



Sommaire

TP1 - Adresse IP et masque de sous-réseau.....	2
TP2 - Classe IP en Python.....	3
TP3 - Adresse IP et notation CIDR.....	4
TP4 - Classe IP et notation CIDR en Python.....	5
TP5 - Nombre d'hôtes du réseau.....	6
TP6 - Nom de domaine.....	7
TP7 - Exemple de réseau.....	8
TP8 - Simuler un réseau avec Filius : étape1.....	9
TP9 - Simuler un réseau avec Filius : étape2.....	10
TP10 - Ajouter un serveur DNS et un serveur web avec Filius : étape3.....	11
TP11 - Étude d'un réseau avec Filius.....	12
TP12 - Table de routage simplifiée avec le protocole RIP.....	13
TP13 - Table de routage détaillée avec le protocole RIP.....	14
TP14 - Table de routage avec le protocole OSPF.....	15
TP15 - Algorithme de Dijkstra.....	16

TP1 - Adresse IP et masque de sous-réseau

1. On considère l'adresse IP : **198.165.145.58** avec le masque de sous réseau **255.255.0.0** .

- Écrire l'adresse IP ci-dessus en binaire :

- Écrire le masque ci-dessus en binaire :

- En déduire l'adresse du réseau en binaire :

- En déduire l'adresse du réseau en écriture décimale :

2. On considère l'adresse IP : **198.165.145.58** avec le masque de sous réseau **255.255.254.0** .

- Écrire l'adresse IP ci-dessus en binaire :

- Écrire le masque ci-dessus en binaire :

- En déduire l'adresse du réseau en binaire :

- En déduire l'adresse du réseau en écriture décimale :



Lever la main pour valider ce TP.

TP2 - Classe IP en Python

- On considère l'adresse IP : **192.168.148.17** et le masque de sous-réseau : **255.255.248.0** . Taper dans la console de Thonny les opérations logiques : 192 & 255 pour chacun des 4 octets afin d'obtenir l'adresse du réseau :

- Sur votre compte, dans le dossier **NSI** , créer un sous-dossier **TP-Chapitre8** dans lequel on rangera les TP de ce chapitre.
- Récupérer le fichier **ip.py** donné en ressource et l'enregistrer dans le dossier **TP-Chapitre8** sur votre compte.
- Compléter le code de la fonction **adresse_reseau()** ci-dessous qui renvoie l'adresse IP du réseau.

```
def adresse_reseau(ip1, masque):  
    chaine =   
    chaine += str(ip1.liste[0] & masque.liste[0])  
    for i in range(1, 4):  
        chaine +=   
    ad_reseau = IP(chaine)  
    return ad_reseau
```

- Compléter le code de la fonction **adresse_reseau()** dans le fichier **ip.py** . Puis vérifier que cette fonction s'exécute correctement. Exemple de retour attendu dans la console :

```
>>> ip1 = IP("192.168.137.11")  
>>> masque = IP("255.255.240.0")  
>>> print(adresse_reseau(ip1, masque))  
  
192.168.128.0
```

- On considère l'adresse IP : **192.208.127.19** et le masque de sous-réseau : **255.255.224.0** . Utiliser la fonction **adresse_reseau()** pour déterminer l'adresse du réseau :

- On considère l'adresse IP : **178.105.186.27** et le masque de sous-réseau : **255.255.128.0** . Utiliser la fonction **adresse_reseau()** pour déterminer l'adresse du réseau :



Lever la main pour valider ce TP.

TP3 - Adresse IP et notation CIDR

1. On considère l'adresse IP : **198.165.145.58/24** en notation CIDR.

- Écrire le masque en écriture binaire :

- Écrire le masque en écriture décimale :

- En déduire l'adresse du réseau en écriture binaire :

- En déduire l'adresse du réseau en écriture décimale :

2. On considère l'adresse IP : **198.165.145.58/20** en notation CIDR.

- Écrire le masque en écriture binaire :

- Écrire le masque en écriture décimale :

- En déduire l'adresse du réseau en écriture binaire :

- En déduire l'adresse du réseau en écriture décimale :



Lever la main pour valider ce TP.

TP4 - Classe IP et notation CIDR en Python

- Récupérer le fichier **cidr.py** donné en ressource et l'enregistrer dans le dossier **TP-Chapitre8** sur votre compte.
- Compléter ci-dessous le code de la fonction **ad_reseau_cidr()** qui renvoie l'adresse IP du réseau.

```
def ad_reseau_cidr(ip_cidr1):
    liste = ip_cidr1.split('/')
    ip1 = IP(liste[0])
    nb_1 = int(liste[1]) # Nombre de 1 dans le masque
    chaine = ""
    octet = ""
    i = 0
    while i < 32:
        #print(chaine)
        if i < nb_1 :
            octet += 
        else :
            octet += 
        i += ....
        if i % 8 == 0 and i < 32 :
            chaine += str(int(octet,2)) + 
            octet = ""
        if i == 32 :
            chaine += 
    masque = IP(chaine)
    return adresse_reseau(ip1, masque)
```

- Rédiger cette fonction dans le fichier **cidr.py** avec **Thonny**.
- Vérifier que cette fonction s'exécute correctement. Exemple de retours attendus dans la console :

```
>>> print(ad_reseau_cidr("192.168.145.12/24"))
192.168.145.0

>>> print(ad_reseau_cidr("192.168.129.32/20"))
192.168.128.0
```

- On considère l'adresse IP : **192.198.127.19/23** en notation CIDR.
Utiliser la fonction **ad_reseau_cidr()** pour déterminer l'adresse du réseau :



Lever la main pour valider ce TP.

TP5 - Nombre d'hôtes du réseau

1. On considère l'adresse IP : **208.64.127.11/24** en notation CIDR.

- Écrire le masque en écriture décimale :

- Écrire l'adresse du réseau en écriture décimale :

- Écrire l'adresse de diffusion de ce réseau en écriture décimale :

- Déterminer le nombre d'hôtes de ce réseau :

1. On considère l'adresse IP : **128.64.65.10/20** en notation CIDR.

- Écrire le masque en écriture décimale :

- Écrire l'adresse du réseau en écriture décimale :

- Écrire l'adresse de diffusion de ce réseau en écriture décimale :

- Déterminer le nombre d'hôtes de ce réseau :



Lever la main pour valider ce TP.

TP6 - Nom de domaine

En utilisant le lien suivant

<https://www.my-ip-finder.fr/dnslookup-nom-de-domaine-ip-et-localisation/>

déterminer l'adresse IP des noms de domaine suivant :

- www.ac-grenoble.fr

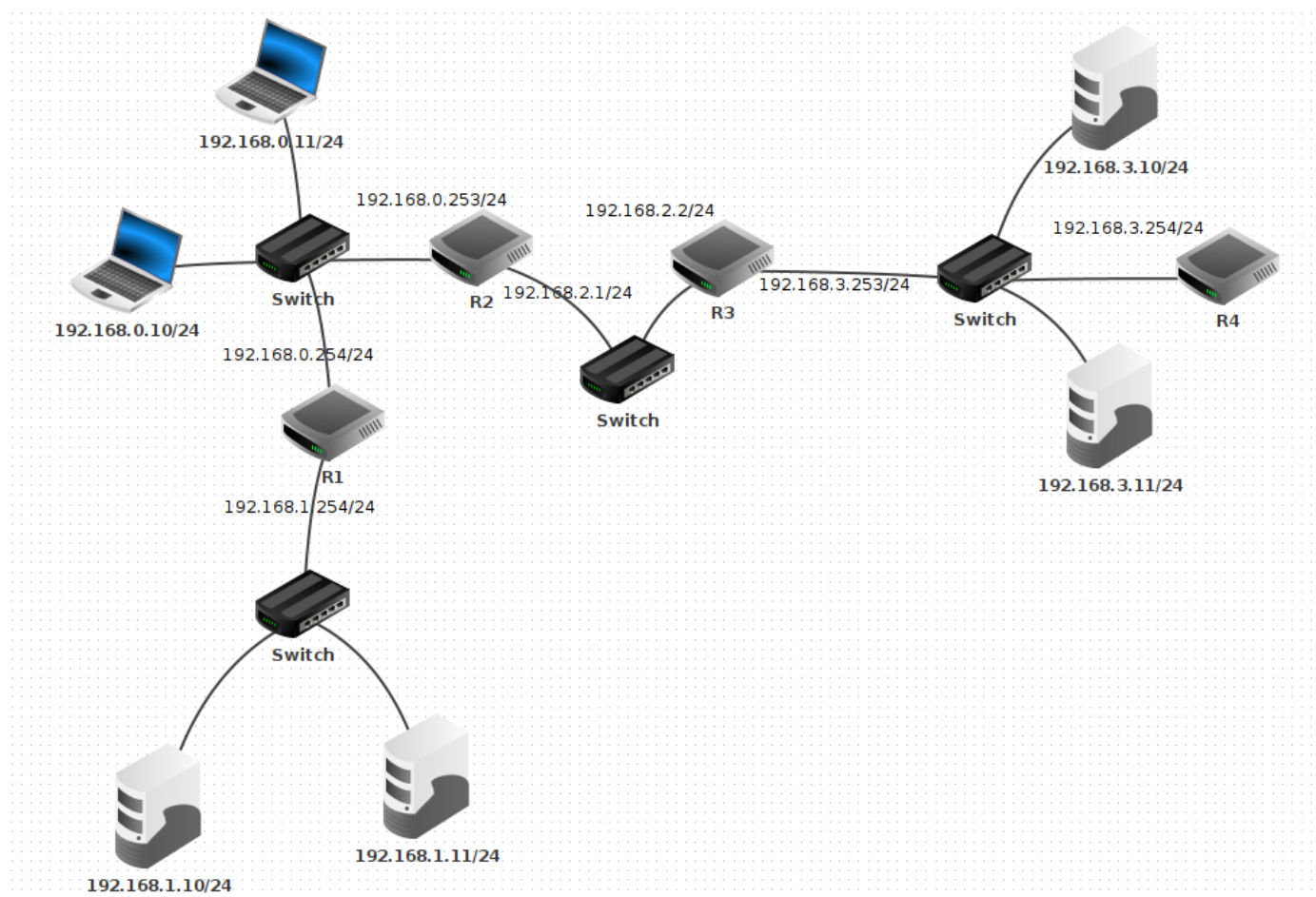
- www.impots.gouv.fr



Lever la main pour valider ce TP.

TP7 - Exemple de réseau

On considère le réseau ci-dessous :



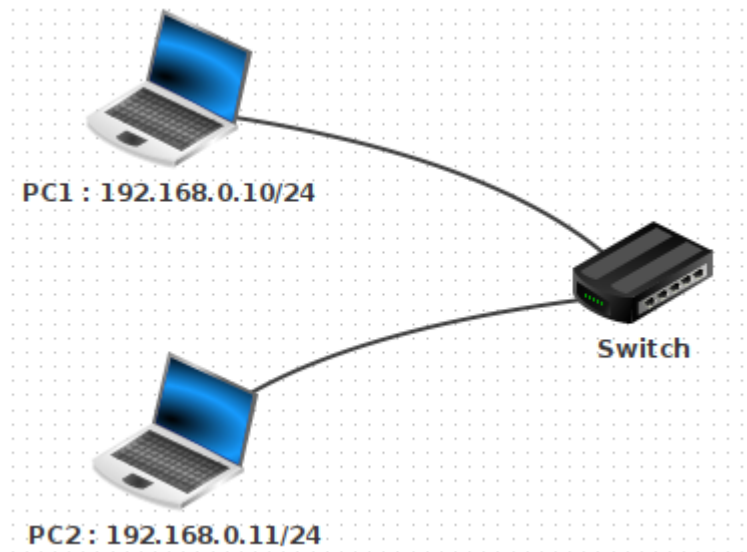
Ce réseau contient quatre sous-réseaux. Donner ci-dessous les adresses réseaux de ces quatre sous-réseaux :





Lever la main pour valider ce TP.

TP8 - Simuler un réseau avec Filius : étape1

- Lancer le logiciel **FILIUS** et créer un nouveau fichier que l'on enregistrera dans le dossier **TP-Chapitre8** sur votre compte, sous le nom **res1.flc**.
- Créer un réseau de deux ordinateurs reliés par un **switch** comme ci-dessous :

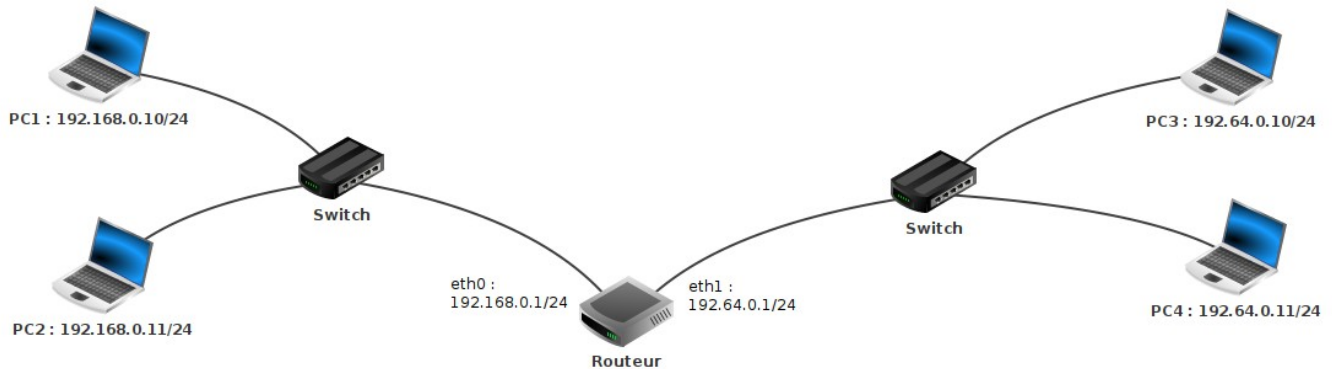



- Configurer bien les adresses IP des deux ordinateurs comme indiqué ci-dessus.
- Passer en mode simulation en cliquant sur la flèche verte : .
- En faisant un clic droit sur le PC1, choisir **Afficher le bureau** puis cliquer sur **Installation des logiciels** et avec la flèche verte installer sur le PC1 la **Ligne de commande** puis cliquer sur **Appliquer les modifications**.
- Cliquer ensuite sur **Ligne de commande** que l'on vient d'installer sur le PC1.
- Taper la commande **ipconfig** pour vérifier que le PC1 a bien l'adresse IP : 192.168.0.10.
- Taper le commande **ping 192.168.0.11** pour voir si on arrive à envoyer des paquets au PC2.
- Taper le commande **tracert 192.168.0.11** pour voir la route empruntée par les paquets.
- On arrête la simulation en cliquant sur l'icône du marteau :  et on repasse en **mode conception**.

 **Lever la main pour valider ce TP.**

TP9 - Simuler un réseau avec Filius : étape2

- Dans le logiciel **FILIUS** enregistrer le fichier **res1.fls** sous le nom **res2.fls** .
- Augmenter le réseau, en rajoutant du matériels comme dans l'exemple ci-dessous.
Bien configurer toutes les adresses IP notamment celles des deux interfaces du routeur.




- Ne pas oublier de mentionner les Passerelles pour chaque PC .
- Passer en mode simulation en cliquant sur la flèche verte :  .
- Lancer ensuite la **Ligne de commande** du PC1.
- Taper la commande **ipconfig** pour vérifier que le PC1 a bien sa passerelle de renseignée.

L'adresse IP Passerelle pour le PC1 est :

L'adresse IP Passerelle pour le PC3 est :

- Taper le commande **ping 192.64.0.10** pour voir si on arrive à envoyer des paquets au PC3.
- Taper le commande **tracert 192.64.0.10** pour voir la route empruntée par les paquets.

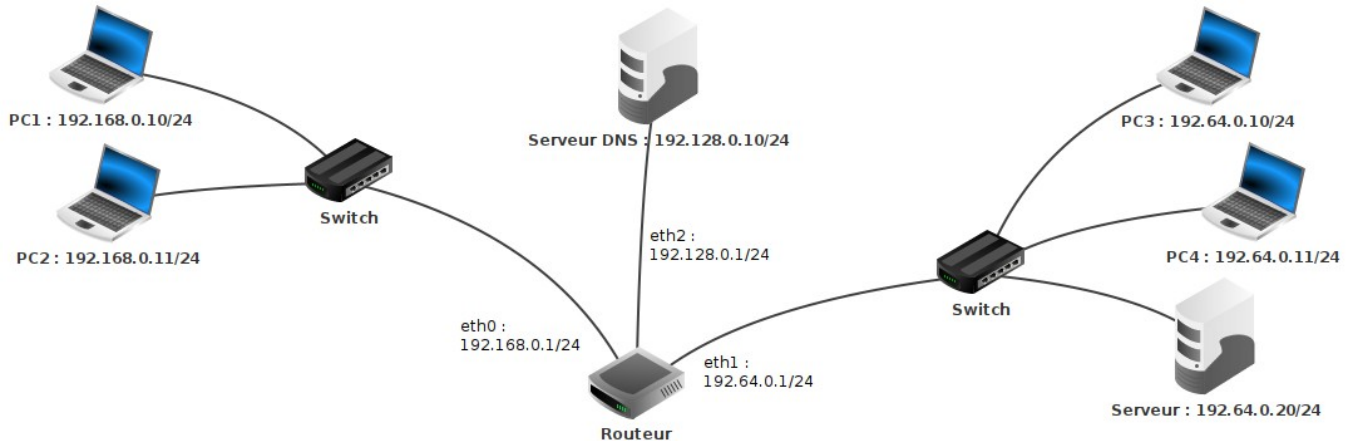
Route obtenue :

- On arrête la simulation en cliquant sur l'icône du marteau :  et on repasse en mode conception.

 **Lever la main pour valider ce TP.**

TP10 - Ajouter un serveur DNS et un serveur web avec Filius : étape3

- Dans le logiciel **FILIUS** enregistrer le fichier **res2.fls** sous le nom **res3.fls**.
- Ajouter un serveur d'adresse **192.64.0.20** dans le réseau dont l'adresse réseau est **192.64.0.0/24** comme dans la capture ci-dessous. Bien configurer ce nouveau poste. Passer en mode simulation et installer un **Serveur web** sur ce poste et le démarrer.



- Ajouter une interface réseau au routeur avec l'adresse **192.128.0.1/24** pour cela faire un clic droit sur le routeur, puis choisir **Configurer**. Dans l'onglet **Général** cliquer sur **Gérer les connexions** et rajouter une interface réseau en cliquant sur le **+**.
- Ajouter ensuite un serveur DNS connecté à cette interface réseau d'adresse IP **192.128.0.10/24** comme dans la capture ci-dessus. Passer ensuite en mode simulation et installer un **Serveur DNS** sur ce serveur. Cliquer sur ce serveur DNS et dans l'onglet **Adresse (A)** entrer le nom de domaine : **www.nsi.com** avec l'adresse du serveur **192.64.0.20**. Démarrer le serveur DNS.
- Repasser en mode conception, et pour tous les postes, renseigner l'adresse du serveur DNS en mettant, l'adresse du serveur DNS : **192.128.0.10**.
- Repasser en mode simulation et lancer le terminal du PC1.
- Taper la commande **host www.nsi.com** pour vérifier qu'il trouve bien l'adresse IP du serveur web.
- Installer un **Navigateur web** sur le PC1, cliquer dessus et taper dans la barre d'URL : **www.nsi.com** et vérifier qu'on obtient bien la page d'accueil du site comme ci-dessous.



Lever la main pour valider ce TP.

TP11 - Étude d'un réseau avec Filius

- Récupérer le fichier **res4.flis** en ressource.
- Ouvrir ce fichier avec le logiciel **FILIUS** .
- L'adresse IP du poste **M14** est :
- L'adresse IP du poste **M9** est :
- Lancer le terminal du poste **M14** et utiliser la commande **tracert** pour trouver la route des paquets pour aller du poste **M14** au poste **M9** .

Route obtenue :

- Repasser en mode conception, et supprimer le câble entre le routeur F et le routeur E.
- Attendre quelques minutes, pour que les tables de routage se recalculent.
- Repasser en mode simulation et lancer le terminal du poste **M14** . Vérifier avec la commande **ping** que le poste **M9** est accessible. Puis, utiliser la commande **tracert** pour trouver la nouvelle route des paquets pour aller du poste **M14** au poste **M9** .

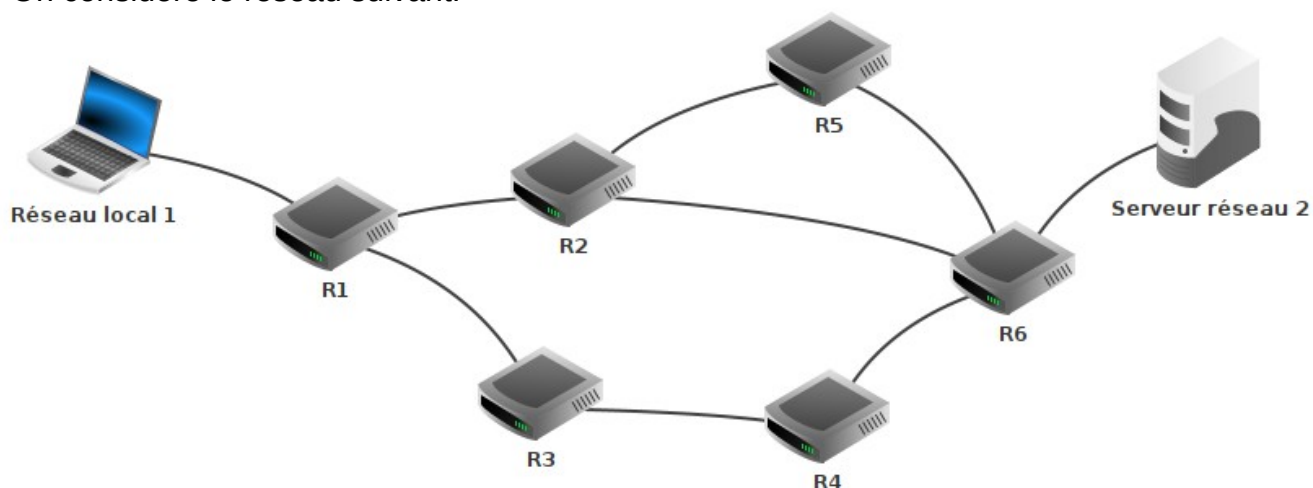
Nouvelle route :



Lever la main pour valider ce TP.

TP12 - Table de routage simplifiée avec le protocole RIP

On considère le réseau suivant.



Compléter les table de routage suivant en respectant le protocole RIP.

Table de routage du routeur R1		
Destinataire	Passerelle	Nb de sauts
R2		
R3		
R4		
R5		
R6		

Table de routage du routeur R2		
Destinataire	Passerelle	Nb de sauts
R1		
R3		
R4		
R5		
R6		

Table de routage du routeur R3		
Destinataire	Passerelle	Nb de sauts
R1		
R2		
R4		
R5		
R6		

Table de routage du routeur R4		
Destinataire	Passerelle	Nb de sauts
R1		
R2		
R3		
R5		
R6		

Table de routage du routeur R5		
Destinataire	Passerelle	Nb de sauts
R1		
R2		
R3		
R4		
R6		

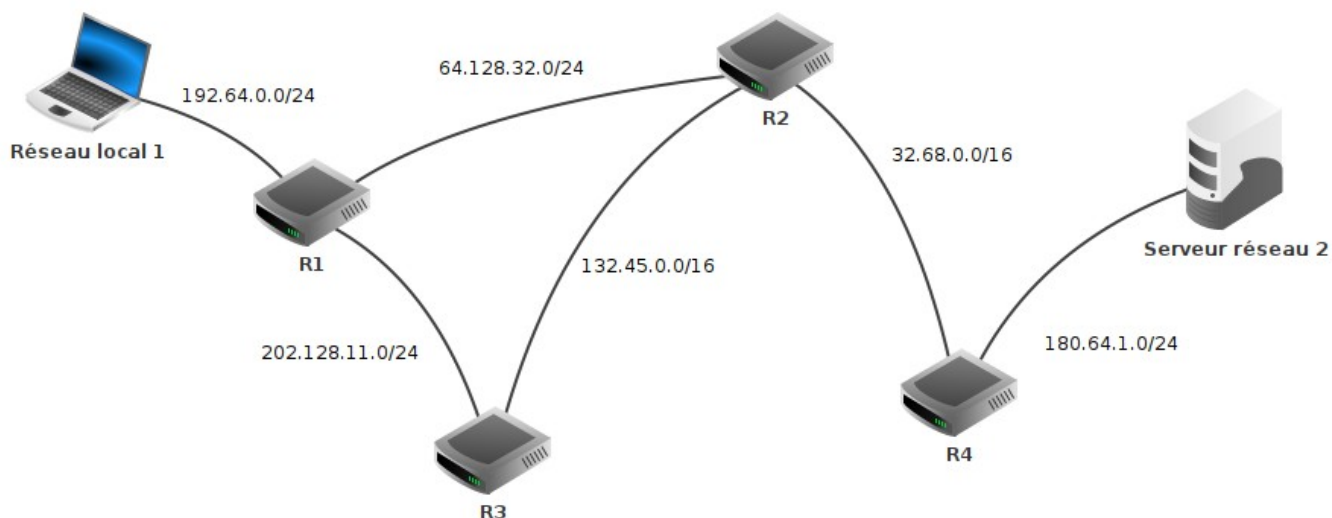
Table de routage du routeur R6		
Destinataire	Passerelle	Nb de sauts
R1		
R2		
R3		
R4		
R5		



Lever la main pour valider ce TP.

TP13 - Table de routage détaillée avec le protocole RIP

On considère le réseau suivant.



Compléter les table de routage suivant en respectant le protocole RIP.

Table de routage du routeur R1		
Destinataire	Passerelle	Nb de sauts
192.64.0.0		
202.128.11.0		
64.128.32.0		
32.68.0.0		
180.64.1.0		

Table de routage du routeur R3		
Destinataire	Passerelle	Nb de sauts
192.64.0.0		
202.128.11.0		
64.128.32.0		
32.68.0.0		
180.64.1.0		

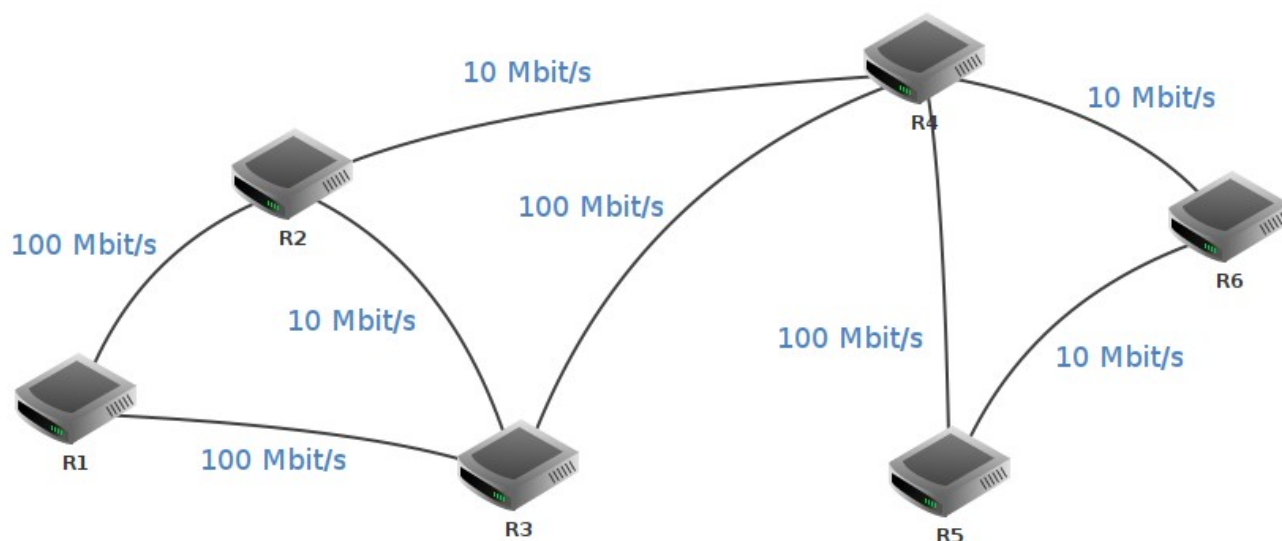
Table de routage du routeur R2		
Destinataire	Passerelle	Nb de sauts
192.64.0.0		
202.128.11.0		
64.128.32.0		
32.68.0.0		
180.64.1.0		

Table de routage du routeur R4		
Destinataire	Passerelle	Nb de sauts
192.64.0.0		
202.128.11.0		
64.128.32.0		
32.68.0.0		
180.64.1.0		

 Lever la main pour valider ce TP.

TP14 - Table de routage avec le protocole OSPF

On considère le réseau suivant.



Compléter les tables de routage ci-dessous en respectant le protocole OSPF.

Table de routage du routeur R1		
Destinataire	Passerelle	Distance
R2		
R3		
R4		
R5		
R6		

Table de routage du routeur R2		
Destinataire	Passerelle	Distance
R1		
R3		
R4		
R5		
R6		

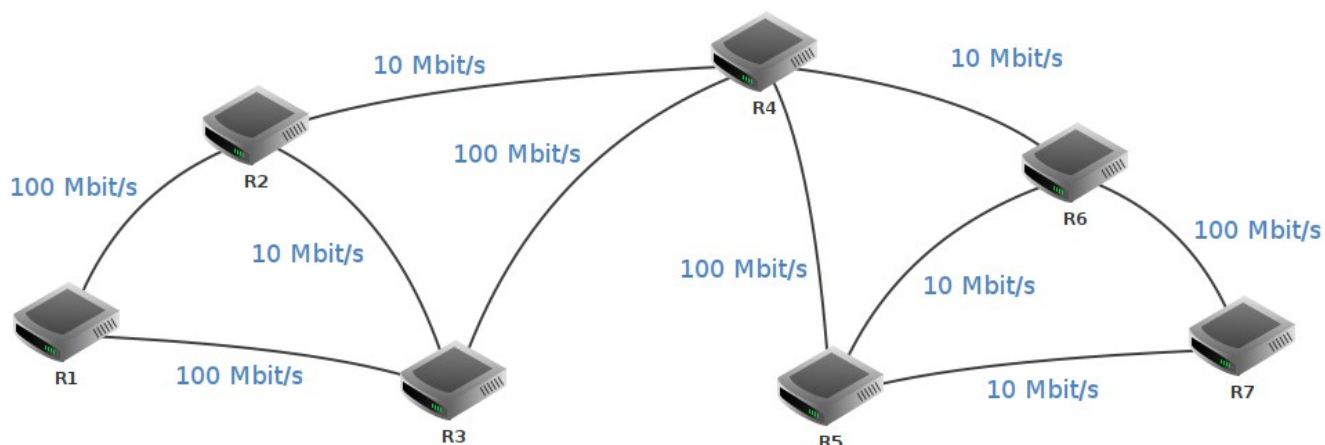
Table de routage du routeur R3		
Destinataire	Passerelle	Distance
R1		
R2		
R4		
R5		
R6		

Table de routage du routeur R4		
Destinataire	Passerelle	Distance
R1		
R2		
R3		
R5		
R6		

 Lever la main pour valider ce TP.

TP15 - Algorithme de Dijkstra

On considère le réseau suivant.



Appliquer l'algorithme de Dijkstra pour déterminer la route la plus courte selon le protocole OSPF entre le routeur R1 et le routeur R7.

R1	R2	R3	R4	R5	R6	R7	Sélection
0	oo	oo	oo	oo	oo	oo	R1

La route la plus courte est et sa distance est .



Lever la main pour valider ce TP.