralement appelée Switched Port Analyzer (SPAN) ou analyseur de ports commutés à distance (RSPAN). D'autres fournisseurs portent des noms différents, comme Roving Analysis Port (RAP) sur les commutateurs 3Com. Les ingénieurs réseau ou les administrateurs utilisent la mise en miroir des ports pour analyser et déboguer les données ou diagnostiquer les erreurs sur un réseau. Il aide les administrateurs à surveiller de près les performances du réseau et les alerte en cas de problème. Il peut être utilisé pour mettre en miroir le trafic entrant ou sortant (ou les deux) sur une ou plusieurs interfaces.

### Objectif

Ce TP consiste à mettre en place un switch manageable, reliant trois machines, dont l'une des interfaces représente le port mirroring comme le montre la figure ci-dessous :



### Architecture de déploiement d'un port mirroring

#### ✓ Configuration du switch

Pour mettre œuvre le port mirroring, nous allons copier les trames des ports f1/0 et f1/1 session 1 comme le montre la figure ci-dessous :

```
switch#conf t
Enter configuration commands, one per line. End with CNTL/Z.
switch(config)#monitor session 1 source interface f1/0
switch(config)#monitor session 1 source interface f1/1
switch(config)#[
```

Ensuite, nous allons copier la session sur le port f1/2 qui sera notre port mirroring

```
switch(config)#monitor session 1 destination interface f1/2  
switch(config)#[ ]
```

Maintenant, nous allons visualiser les ports sources et le port destination comme suit :

```
switch#show monitor session 1  
Session 1  
-----  
Source Ports:  
    RX Only:      None  
    TX Only:      None  
    Both:         Fa1/0-1  
Source VLANs:  
    RX Only:      None  
    TX Only:      None  
    Both:         None  
Destination Ports: Fa1/2  
Filter VLANs:     None  
  
switch#[ ]
```

## ✓ Configuration de machines

Ci-dessous nous avons la configuration des machines PC1, PC2, PC3 :

```
PC1> ip 192.168.1.1/24  
Checking for duplicate address...  
PC1 : 192.168.1.1 255.255.255.0  
  
PC1> [ ]
```

```
PC2> ip 192.168.1.2/24  
Checking for duplicate address...  
PC2 : 192.168.1.2 255.255.255.0  
  
PC2> [ ]
```

```
PC3> ip 192.168.1.3/24  
Checking for duplicate address...  
PC3 : 192.168.1.3 255.255.255.0  
  
PC3> [ ]
```

Cependant, PC2 va faire un ping à PC1 et nous allons visualiser les trames sur wireshark sur le câble qui relie f1/2 et le PC3

Fichier	Éditer	Vue	Aller	Capture	Analyser	Statistiques	Telephoneie	Wireless	Outils	Aide	
No.	Time	Source	Destination	Protocol	Length	Info					
63	94.006175	c4:01:0a:d7:f1:01	Spanning-tree-(for-> STP	60	Conf. Root = 32768/0/c4:01:0a:d7:00:00 Cost = 0 Port = 0x802a						
64	96.000165	c4:01:0a:d7:f1:01	Spanning-tree-(for-> STP	60	Conf. Root = 32768/0/c4:01:0a:d7:00:00 Cost = 0 Port = 0x802a						
65	98.005174	c4:01:0a:d7:f1:01	Spanning-tree-(for-> STP	60	Conf. Root = 32768/9/c4:01:0a:d7:00:09 Cost = 0 Port = 0x802a						
66	99.998965	c4:01:0a:d7:f1:01	Spanning-tree-(for-> STP	60	Conf. Root = 32768/9/c4:01:0a:d7:00:09 Cost = 0 Port = 0x802a						
67	102.002966	c4:01:0a:d7:f1:01	Spanning-tree-(for-> STP	60	Conf. Root = 32768/9/c4:01:0a:d7:00:09 Cost = 0 Port = 0x802a						
68	104.007268	c4:01:0a:d7:f1:01	Spanning-tree-(for-> STP	60	Conf. Root = 32768/9/c4:01:0a:d7:00:09 Cost = 0 Port = 0x802a						
69	106.001698	c4:01:0a:d7:f1:01	Spanning-tree-(for-> STP	60	Conf. Root = 32768/9/c4:01:0a:d7:00:09 Cost = 0 Port = 0x802a						
70	108.008136	c4:01:0a:d7:f1:01	Spanning-tree-(for-> STP	60	Conf. Root = 32768/9/c4:01:0a:d7:00:09 Cost = 0 Port = 0x802a						
71	110.000765	c4:01:0a:d7:f1:01	Spanning-tree-(for-> STP	60	Conf. Root = 32768/9/c4:01:0a:d7:00:09 Cost = 0 Port = 0x802a						
72	112.003539	c4:01:0a:d7:f1:01	Spanning-tree-(for-> STP	60	Conf. Root = 32768/9/c4:01:0a:d7:00:09 Cost = 0 Port = 0x802a						
73	114.007522	c4:01:0a:d7:f1:01	Spanning-tree-(for-> STP	60	Conf. Root = 32768/9/c4:01:0a:d7:00:09 Cost = 0 Port = 0x802a						
74	116.000927	c4:01:0a:d7:f1:01	Spanning-tree-(for-> STP	60	Conf. Root = 32768/9/c4:01:0a:d7:00:09 Cost = 0 Port = 0x802a						
75	118.004684	c4:01:0a:d7:f1:01	Spanning-tree-(for-> STP	60	Conf. Root = 32768/9/c4:01:0a:d7:00:09 Cost = 0 Port = 0x802a						
76	120.008412	c4:01:0a:d7:f1:01	Spanning-tree-(for-> STP	60	Conf. Root = 32768/9/c4:01:0a:d7:00:09 Cost = 0 Port = 0x802a						
77	122.001598	c4:01:0a:d7:f1:01	Spanning-tree-(for-> STP	60	Conf. Root = 32768/9/c4:01:0a:d7:00:09 Cost = 0 Port = 0x802a						
78	124.004264	c4:01:0a:d7:f1:01	Spanning-tree-(for-> STP	60	Conf. Root = 32768/9/c4:01:0a:d7:00:09 Cost = 0 Port = 0x802a						
79	126.007478	c4:01:0a:d7:f1:01	Spanning-tree-(for-> STP	60	Conf. Root = 32768/9/c4:01:0a:d7:00:09 Cost = 0 Port = 0x802a						
80	128.001175	c4:01:0a:d7:f1:01	Spanning-tree-(for-> STP	60	Conf. Root = 32768/9/c4:01:0a:d7:00:09 Cost = 0 Port = 0x802a						
81	130.004697	c4:01:0a:d7:f1:01	Spanning-tree-(for-> STP	60	Conf. Root = 32768/9/c4:01:0a:d7:00:09 Cost = 0 Port = 0x802a						
82	131.152550	c4:01:0a:d7:f1:01	CDP/VTP/DTP/PagP/UD.. CDP	349	Device ID: switch Port ID: FastEthernet1/1						
83	131.999943	c4:01:0a:d7:f1:01	Spanning-tree-(for-> STP	60	Conf. Root = 32768/9/c4:01:0a:d7:00:09 Cost = 0 Port = 0x802a						
84	134.002561	c4:01:0a:d7:f1:01	Spanning-tree-(for-> STP	60	Conf. Root = 32768/9/c4:01:0a:d7:00:09 Cost = 0 Port = 0x802a						
85	136.005534	c4:01:0a:d7:f1:01	Spanning-tree-(for-> STP	60	Conf. Root = 32768/9/c4:01:0a:d7:00:09 Cost = 0 Port = 0x802a						
86	137.999982	c4:01:0a:d7:f1:01	Spanning-tree-(for-> STP	60	Conf. Root = 32768/9/c4:01:0a:d7:00:09 Cost = 0 Port = 0x802a						
87	140.001928	c4:01:0a:d7:f1:01	Spanning-tree-(for-> STP	60	Conf. Root = 32768/9/c4:01:0a:d7:00:09 Cost = 0 Port = 0x802a						
88	142.004602	c4:01:0a:d7:f1:01	Spanning-tree-(for-> STP	60	Conf. Root = 32768/9/c4:01:0a:d7:00:09 Cost = 0 Port = 0x802a						
89	144.007514	c4:01:0a:d7:f1:01	Spanning-tree-(for-> STP	60	Conf. Root = 32768/0/c4:01:0a:d7:00:00 Cost = 0 Port = 0x802a						
90	146.000870	c4:01:0a:d7:f1:01	Spanning-tree-(for-> STP	60	Conf. Root = 32768/0/c4:01:0a:d7:00:00 Cost = 0 Port = 0x802a						
91	148.003858	c4:01:0a:d7:f1:01	Spanning-tree-(for-> STP	60	Conf. Root = 32768/0/c4:01:0a:d7:00:00 Cost = 0 Port = 0x802a						

► Frame 38: 9 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface 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)

► Destination: Private\_66:68:01 (00:50:79:66:68:01)

► Source: Private\_66:68:00 (00:50:79:66:68:00)

Type: IPv4 (0x0800)

► Internet Protocol Version 4, Src: 192.168.1.1, Dst: 192.168.1.2

► Internet Control Message Protocol