Architecture et OS	Chap 01	**Routage**

1.1 Rappels et contextualisation

1.1.1 Internet

Un peu de contexte

Internet est un **réseau de réseaux** : il relie entre eux des millions de réseaux locaux (maison, lycée, entreprises, datacenters, etc.). Chaque réseau local contient un nombre limité de machines, connectées à un commutateur (**switch**), et il est relié au reste d'Internet via un **routeur**.

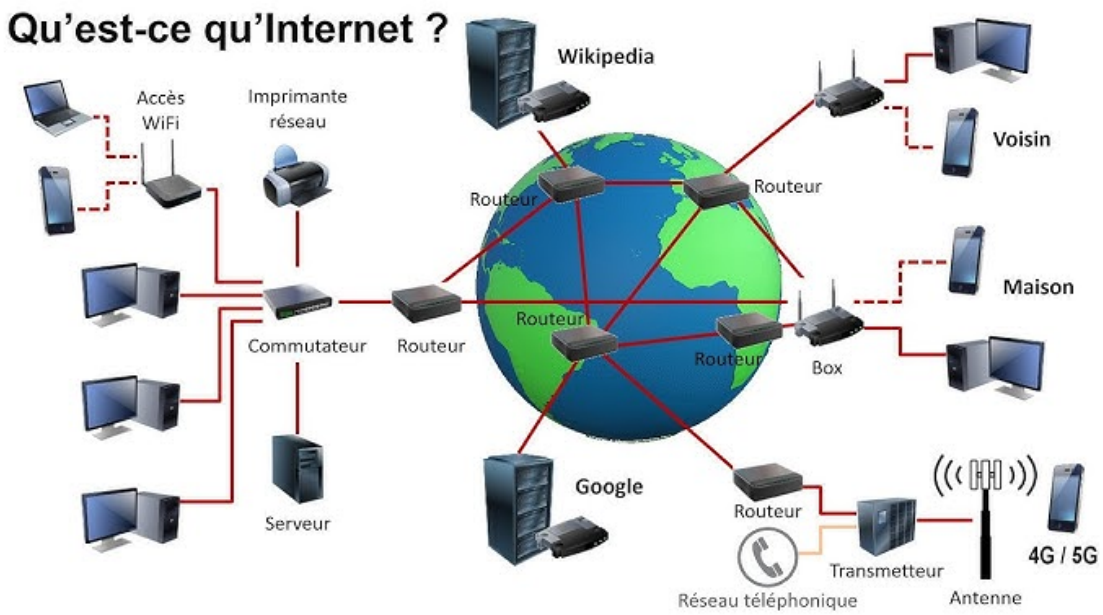
Lorsqu'une machine envoie un message (par exemple une requête web), ce message est découpé en **paquets** qui vont devoir traverser plusieurs réseaux intermédiaires. Le problème à résoudre est alors toujours le même :

Comment faire parvenir un paquet depuis une machine source jusqu'à une machine destination, située n'importe où dans le monde ?

Pour répondre à cette question, il est nécessaire :

- d'**identifier** de manière unique la source et la destination ;
- d'**organiser** les machines en sous-réseaux ;
- d'**acheminer** les paquets de réseau en réseau.

Le protocole **IP** (Internet Protocol) fournit un adressage permettant d'identifier les machines et leurs réseaux. Le **routage** désigne l'ensemble des mécanismes qui permettent, ensuite, de choisir un chemin à travers les routeurs pour atteindre la destination.



1.1.2 Les adresses IP

Définition 1 — Adresse IP

Une **adresse IP** est un identifiant numérique attribué à une **interface réseau** (ordinateur, téléphone, TV etc..) utilisant le protocole IP.
Elle sert principalement à deux choses :

- **identifier** l'interface destination (où le paquet doit arriver) ;
- permettre aux routeurs de **transférer** le paquet de réseau en réseau dans la bonne direction.

Exemple 1 — La Poste

On peut comparer une adresse IP à une adresse postale :

- la **partie réseau** joue le rôle de la ville ;
- la **partie machine** correspond à une maison dans cette ville.

Pour distribuer correctement le courrier, il faut connaître à la fois la ville et la maison. De même, pour acheminer un paquet, il faut identifier à la fois le réseau de destination et la machine à l'intérieur de ce réseau.

Remarque 1. Une adresse IP n'identifie pas une personne. Elle peut changer au cours du temps (adresse attribuée par le fournisseur d'accès, par un serveur DHCP, changement de réseau Wi-Fi, etc.).

Définition 2 — Adresse IPv4

Une **adresse IPv4** est codée sur **32 bits** (un **bit** est un chiffre 0 ou 1).
On l'écrit le plus souvent sous forme **décimale pointée** :

$$a.b.c.d$$

où chaque nombre correspond à un **octet** (8 bits), donc une valeur entre 0 et 255.

Plage de valeur

Le plus petit nombre représentable sur 8 bits est :

$$00000000_2 = 0$$

Le plus grand nombre représentable sur 8 bits est :

$$11111111_2 = 2^7 + 2^6 + \cdots + 2^0 = 255.$$

Cela nous donne donc 256 valeurs possibles.

Remarque 2. Le 2 en indice nous indique qu'on écrit en base 2, en binaire.

4 octets c'est trop peu.

Le protocole IPv4 utilise des adresses codées sur **32 bits**, ce qui permet de définir environ 2^{32} adresses distinctes (soit un peu plus de 4 milliards). Avec la généralisation d'Internet (objets connectés, smartphones, serveurs, etc.), ce nombre s'est révélé insuffisant.

Pour répondre à cette pénurie d'adresses, un nouveau protocole a été conçu : **IPv6**. Les adresses IPv6 sont codées sur **16 octets** donc **128 bits**, ce qui permet un nombre d'adresses considérablement plus grand (2^{128}).

Elles sont écrites en **hexadécimal** (que vous devez maîtriser également mais nous ne revenons pas dessus dans ce cours), ce qui rend leur écriture plus compacte malgré leur taille. IPv4 et IPv6 coexistent aujourd'hui, le passage à IPv6 se faisant progressivement.

1.1.3 Conversions

Lecture d'un octet avec les puissances de 2

Un octet $(b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0)_2$ représente le nombre :

$$b_7 \cdot 2^7 + b_6 \cdot 2^6 + \cdots + b_1 \cdot 2^1 + b_0 \cdot 2^0.$$

Autrement dit, chaque bit indique si l'on « prend » ou non la puissance de 2 correspondante.

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1

Exemple 2 — Exemple de lecture d'un octet

Considérons l'octet binaire suivant :

$$(11001010)_2$$

On repère les bits à 1 et on additionne les puissances de 2 correspondantes :

$$11001010_2 = 1 \cdot 128 + 1 \cdot 64 + 0 \cdot 32 + 0 \cdot 16 + 1 \cdot 8 + 0 \cdot 4 + 1 \cdot 2 + 0 \cdot 1$$

On obtient donc :

$$11001010_2 = 202_{10}.$$

Exercice 1 — Conversion d'adresses IP

Écrire sous forme décimale pointée les adresses IP suivantes :

1. `11000000.10101000.00000001.00001110`

2. `10000010.00010000.00001010.00100001`

Définition 3 — Méthode des divisions successives

Pour convertir un nombre décimal en un nombre binaire, on applique la méthode suivante :
Pour convertir N_{10} en binaire :

- on divise N par 2,
- on note le reste (0 ou 1),
- on recommence avec le quotient,
- on lit les restes **de bas en haut**.

Exemple 3 — Décimal $\rightarrow$ binaire

Convertissons 45_{10} en binaire.

Nombre	Division par 2	Reste
45	$45 = 2 \times 22 + 1$	1
22	$22 = 2 \times 11 + 0$	0
11	$11 = 2 \times 5 + 1$	1
5	$5 = 2 \times 2 + 1$	1
2	$2 = 2 \times 1 + 0$	0
1	$1 = 2 \times 0 + 1$	1

On lit les restes du bas vers le haut :

$$45_{10} = (101101)_2.$$

On peut vérifier en décomposant :

$$(101101)_2 = 32 + 8 + 4 + 1 = 45_{10}.$$

Exercice 2 — Conversion base d'entiers

Convertir en base 2 les nombres suivants :

1. 615_{10}

2. 2048_{10}

Exercice 3 — Conversion d'adresses IP

Ecrire en binaire les adresses IP suivantes :

1. `10.0.0.1`

2. `172.16.5.3`

1.1.4 Masque de sous-réseau et calcul de l'adresse réseau

Pourquoi un masque de sous-réseau ?

Dans un réseau IP, toutes les machines ne sont pas directement reliées entre elles : elles sont regroupées en **sous-réseaux**. On associe alors à chaque adresse un **masque de sous-réseau** dont le rôle est de préciser pour une adresse IP donnée :

- quels bits de l'adresse IP identifient le **réseau** ;
- quels bits identifient la **machine** à l'intérieur de ce réseau.

Deux machines peuvent communiquer directement si et seulement si elles appartiennent au **même réseau**. Déterminer le réseau d'une adresse IP est donc une opération fondamentale.

Définition 4 — Masque de sous-réseau

Un masque de sous-réseau est un nombre de **32 bits** composé :

- de bits à 1 pour la partie **réseau** ;
- de bits à 0 pour la partie **machine**.

Il est souvent noté sous la forme `/n`, où n est le nombre de bits à 1 dans le masque.

Exemple 4 — Un masque

Le masque $/12$ correspond donc à $11111111.11111111.11110000.00000000$ en binaire

Calcul de l'adresse réseau

L'**adresse réseau** d'une machine est obtenue en :

- conservant les bits réseau ;
- mettant à 0 tous les bits machine.

En pratique, ce calcul revient à effectuer un **ET logique (multiplication logique)** entre l'adresse IP et le masque de sous-réseau :

$$\text{adresse réseau} = \text{adresse IP } \textbf{ET} \text{ masque}.$$

Exemple 5 — Une adresse avec un masque

Soit l'adresse :

$$\texttt{172.16.131.9/20}$$

Un masque $/20$ signifie :

- 20 bits pour le réseau ;
- $32 - 20 = 12$ bits pour les machines.

Le masque binaire correspondant est :

$$11111111.11111111.11110000.00000000$$

On écrit l'adresse IP en binaire :

Adresse IP	:	10101100	10010000	00000011	00001001
Masque	:	11111111	11111111	11110000	00000000
ET logique	:	10101100	10010000	00000000	00000000

Chaque bit du résultat vaut 1 uniquement si les deux bits étaient à 1. Les bits machine sont automatiquement mis à 0.
On obtient donc l'adresse réseau :

$$\texttt{172.16.128.0/20}.$$

Brancher des machines sur un réseau

Une fois l'adresse réseau déterminée, la **première adresse utilisable** correspond à la **première machine** pouvant être connectée au réseau. Elle est obtenue en ajoutant 1 à l'adresse réseau (c'est-à-dire en mettant à 1 le bit de poids le plus faible de la partie machine).
Dans l'exemple précédent, l'adresse réseau est :

`172.16.128.0/20`.

La première machine du réseau a donc pour adresse :

`172.16.128.1/20`.

L'adresse `172.16.128.0` étant réservée pour identifier le réseau, elle ne peut pas être attribuée à une machine.

Exercice 4 — Déterminer l'adresse réseau

Calculer l'adresse réseau associée aux adresses suivantes :

1. `192.168.14.67/26`

   ---

2. `172.31.129.200/19`

Définition 5 — Capacité utilisable d'un réseau

Si un réseau utilise n bits pour la partie réseau, il reste $32 - n$ bits pour identifier les machines.
Le nombre total d'adresses possibles est alors :

$$2^{32-n}.$$

Parmi ces adresses :

- la **première** correspond à l'**adresse réseau** ;
- la **dernière** correspond à l'**adresse de broadcast**.

Ces deux adresses ne peuvent pas être attribuées à des machines. Le nombre d'**adresses utilisables** est donc :

$$2^{32-n} - 2.$$

Exemple 6 — Réseau /20

Un réseau /20 dispose de $32 - 20 = 12$ bits pour les machines, soit :

$$2^{12} = 4096$$

adresses au total.
En retirant l'adresse réseau et l'adresse de broadcast, il reste :

$$4096 - 2 = 4094$$

adresses réellement attribuables à des machines.

Exercice 5 — Capacité de réseaux

Pour chaque masque, indiquer le **nombre d'adresses utilisables** pour brancher les machines :

1. /16

   ---

2. /20

   ---

3. /28

1.1.5 Réseau local et communication

Définition 6 — Réseau local

Un **réseau local** est un ensemble de machines qui partagent la même adresse réseau.

Local ou distant ?

Pour envoyer un paquet, une machine compare :

- son adresse réseau ;
- l'adresse réseau de la destination.
- si les deux adresses réseau sont identiques, la communication est **locale** ;
- sinon, la communication est **distante** et passe par la **passerelle par défaut**.

Définition 7 — Passerelle par défaut

La **passerelle par défaut** est un routeur vers lequel une machine envoie tous les paquets destinés à un autre réseau que le sien.

Exercice 6 — Communication locale ou distante

Indiquer si la communication est locale ou distante :

1. `192.168.1.62/26` vers `192.168.1.65/26`

   ---

2. `172.16.15.254/20` vers `172.16.16.1/20`

1.2 Représenter les réseaux

1.2.1 Un tas de routeurs

Du réseau local au réseau global

Dans la section précédente, nous avons vu qu'une machine peut déterminer si une destination est :

— dans le **même réseau local** (communication directe) ;

— dans un **réseau différent** (communication distante).

Dans le second cas, la machine ne peut pas envoyer directement ses paquets à la destination : elle ne connaît pas le chemin à travers Internet. Elle confie donc ces paquets à un équipement spécialisé : le **routeur**.

Définition 8 — Routeur

Un **routeur** est un équipement réseau relié à plusieurs réseaux. Il reçoit des paquets IP et décide vers quel réseau (ou quel autre routeur) les transmettre afin de les rapprocher de leur destination.

Définition 9 — Saut

On appelle **saut** (**hop**) le passage d'un paquet d'un routeur à un autre.
Le **nombre de sauts** entre deux routeurs est donc le nombre de routeurs intermédiaires traversés.

Décision locale

Un point très important : un routeur ne connaît pas « tout Internet ». Il ne calcule pas systématiquement le chemin complet.
À la place, il applique un principe simple : **à chaque paquet reçu, il choisit la prochaine étape (next hop).**
Chaque routeur prend donc une décision **locale**, et l'enchaînement de ces décisions produit un chemin global.

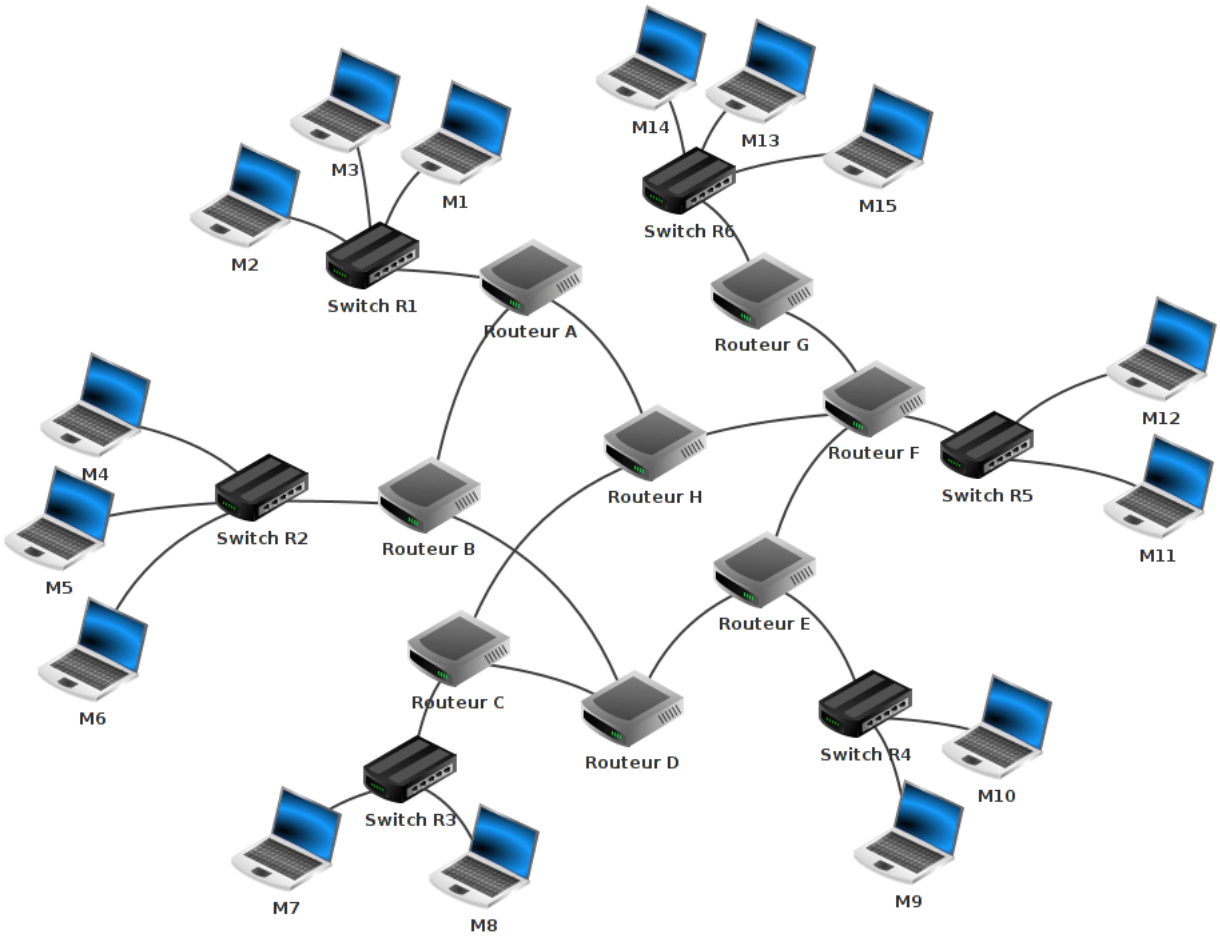


FIGURE 1 – Un exemple de réseau

Exercice 7 — Observer et raisonner sur un réseau

On considère le réseau représenté ci-dessus. Il est composé de :

- machines (M1 à M15) ;
- commutateurs (switches) ;
- routeurs (A à H).

1. Citer deux machines qui appartiennent au **même réseau local**. Justifier.

   ---

2. La machine M1 souhaite envoyer un message à la machine M3.
   - Cette communication est-elle locale ou distante ?
   - Par quel type d'équipement passe le message ?

3. La machine M1 souhaite maintenant envoyer un message à la machine M12.
   - Cette communication est-elle locale ou distante ?
   - Quel est le **premier routeur** traversé par le message ?

4. Donner un chemin possible permettant à un message d'aller de M1 à M12.

5. En comptant uniquement les **routeurs traversés**, combien de **sauts** ce chemin comporte-t-il ?

6. Existe-t-il un autre chemin possible entre M1 et M12 ? Si oui, le nombre de sauts est-il le même ? Quel est le plus court ?

7. Fort de votre expérience cette année, à quoi cela vous fait-il donc penser ?

1.2.2 Il s'agissait de graphes depuis le début !

Lien direct avec les graphes

On peut modéliser un ensemble de routeurs par un **graphe** :

- chaque routeur est un **sommet** ;
- chaque liaison entre deux routeurs est une **arête** (ou un arc).

Un trajet possible d'un paquet correspond alors à un **chemin** dans ce graphe.

Dans ce chapitre, on s'intéressera notamment à deux manières de choisir un chemin :

- **RIP** : minimiser le **nombre de sauts** (nombre d'arêtes empruntées) ;
- **OSPF** : minimiser un **coût** associé aux liaisons.

Nous reviendrons sur ces deux critères dans la suite.

Exercice 8

Traduisez la situation précédente par un graphe représentant les routeurs avec des cercles et les réseaux avec des rectangles :

1.3 La table de routage

Comment un routeur prend une décision

Lorsqu'un routeur reçoit un paquet IP, il ne transporte pas le **le chemin global à suivre**.
Son rôle est uniquement de décider :

À qui transmettre ce paquet maintenant, et par quelle interface ?

Cette décision est prise à l'aide d'une **table de routage**, qui contient l'ensemble des règles locales permettant d'acheminer les paquets.

Définition 10 — Table de routage

La **table de routage** est une table de règles qui associe, à chaque **réseau de destination**, la manière dont un paquet doit être transmis.
Chaque règle indique :

- un **réseau de destination** (sous la forme adresse + masque) ;
- soit une **interface de sortie** ;
- soit une **passerelle**, c'est-à-dire l'adresse IP d'un **routeur voisin**, atteinte via une interface.

Réseau directement connecté

Un réseau est dit **directement connecté** à un routeur lorsque le routeur possède une **interface IP appartenant à ce réseau**.
Dans ce cas :

- le routeur peut communiquer directement avec les machines du réseau ;
- aucune passerelle n'est nécessaire ;
- le paquet est envoyé directement sur l'interface correspondante.

Si le réseau de destination n'est pas directement connecté, le routeur doit transmettre le paquet à un **autre routeur**, appelé **passerelle**.

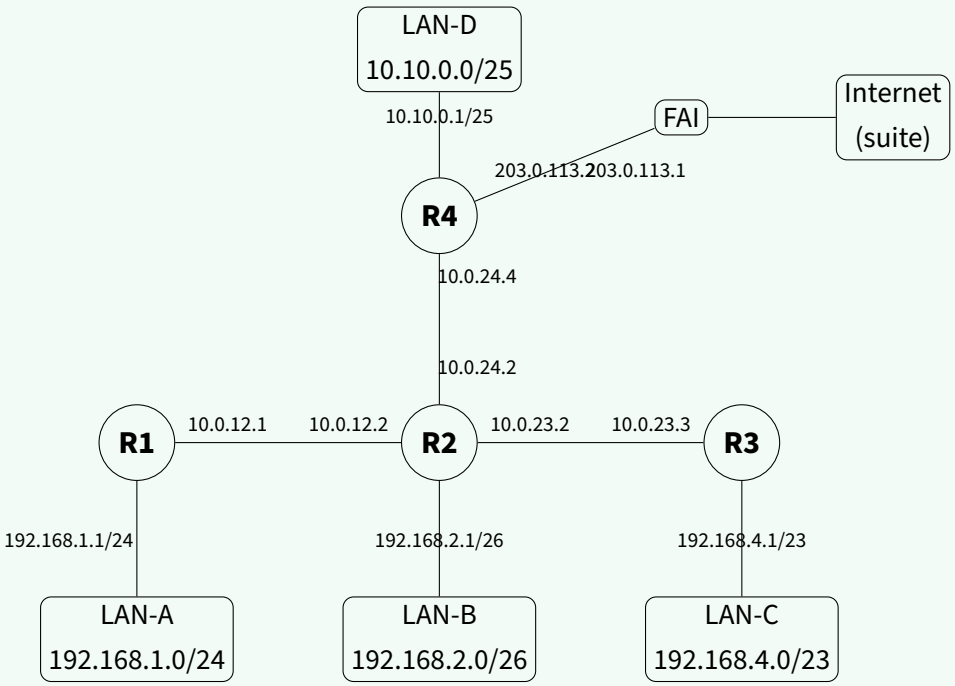
Définition 11 — Interface de sortie et passerelle

- L'**interface de sortie** indique **par où le paquet quitte physiquement le routeur** (ex. `eth0`, `eth1`, ou une adresse IP locale).
- La **passerelle** est l'adresse IP d'un **routeur voisin** vers lequel le paquet est transmis lorsque la destination n'est pas directement accessible.

Remarque 3. Il n'existe jamais de passerelle sans interface de sortie : un paquet est toujours envoyé sur une interface physique, qu'il soit transmis directement ou via un autre routeur.

Exemple 7 — Exemple de réseau et table de routage

Considérons le réseau suivant contenant 4 routeurs, (R1 à R4) et 4 réseaux locaux.



On donne maintenant la **table de routage du routeur** $R2$.

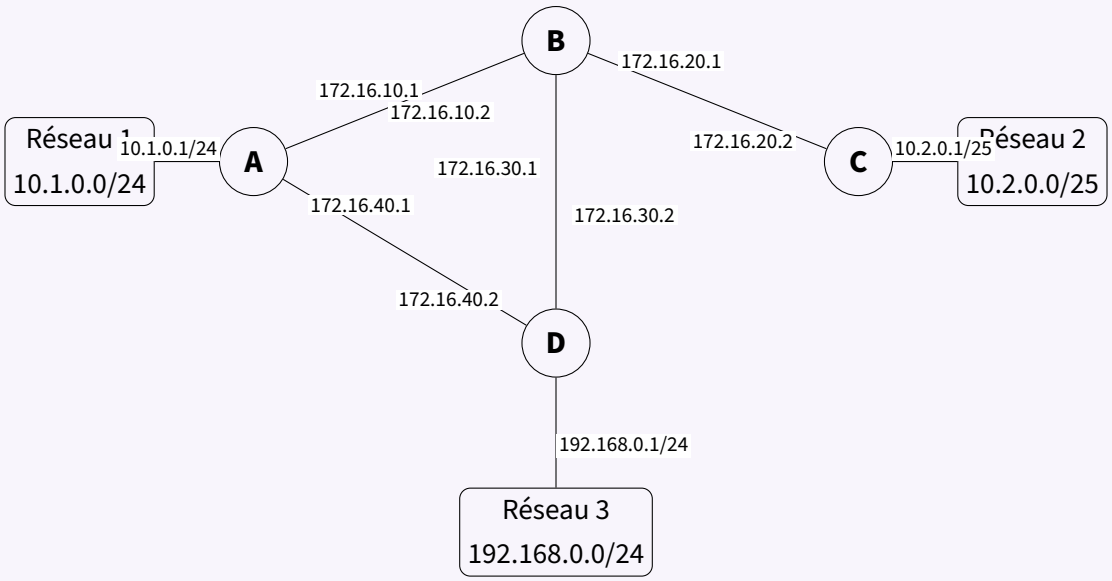
Réseau de destination	Passerelle	Interface de sortie
192.168.2.0/26 (LAN-B)	—	192.168.2.1/26
192.168.1.0/24 (LAN-A)	10.0.12.1 (R1)	10.0.12.2
192.168.4.0/23 (LAN-C)	10.0.23.3 (R3)	10.0.23.2
10.10.0.0/25 (LAN-D)	10.0.24.4 (R4)	10.0.24.2
0.0.0.0/0 (défaut)	10.0.24.4	10.0.24.2

Lecture :

- $R2$ reconnaît le réseau `LAN-B` comme directement connecté ;
- pour chaque autre réseau, il choisit un **routeur voisin** identifié par son **adresse IP** sur le lien inter-routeur ;
- la route par défaut pointe vers $R4$, qui donne accès au **FAI** puis à la **suite d'Internet**.

Exercice 9 — Routage

On considère le réseau suivant indiquant les interfaces IP sur chaque liaison.



On donne des **extraits de tables de routage** (destination, passerelle, interface de sortie). Les noms des réseaux sont rappelés entre parenthèses.

Table du routeur A :

Destination	Passerelle	Sortie
10.1.0.0/24 (Réseau 1)	—	10.1.0.1
10.2.0.0/25 (Réseau 2)	172.16.10.2	172.16.10.1
192.168.0.0/24 (Réseau 3)	172.16.40.2	172.16.40.1

Table partielle du routeur C :

Destination	Passerelle	Sortie
10.2.0.0/25 (Réseau 2)	—	10.2.0.1
192.168.0.0/24 (Réseau 3)	172.16.20.1	172.16.20.2

1. À quel routeur correspond l'adresse `172.16.40.2` ?

2. D'après la table de A, par où passe un paquet partant du `Réseau 1` vers le `Réseau 3`

3. Compléter la table de routage du routeur B pour les trois réseaux locaux (Réseau 1, Réseau 2, Réseau 3). Préciser passerelle et interface de sortie.

4. Expliquer le trajet d'un paquet allant d'une machine du `Réseau 2` vers une machine du `Réseau 3`.

---

5. On coupe la liaison entre A et D.
   (a) Le `Réseau 3` est-il encore atteignable depuis A ? Justifier.

   (b) Quelle(s) entrée(s) de table de routage doivent être modifiées (et sur quel(s) routeur(s)) pour que le trafic vers le `Réseau 3` continue à passer correctement ?

1.4 RIP : minimiser le nombre de sauts

Un problème majeur

Jusqu'ici, nous avons supposé que la table de routage était donnée.
Mais dans un réseau réel :

- des liaisons peuvent tomber en panne ;
- de nouveaux réseaux peuvent apparaître ;
- certains chemins peuvent devenir moins performants.

Mettre à jour manuellement les tables de routage est alors impossible à grande échelle.

C'est pour cette raison que des **protocoles de routage** comme **RIP** et **OSPF** ont été conçus : ils permettent aux routeurs de **construire automatiquement** leurs tables de routage.

Le protocole RIP

RIP (**Routing Information Protocol**) est un protocole très simple :

- chaque routeur discute **uniquement avec ses voisins** ;
- il échange régulièrement une liste du type : « **pour aller vers tel réseau, je pense que ça coûte** d **sauts** » ;
- il garde, pour chaque réseau, la route avec le **plus petit nombre de sauts**.

La mise à jour dans le protocle RIP

Pour chaque **réseau de destination** N, un routeur garde :

- une **distance** d (en nombre de sauts) ;
- un **voisin** (le **next hop**) vers lequel envoyer le paquet.

Quand un routeur reçoit de son voisin V : « **vers** N **je suis à distance** d_V », il se dit :

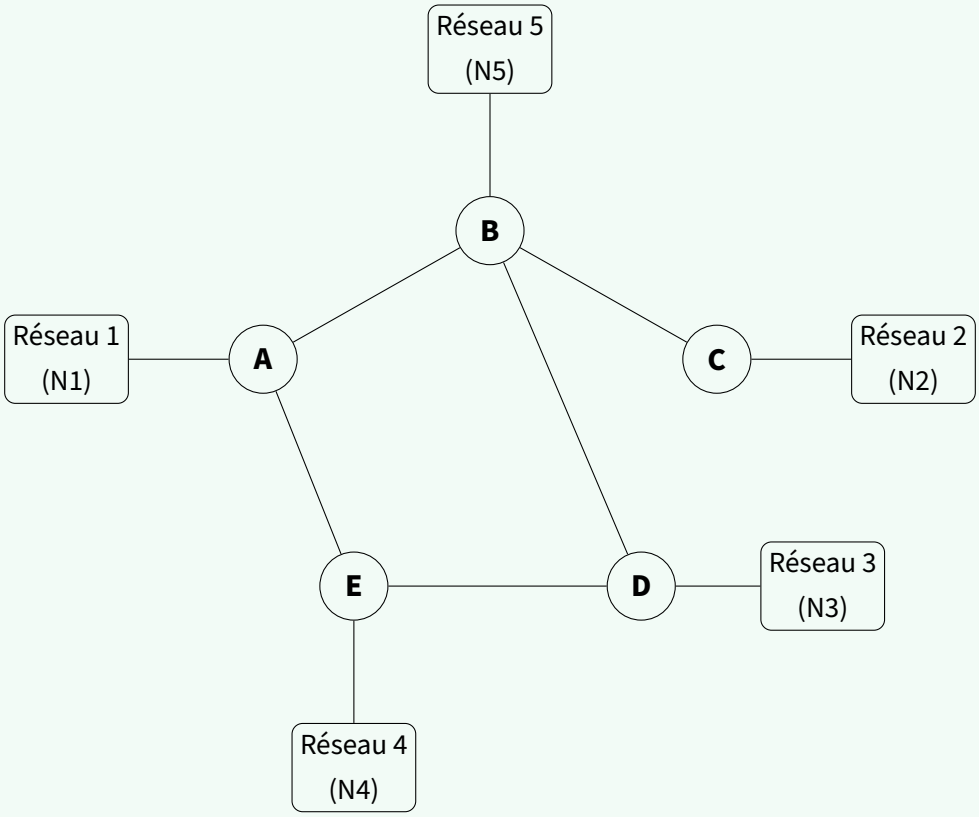
$$\text{si je passe par } V, \text{ alors vers } N \text{ je suis à } (d_V + 1).$$

Ensuite, il compare :

- s'il ne connaissait pas N : il ajoute la route ;
- s'il connaissait N mais avec une distance plus grande : il remplace par mieux ;
- si c'est plus mauvais : il ignore.

Exemple 8 — Construction des tables RIP : propagation pas à pas

On considère le réseau suivant (5 routeurs et 5 réseaux locaux).



Étape 0 — Initialisation (réseaux directement connectés).

A			B			C			D			E		
Dest	d	hop	Dest	d	hop	Dest	d	hop	Dest	d	hop	Dest	d	hop
N1	0	—	N5	0	—	N2	0	—	N3	0	—	N4	0	—

Étape 1 — Premier échang.

Chaque routeur envoie à chacun de ses voisins la liste de ce qu'il connaît à l'étape 0.

Après traitement de ces annonces, on obtient :

A			B			C			D			E		
Dest	d	hop	Dest	d	hop	Dest	d	hop	Dest	d	hop	Dest	d	hop
N1	0	—	N5	0	—	N2	0	—	N3	0	—	N4	0	—
N5	1	B	N1	1	A	N5	1	B	N5	1	B	N1	1	A
N4	1	E	N2	1	C				N4	1	E	N3	1	D
			N3	1	D									

Étape 2 — Deuxième échange.
À ce tour, chaque routeur envoie à ses voisins **tout ce qu'il connaît maintenant** (après l'étape 1).

Après traitement de ces annonces, on obtient :

A			B			C			D			E		
Dest	d	hop	Dest	d	hop	Dest	d	hop	Dest	d	hop	Dest	d	hop
N1	0	—	N5	0	—	N2	0	—	N3	0	—	N4	0	—
N5	1	B	N1	1	A	N5	1	B	N5	1	B	N1	1	A
N4	1	E	N2	1	C	N1	2	B	N4	1	E	N3	1	D
N2	2	B	N3	1	D	N3	2	B	N1	2	E	N5	2	A
N3	2	B	N4	2	A				N2	2	B			

Étape 3 — Troisième échange.
Les annonces du tour 3 améliorent uniquement les visions de E et C.

A			B			C			D			E		
Dest	d	hop	Dest	d	hop	Dest	d	hop	Dest	d	hop	Dest	d	hop
N1	0	—	N5	0	—	N2	0	—	N3	0	—	N4	0	—
N5	1	B	N1	1	A	N5	1	B	N5	1	B	N1	1	A
N4	1	E	N2	1	C	N1	2	B	N4	1	E	N3	1	D
N2	2	B	N3	1	D	N1	2	B	N1	2	B	N5	2	A
N3	2	B	N4	2	A	N4	3	B	N2	2	B	N2	3	A

Détection de panne

RIP n'est pas qu'un « calcul initial » : il doit aussi **réagir aux pannes**.

Idée :

- Les routeurs envoient des **mises à jour périodiques**.
- Si un routeur **n'entend plus parler** d'une route pendant un certain temps, il la considère **périmée** : il augmente sa distance, puis finit par **supprimer** l'entrée.

Pour accélérer la réaction, il existe souvent des **mises à jour déclenchées** : quand une route devient mauvaise (panne détectée), on en informe vite les voisins au lieu d'attendre la prochaine période.

Boucles et « compte à l'infini »

Le danger des protocoles à vecteur de distance comme RIP, c'est qu'en cas de panne, des routeurs peuvent se convaincre mutuellement qu'une route existe encore, et créer une **boucle**.

Scénario typique :

— C et B savaient atteindre un réseau N « quelque part plus loin ».

— Une panne coupe l'accès réel à N.

— B pense : « C doit encore savoir », et C pense : « B doit encore savoir ».

Résultat : B envoie vers C, C renvoie vers B, etc. Et pire : à chaque échange, la distance annoncée augmente petit à petit :

$$2,\ 3,\ 4,\ 5,\ \ldots$$

On appelle cela le **problème du compte à l'infini**.
Sans garde-fou, la distance pourrait croître sans fin. En RIP on fixe une borne : au-delà d'un certain seuil, on déclare que la route est **injoignable**.

Exercice 10 — Une panne dans notre réseau

On reprend le réseau de l'exemple (A, B, C, D, E et N1..N5).

1. Quel routeur est **le seul** directement connecté à N3 ?

   ---

2. On suppose que le routeur D tombe en panne (il disparaît du réseau).

   (a) Quel(s) réseau(x) local(aux) devient/deviennent injoignable(s) ?

   (b) Donner un exemple de **boucle possible** (entre quels routeurs ?) si l'information se propage mal.

1.5 OSPF : minimiser un coût avec Dijkstra

Pourquoi aller plus loin que RIP ?

Le protocole RIP a volontairement été présenté en premier car il est simple : il ne regarde que le **nombre de sauts**.

Mais dans un réseau réel :

- toutes les liaisons n'ont pas la même **qualité** ;
- certains liens sont rapides (fibre), d'autres lents ou saturés ;
- deux chemins avec le même nombre de sauts peuvent avoir des performances très différentes.

Le protocole **OSPF** (**Open Shortest Path First**) corrige ce problème en utilisant une idée plus riche :

chaque liaison a un coût, et on cherche le chemin de coût total minimal.

Modélisation par un graphe pondéré

Avec le protocole OSPF, on modélise le réseau par un **graphe pondéré** et le poids sur chaque arêtes peut dépendre de plusieurs critères :

- débit de la liaison ;
- latence ;
- qualité ou fiabilité ;
- choix de l'administrateur réseau.

Le problème à résoudre est alors exactement celui déjà rencontré dans le chapitre d'algorithmique des graphes :

trouver les plus courts chemins dans un graphe pondéré.

Définition 12 — Principe général d'OSPF

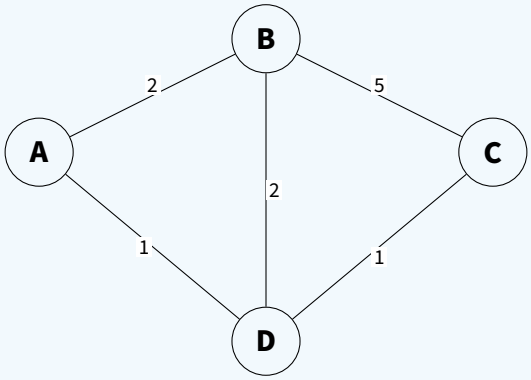
Le protocole OSPF fonctionne en trois grandes étapes :

1. Chaque routeur décrit ses **liaisons** et leurs **coûts**.
2. Ces informations sont **diffusées** à tous les routeurs de la zone OSPF.
3. Chaque routeur reconstruit le graphe et applique **Dijkstra**.

Ainsi, tous les routeurs possèdent la même carte du réseau, mais chacun calcule sa **table de routage locale**.

Exemple

Considérons le graphe pondéré suivant :



Depuis A, il existe plusieurs chemins vers C :

- $A \to B \to C$ de coût $2 + 5 = 7$;
- $A \to D \to C$ de coût $1 + 1 = 2$;
- $A \to B \to D \to C$ de coût $2 + 2 + 1 = 5$.

L'algorithme de **Dijkstra** sélectionne le chemin de coût minimal :

$$A \to D \to C.$$

OSPF fera exactement ce choix dans la table de routage de A.

Avantages et inconvénients d'OSPF

Avantages	Inconvénients
Chemins mieux adaptés aux performances réelles du réseau (coûts, débits, latence, etc.)	Protocole plus complexe à comprendre et à configurer que RIP
Convergence rapide en cas de panne ou de modification du réseau	Calculs plus coûteux (application de l'algorithme de Dijkstra)
Pas de problème de « compte à l'infini » ni de boucles persistantes	Besoin de plus de mémoire et de puissance de calcul sur les routeurs

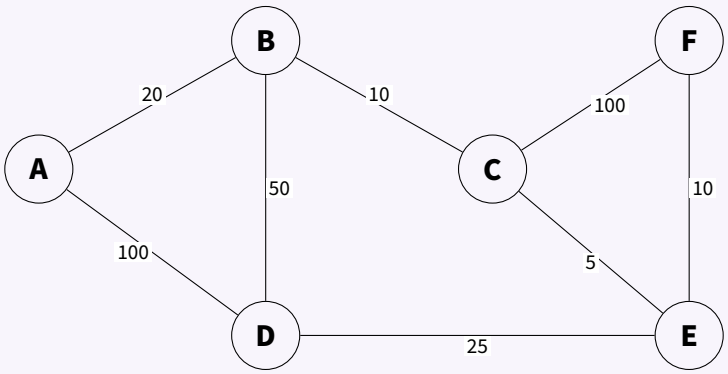
Exercice 11 — OSPF : calcul des coûts à partir des débits et choix des routes

On considère le réseau d'entreprise suivant, dont les routeurs sont modélisés par un graphe pondéré. Le protocole de routage utilisé est **OSPF**.

Dans ce sujet, le **coût OSPF** d'une liaison est calculé à partir de son **débit** par la formule :

$$\text{coût} = \frac{100}{\text{débit (en Mbit/s)}}$$

Le coût est ensuite **arrondi à l'entier le plus proche**.



Les nombres indiqués sur les liaisons sont les **débits** en Mbit/s.

1. Compléter le tableau suivant en calculant le **coût OSPF** de chaque liaison.

Liaison	Débit (Mbit/s)	Coût
A–B	20	....
A–D	100	....
B–C	10	....
B–D	50	....
C–F	100	....
D–E	25	....
E–F	10	....
C–E	5	....

2. Le routeur A doit envoyer des paquets vers le routeur F.
   (a) Quel est le **chemin de coût minimal** choisi par OSPF ?

(b) Quel est alors le **next hop** (premier routeur) dans la table de routage de A pour atteindre F ?

---

3. On suppose maintenant que la liaison $C--E$ est remplacée par une liaison de **140 Mbit/s**.

   (a) Calculer le nouveau coût de la liaison $C--E$.

   (b) Le chemin choisi par OSPF de A vers F change-t-il ?

   ---

4. On coupe la liaison $A--B$.

   (a) Le routeur A peut-il encore atteindre F ? Si oui, donner un plus court chemin.

   (b) Quel est le nouveau **next hop** de A pour atteindre F ?
