Exercice 1 (6 points)

Cet exercice porte sur la programmation, les réseaux et les protocoles de routage.

Rappels :

Une adresse IPv4 est composée de 4 octets, soit 32 bits. Elle est notée `a.b.c.d`, où `a`, `b`, `c` et `d` sont les écritures décimales des valeurs des 4 octets. Cette écriture est nommée *notation décimale pointée*.

La notation `a.b.c.d/n` est appelée notation *CIDR* (*Classless Inter Domain Routing*), l'entier `n` représentant le *masque* du réseau. Les `n` premiers bits à gauche dans l'adresse IP représentent la partie *réseau*, les bits à droite qui suivent représentent la partie *machine*.

- L'adresse IPv4 dont tous les bits de la partie machine sont à `0` est appelée *adresse du réseau*.
- L'adresse IPv4 dont tous les bits de la partie machine sont à `1` est appelée *adresse de diffusion*.
- Le masque du réseau est composé de 4 octets : les `n` premiers bits à gauche sont égaux à `1` et les bits à droite qui suivent sont égaux à `0`.

Dans un réseau informatique, lorsqu'une machine cherche à transmettre des données à une autre machine, elle les transmet sans passer par un routeur si le destinataire fait partie du même réseau, sinon, elle transmet les données à un routeur qui fait office de passerelle entre les différents réseaux.

Dans le schéma réseau de la figure 1, toutes les machines du réseau `192.168.1.0/24` ont pour adresse de passerelle celle de l'interface `G0` du routeur `R1`, soit `192.168.1.254`.

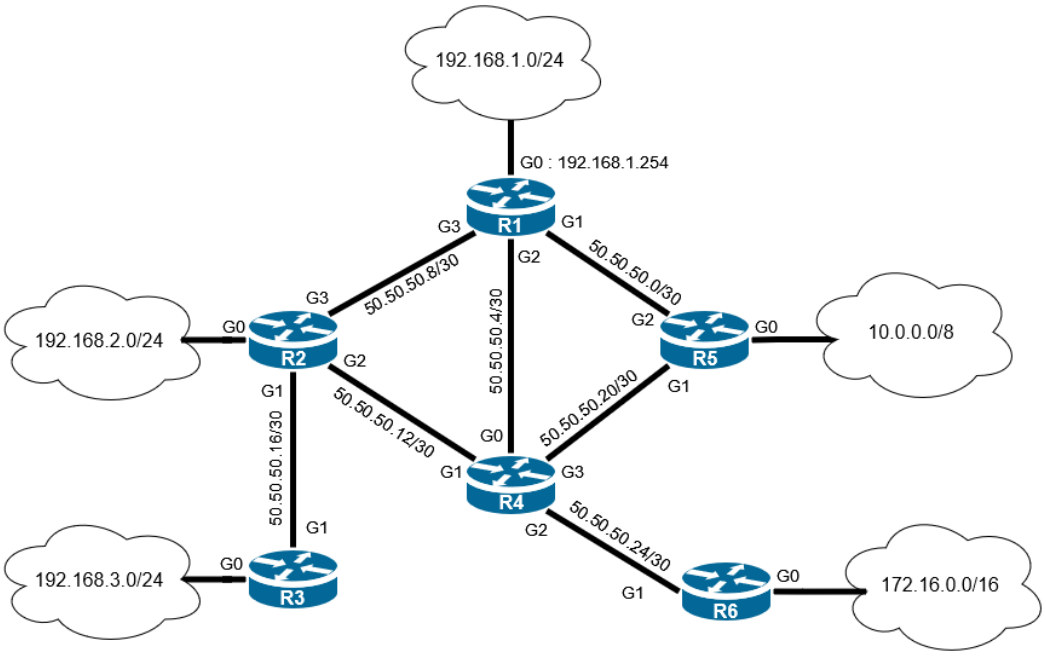

Figure 1. Réseau

1. Donner le nom, ainsi que l'interface, du routeur qui constitue la passerelle pour les machines du réseau `192.168.2.0/24`.

La politique d'attribution des adresses IP dans les réseaux nous impose de choisir la dernière adresse IP disponible dans le réseau pour la passerelle. Ainsi, dans le réseau `192.168.1.0/24`, la passerelle a pour adresse IP `192.168.1.254`.

2. Donner l'adresse IP à attribuer à la passerelle du réseau `172.16.0.0/16`.

Dans le réseau `50.50.50.4/30`, l'interface G2 du routeur R1 a pour adresse IP `50.50.50.5`.

3. Lister les quatre adresses du réseau `50.50.50.4/30` et attribuer une adresse IP à l'interface G0 du routeur R4.

Pour choisir la bonne interface de sortie, la passerelle utilise une table de routage qui identifie une interface par où sortir pour trouver le réseau de destination des données et un nombre appelé *métrique* qui représente le coût de la liaison. Cette métrique dépend du type de routage mis en œuvre, manuel (statique) ou automatique (protocoles RIP, OSPF, …). Dans le cas d'un routage automatique utilisant le protocole RIP, la métrique correspond au nombre minimum de routeurs à traverser pour joindre le réseau de destination.

4. Recopier et compléter les lignes manquantes de la table de routage du routeur R1 dans le cas d'un routage automatique utilisant le protocole RIP. En cas d'égalité de métrique, on choisira l'interface de numéro le plus faible.

Table de routage du routeur R1		
Réseau de destination	Interface de sortie	Métrique
192.168.1.0/24	G0	0
192.168.2.0/24		
192.168.3.0/24		
172.16.0.0/16	G2	2
10.0.0.0/8	G1	1
50.50.50.0/30	G1	0
50.50.50.4/30	G2	0
50.50.50.8/30	G3	0
50.50.50.12/30	G2	1
50.50.50.16/30	G3	1
50.50.50.20/30		
50.50.50.24/30		

On décide de modéliser la table de routage du routeur R1 par un tableau de triplets contenant l'adresse du réseau de destination et son masque en notation CIDR (sous forme d'un quintuplet), le nom de de l'interface de sortie vers le réseau de destination et la métrique.

5. Recopier et compléter les lignes manquantes pour définir le tableau `t_routage` pour qu'il modélise complètement la table de routage du routeur R1 (vous écrirez les numéros de lignes complétées).

```
 1 t_routage = [ ((192, 168, 1, 0, 24), 'G0', 0),
 2               ...,
 3               ...,
 4               ((172,  16,  0,  0, 16), 'G2', 2),
 5               ((10,    0,  0,  0,  8), 'G1', 1),
 6               ((50,   50, 50,  0, 30), 'G1', 0),
 7               ((50,   50, 50,  4, 30), 'G2', 0),
 8               ((50,   50, 50,  8, 30), 'G3', 0),
 9               ((50,   50, 50, 12, 30), 'G2', 1),
10               ((50,   50, 50, 16, 30), 'G3', 1),
11               ...,
12               ...
13             ]
```

On trouve l'adresse du réseau auquel appartient une adresse IP en appliquant un opérateur ET bit à bit entre l'adresse IP et le masque de sous-réseau.

Par exemple, pour trouver le réseau auquel appartient l'adresse IP `192.168.1.10/24` on fait :

```
adresse IP :  11000000.10101000.00000001.00001010
masque     :  11111111.11111111.11111111.00000000
           &  --------------------------------
              11000000.10101000.00000001.00000000
```

On peut conclure que l'adresse IP `192.168.1.10/24` appartient au réseau `192.168.1.0/24`.

On dispose d'une fonction `et_bit_a_bit` qui renvoie le résultat de l'opération ET bit à bit entre deux entiers. Ainsi `et_bit_a_bit(10, 252)` renverra 8 car $8 = (0000\ 1000)_2$, $10 = (0000\ 1010)_2$ et $252 = (1111\ 1100)_2$.

De plus, on dispose d'une fonction `mask_for_size(size)` qui prend en paramètre un entier `size` qui représente un masque de sous-réseau en notation CIDR et le renvoie en notation décimale sous la forme d'un quadruplet (a, b, c, d) d'entiers compris entre 0 et 255.

Exemples :

```
>>> mask_for_size(24)
(255, 255, 255, 0)

>>> mask_for_size(26)
(255, 255, 255, 192)
```

6. Donner la sortie de l'exécution du code suivant.

   ```
   >>> mask_for_size(30)
   ```

7. Déterminer à quel réseau appartient l'adresse IP `50.50.50.6/30` en écrivant l'opération ET bit à bit effectuée. On convertira 50 et 6 en binaire.

8. Recopier et compléter la fonction `is_in_network(address, network)` qui prend en paramètres une adresse IP `address` et l'adresse d'un réseau `network`, chacune fournie sous la forme d'un tuple, et qui renvoie `True` si l'adresse IP appartient au réseau, et `False` sinon.

   Exemples :

   ```
   >>> is_in_network((192, 168, 1, 1), (192, 168, 1, 0, 24))
   True
   >>> is_in_network((192, 168, 1, 1), (192, 168, 2, 0, 24))
   False
   ```

```
1 def is_in_network(address, network):
2     network_mask = mask_for_size(...)
3     for i in range(4):
4         if et_bit_a_bit(network_mask[i], address[i]) != ...:
5             return ...
6     return ...
```

Le routeur sélectionne l'interface de sortie vers le réseau auquel appartient l'adresse IP de destination.

9. Écrire la fonction `choose_interface(t_routage, destination_ip)` qui prend en paramètres un tableau `t_routage` qui modélise une table de routage et un quadruplet `destination_ip` qui représente une adresse IP, et qui renvoie l'interface de sortie du routeur si l'adresse IP `destination_ip` est dans un des réseaux présents dans `t_routage`. Si l'adresse IP de destination n'est pas présente, la fonction renvoie `None`.

   Exemples :

   ```
   >>> choose_interface(t_routage, (192, 168, 1, 12))
   G0
   >>> choose_interface(t_routage, (192, 168, 5, 12))
   None
   ```

Dans le cas d'un routage automatique utilisant le protocole OSPF, la métrique tient compte du débit des liaisons dont le coût est calculé selon la formule suivante (le débit est donné en bits/seconde) :

$$\text{coût} = \frac{10^{10}}{\text{Débit}}$$

Les débits des liaisons entre les routeurs sont donnés ci-dessous :

Liaisons inter-routeurs	Type	Débit
R1 - R2	Gigabit Ethernet	1 Gb/s
R1 - R4	Fast Ethernet	100 Mb/s
R1 - R5	Gigabit Ethernet	1 Gb/s
R2 - R3	Gigabit Ethernet	1 Gb/s
R2 - R4	Fibre	10 Gb/s
R4 - R5	Gigabit Ethernet	1 Gb/s
R4 - R6	Fibre	10 Gb/s

10. Calculer les coûts des trois types de liaisons.

11. Recopier et compléter les lignes manquantes de la table de routage du routeur R1 dans le cas d'un routage automatique utilisant le protocole OSPF. Un réseau directement connecté au routeur a une métrique de `0`, sinon, la métrique est le coût minimum pour joindre le réseau de destination.

Table de routage du routeur R1		
Réseau de destination	Interface de sortie	Métrique
`192.168.1.0/24`	G0	0
`192.168.2.0/24`		
`192.168.3.0/24`	G3	20
`172.16.0.0/16`		
`10.0.0.0/8`		
`50.50.50.0/30`	G1	0
`50.50.50.4/30`	G2	0
`50.50.50.8/30`	G3	0
`50.50.50.12/30`	G3	10
`50.50.50.16/30`	G3	10
`50.50.50.20/30`		
`50.50.50.24/30`	G3	11