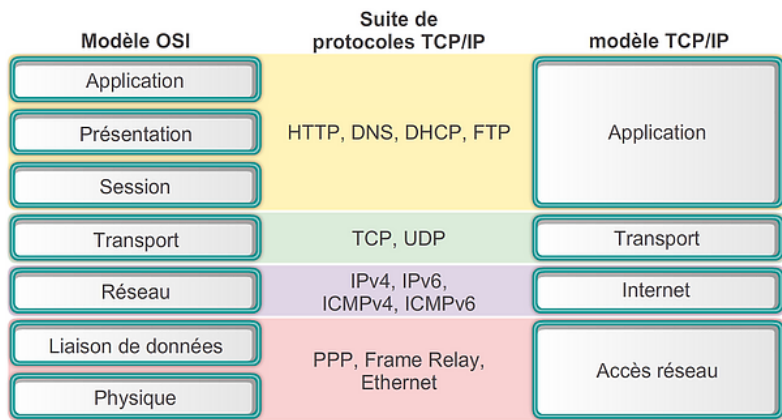


## 1 Quelques rappels

Un réseau informatique connecte plusieurs machines entre elles leur permettant de communiquer. Chaque machine est identifiée sur le réseau par son **adresse IP**. Le réseau internet, à l'échelle mondiale, connecte plusieurs réseaux locaux entre eux.

Les données échangées sont découpées en **paquets** qui transitent sur le réseaux, acheminés d'une machine source à une machine cible à travers des machines spécialisées : les **routeurs**.

Dans le modèle en 4 couches TCP/IP, le protocole TCP intervient au niveau de la couche Transport pour diviser les données en paquets et contrôler le bon acheminement de ces paquets. Au niveau de la couche Internet, le protocole IP s'occupe de l'acheminement des paquets, en s'appuyant sur les adresses IP.



extrait du site <https://linux-note.com/modele-osi-et-tcpip/>

## 2 Routage

Le **routage** des paquets consiste à trouver le « meilleur moyen » de traverser le réseau de la machine source à la machine cible.

En effet, lorsqu'un routeur reçoit un paquet, il doit le transférer à une autre machine. S'il est connecté à la machine cible, il lui communique directement le paquet (cas le plus simple!), mais s'il n'a pas de liaison directe avec la cible, il doit transmettre le paquet à un autre routeur (auquel il est lié) qui s'occupera de poursuivre l'acheminement.

Chaque routeur possède ainsi une **table de routage** qui lui indique en fonction de l'adresse de la machine cible (contenue dans l'en-tête du paquet IP), ou l'adresse de son réseau, vers quel routeur il doit envoyer le paquet.

Le réseau pouvant contenir un très grand nombre de routeurs, plusieurs routes différentes peuvent permettre d'atteindre la cible. Le routage doit donc s'appliquer à être efficace.

Par ailleurs, la topologie d'un réseau est dynamique : elle peut évoluer au cours du temps en fonction des pannes qui peuvent survenir ou de nouveaux routeurs intégrés au réseau par exemple.

Les tables de routage ne doivent donc pas être statiques et doivent s'adapter aux changements dynamiques de topologie du réseau.

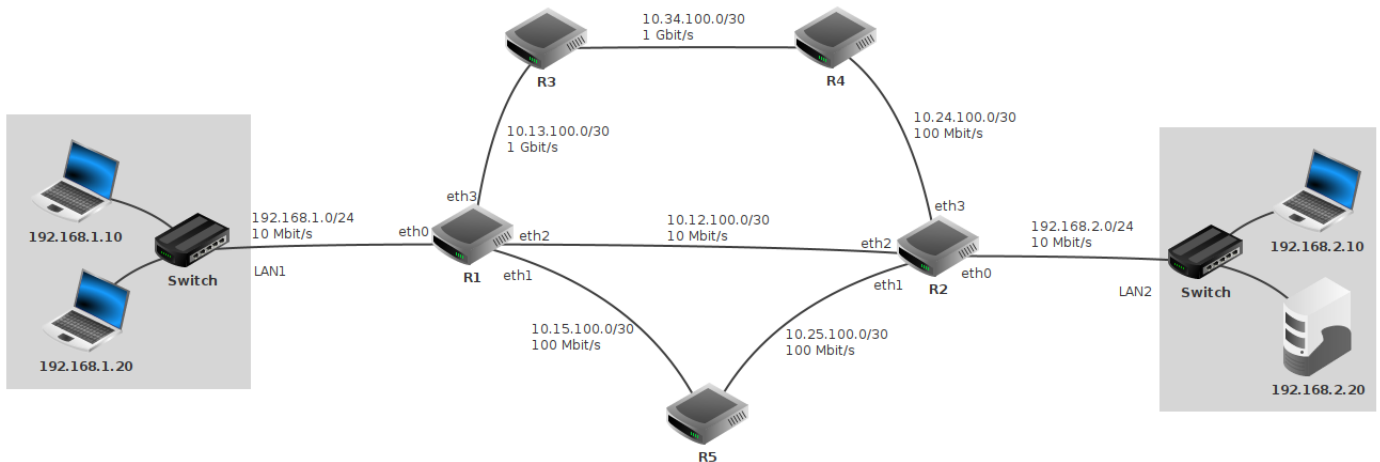
La complexité d'un réseau comprenant un grand nombre de routeurs rend impossible pour un administrateur réseau de centraliser la tâche consistant à connaître la topologie complète du réseau et d'écrire à la main les tables de routage pour chaque routeur en conséquence.

Il faut que le processus d'établissement des tables de routage soit **décentralisé** et que chaque routeur, **en communiquant uniquement avec ses voisins** puisse établir et mettre à jour sa propre table de routage.

### 3 Protocole à vecteurs de distance : RIP

Le protocole **RIP**, Routing information Protocole, repose sur le nombre de sauts (nombres de routeurs à traverser) pour atteindre la cible.

Nous appuierons notre cours sur le réseau suivant :



Dans cette configuration, la table de routage du routeur R1 possédera entre autres la ligne suivante :

cible	interface	passerelle	distance	commentaire
192.168.2.0/24	eth2	10.12.100.2 (R2)	2	vers LAN2

#### Explication :

Pour atteindre une machine du réseau 192.168.2.0/24 (LAN2), il faudra que le routeur utilise son interface réseau eth2 (un routeur a plusieurs interfaces réseaux, ou cartes réseaux, puisqu'il appartient à plusieurs réseaux en même temps), qu'il envoie le paquet à la passerelle d'adresse 10.12.100.2 (désignant le routeur R2 sur son interface eth2) et il y aura 2 sauts pour atteindre ce réseau cible (de R1 à R2, puis de R2 à 192.168.2.0/24).

Le protocole RIP repose sur les étapes suivantes :

1. Les tables de routage de chaque routeur sont initialisées avec l'information de liaison avec les voisins directs.

*Exemple* : La table de routage du routeur R1 est initialisée ainsi :

cible	interface	passerelle	distance	commentaire
192.168.1.0/24	eth0		1	vers LAN1
10.12.100.0/30	eth2		1	vers R2
10.13.100.0/30	eth3		1	vers R3
10.15.100.0/30	eth1		1	vers R5

2. Ensuite, les routeurs communiquent à leurs voisins **périodiquement** les informations de leur table de routage.

Les routeurs envoient des **vecteurs de distance** qui sont des **couples (adresse, distance)**.

*Exemple* : Lors de sa 1ère communication, le routeur R2 (juste après son initialisation) envoie entre autres le vecteur de distance (192.168.2.0/24, 1) au routeur R1. R1 apprend donc que R2 est à la distance 1 du réseau 192.168.2.0/24. Il peut donc ajouter une ligne à sa propre table de routage :

cible	interface	passerelle	distance	commentaire
192.168.2.0/24	eth2	10.12.100.2 (R2)	2	vers LAN2

Remarquez que la distance de R1 à 192.168.2.0/24 vaut 2 : R1 a incrémenté d'une unité la distance que R2 lui a communiquée, puisqu'il est lui-même à la distance 1 de R2.

L'algorithme de mise à jour des tables de routage est assez simple et repose sur l'algorithme de Bellman-Ford de recherche de plus court chemin. Lorsqu'un routeur reçoit une nouvelle route de la part d'un voisin, 4 cas sont envisageables :

- Il découvre une route vers un nouveau réseau inconnu : il l'ajoute à sa table.
- Il découvre une route vers un réseau connu, plus courte que celle qu'il possède dans sa table : il actualise sa table.
- Il découvre une route vers un réseau connu, plus longue que celle qu'il possède dans sa table : il ignore cette route.
- Il reçoit une route vers un réseau connu en provenance d'un routeur déjà existant dans sa table : il met à jour sa table car la topologie du réseau a été modifiée.

Par interrogations successives et de proche en proche, les tables de routage finissent par converger pour savoir vers quelle passerelle diriger un paquet quelle que soit sa destination.



On donne la table de routage complète après convergence du routeur R1. Écrire de même la table de routage de R2 (et même des autres routeurs pour les plus courageux).

cible	interface	passerelle	distance	commentaire
192.168.1.0/24	eth0		1	vers LAN1
192.168.2.0/24	eth2	10.12.100.2 (R2)	2	vers LAN2
10.12.100.0/30	eth2		1	vers R2
10.13.100.0/30	eth3		1	vers R3
10.15.100.0/30	eth1		1	vers R5
10.24.100.0/30	eth2	10.12.100.2 (R2)	2	
10.25.100.0/30	eth1	10.12.100.2 (R2)	2	
10.34.100.0/30	eth3	10.13.100.2 (R3)	2	

*Remarque* : il faut inventer des adresses pour les routeurs sur chacun de leur réseau, mais il faut rester cohérent sur l'ensemble des tables.

### Quelques remarques complémentaires :

- L'inconvénient majeur de ce protocole est la lenteur de convergence pour construire la mise à jour des tables lorsqu'un routeur tombe en panne par exemple.
- Le protocole RIP limite les routes à 15 sauts. Une distance de 16 correspond donc à une distance infinie et donc a une destination impossible à atteindre. Cela limite la taille des réseaux sur lesquels on peut appliquer ce protocole.
- Un routeur peut aussi détecter une panne d'un voisin s'il ne reçoit aucune information au bout d'un certain délai. Il communiquera donc à ses autres voisins une distance de 16 (infinie) vers ce routeur.
- Il y a des compléments dans le protocole pour éviter en particulier les boucles de routage (un paquet tournerait entre les routeurs sans jamais atteindre sa destination...).

## 4 Protocole OSPF à états de liens

Le protocole **OSPF**, Open Shortest Path First, ne se contente pas de compter le nombre de sauts jusqu'à la cible, mais il **prend en compte le débit des connexions entre deux machines afin d'optimiser le temps de transit des données** sur le réseau.

Le **coût** (ou poids) d'une liaison entre deux machines est lié au **débit** de la connexion entre les machines. Une liaison FastEthernet avec un débit de 100 Mbits/s a un coût plus faible qu'un câble Ethernet de débit 10 Mbits/s.

Par convention, le coût vaut :  $c = \frac{10^8}{d}$ , où  $d$  est le débit en bits/s.

Ainsi le coût d'une liaison FastEthernet est de 1 et celui d'un câble Ethernet standard est de 10.

Les informations communiquées entre routeurs dans le protocole OSPF sont des **états de liens**, c'est-à-dire une information sur les connexions d'un routeur avec ses voisins et le coût des liaisons.

Chaque routeur diffuse sur le réseau les états de liens de ses voisins **jusqu'à ce que l'ensemble des routeurs du réseau possède une vue globale des états de liens** du réseau.

Après que **chaque routeur connaît l'ensemble des états de liens** de tout le réseau, il peut mettre en œuvre *localement* un algorithme de recherche du plus court chemin pour atteindre une cible.

L'algorithme de calcul de route repose sur la recherche du plus court chemin dans un graphe pondéré selon l'algorithme de Dijkstra (voir chapitre P1.V : structure de données - graphes).

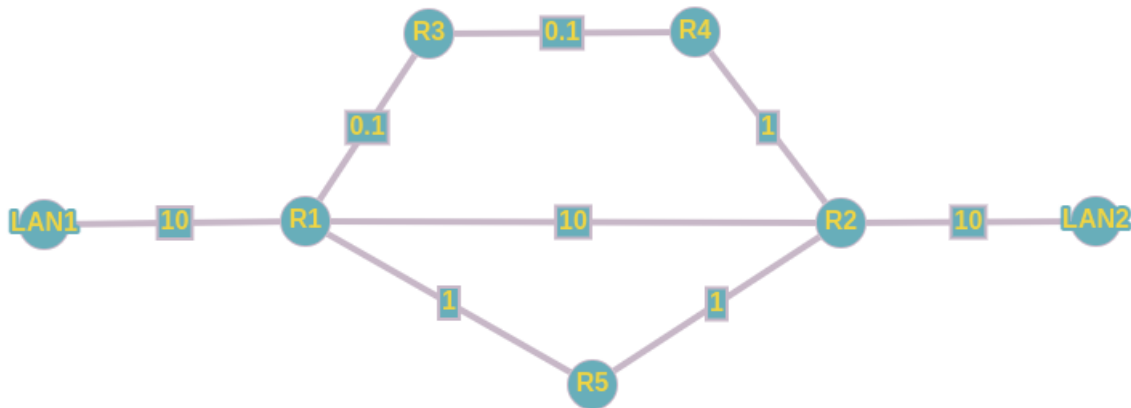
### Remarques :

- Pour de grands réseaux (comme internet), les routeurs sont répartis dans des **zones** de taille limitée, et ces zones sont reliées entre elles par des « routeurs frontières » (Area Border Router) qui constituent une structure nommée Backbone.
- Le protocole OSPF converge plus rapidement que le protocole RIP et ne se limite pas à des réseaux de petite dimension. En revanche, il sollicite beaucoup plus le CPU pour le calcul des routes.
- La communication des états de liens peut avoir lieu de façon privilégiée avec un routeur désigné parmi l'ensemble des routeurs de la zone. Ceci limite le trafic sur le réseau.

Avec le même réseau que précédemment, lorsque l'information a fini de converger, la base de données des états de liens de chaque routeur est la suivante :

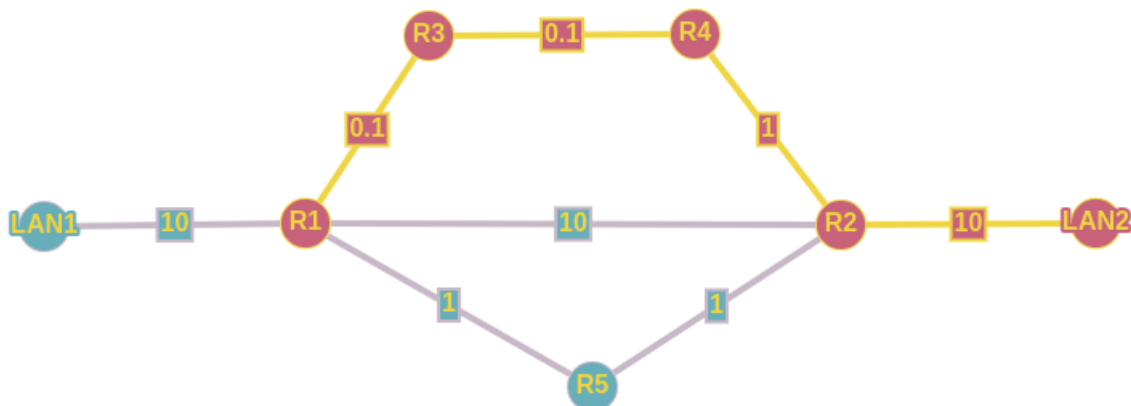
liaison	coût
R1 ↔ LAN1	10
R1 ↔ R2	10
R1 ↔ R3	0,1
R1 ↔ R5	1
R2 ↔ R4	1
R2 ↔ R5	1
R3 ↔ R4	0,1
R2 ↔ LAN2	10

La figure suivante présente alors ce réseau sous forme de graphe où les arêtes sont pondérées du coût de chacune des liaisons. Chaque routeur possède cette représentation du réseau.



Chaque routeur peut alors indépendamment calculer les routes les plus rapides vers n'importe quel point du réseau. Le routeur R1 calcule par exemple la route illustrée ci-après pour communiquer avec le LAN2.

Le chemin selon OSPF a un coût de 11,2 quand celui selon RIP aurait un coût de 20 (malgré 2 sauts de moins).



On donne la table de routage complète ci-dessous du routeur R1. Écrire de même la table de routage de R2.

cible	interface	passerelle	coût	commentaire
192.168.1.0/24	eth0		10	vers LAN1
192.168.2.0/24	eth3	10.13.100.2 (R3)	11,2	vers LAN2
10.12.100.0/30	eth2		10	
10.13.100.0/30	eth3		0,1	
10.15.100.0/30	eth1		1	
10.24.100.0/30	eth2	10.13.100.2 (R3)	1,2	
10.25.100.0/30	eth1	10.15.100.2 (R5)	2	
10.34.100.0/30	eth3	10.13.100.2 (R3)	0,2	

Vous pouvez contrôler vos résultat sur le site :

<http://graphonline.ru/fr/?graph=UjlriAjuBxSTKbvg>

La table de routage précédente peut être obtenue de façon systématique en appliquant l'algorithme de Dijkstra décrit par exemple dans cette vidéo :

<https://www.youtube.com/watch?v=MybdP4kice4>

*Remarque* : même si cet algorithme n'est pas explicitement au programme, il est bon de s'exercer à la mettre en œuvre.

R1	R2	R3	R4	R5	LAN1	LAN2
0 - R1						
<b>0 - R1</b>	10 - R1	0,1 - R1		1 - R1	10 - R1	
x		<b>0,1 - R1</b>	0,2 - R3			
x	1,2 - R4	x	<b>0,2 - R3</b>			
x	2 - R5	x	x	<b>1 - R1</b>		
x	<b>1,2 - R4</b>	x	x	x		11,2 - R2
x	x	x	x	x	<b>10 - R1</b>	
x	x	x	x	x	x	<b>11,2 - R2</b>

Le chemin complet s'obtient alors en remontant depuis la destination jusqu'au point de départ en passant par chaque routeur prédécesseur.

Le prédécesseur de LAN2 (la destination finale) est R2, qui a pour prédécesseur R4, qui a pour prédécesseur R3 qui a enfin comme prédécesseur R1 (le point de départ). Le chemin complet de R1 à LAN2 est donc : R1-R3-R4-R2-LAN2.