

Chapitre 14 - Le routage

Objectifs :

- ▷ Connaître les principaux protocoles de routage
- ▷ Identifier, suivant le protocole de routage utilisé (RIP ou OSPF), la route empruntée par un paquet.

1 Introduction

Exemple

Thomas veut envoyer une lettre de remerciement à son ami *Pacôme Toullemonde*, habitant à Liège (en Belgique), qui lui a prêté un manuel de NSI.

Il rédige donc un courrier, le place dans une enveloppe sur laquelle il inscrit le nom, le prénom, le numéro, la rue, le code postal, la ville et enfin le pays. Puis il dépose le tout à la poste de Saint-Dizier.

L'agent de la poste ne connaît pas *Pacôme*, mais il sait que la lettre doit aller en Belgique.

Une fois arrivée en Belgique, l'agent de poste Belge ne connaît toujours pas *Pacôme*, mais il sait qu'il doit envoyer la lettre vers un nouveau centre de tri, celui de Liège.

Arrivée au centre de Liège, la lettre va au bureau de poste correspondant au quartier de *Pacôme*. Et enfin elle sera remise au facteur qui distribue le courrier dans la rue de Pacôme, afin qu'il la dépose dans sa boîte aux lettres.

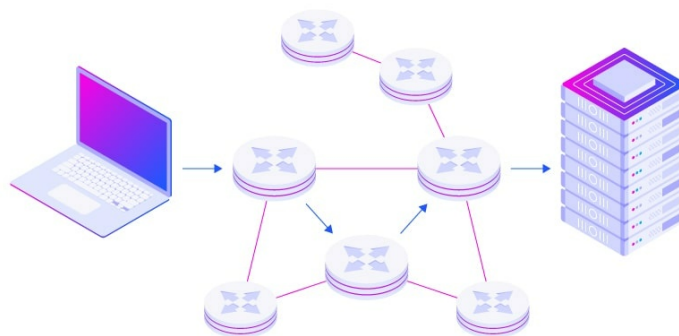
Nous pouvons ici faire une analogie entre les **réseaux postaux** et les **réseaux informatiques** :

- Thomas et son ami sont des **machines**.
- Les centres de tris par des **routeurs**.
- Le facteur est un **switch**.

2 Le principe

De la même façon, le rôle d'un réseau informatique consiste essentiellement à trouver les bons chemins pour amener chaque paquet à sa destination. Un paquet de données qui doit traverser un réseau n'a *a priori* aucune idée du chemin qu'il va devoir suivre. Il fait entièrement confiance aux indications fournies par les routeurs.

Chaque routeur va donc devoir orienter le paquet sur la base d'une et une seule information : l'adresse de la destination du paquet. On parle ici de **relayage**.



Au sein d'un réseau, le relayage consiste essentiellement à trouver la bonne sortie, celle qui mènera un paquet au plus proche possible de sa destination, et à trouver au sein du réseau le chemin qui mènera le paquet à cette sortie.

3 Définitions

A retenir !

Le **routing** est le processus qui permet de sélectionner un chemin dans un ou plusieurs réseaux pour transmettre des données depuis un expéditeur jusqu'à un ou plusieurs destinataires.

Le routage n'est pas exclusivement destiné aux réseaux informatiques tel qu'internet, il existe aussi dans d'autres domaines comme les réseaux de transports, les réseaux téléphoniques, ..., *etc.*

Par la suite nous nous intéresserons uniquement au routage au sein des réseaux informatiques.

4 Le routeur

La box que vous avez chez vous est également un routeur qui possède plusieurs interfaces réseau :

- ▷ une interface connectée au réseau de votre fournisseur d'accès à Internet, par exemple via la fibre.
- ▷ une interface filaire (Ethernet) connectée à votre réseau local
- ▷ une interface Wifi

Un routeur sur internet possède souvent plus de ports et ressemble d'extérieur à un switch. Il dispose d'un logiciel interne lui permettant de communiquer avec ses routeurs voisins pour l'aider à déterminer les meilleures routes à emprunter pour acheminer ses paquets.



Le routeur reçoit un paquet sur l'un de ses ports et sa mission consiste essentiellement à déterminer sur lequel de ses autres ports il doit ré-acheminer le paquet. Il doit le faire très vite et s'adapter à un environnement qui change (routes qui apparaissent et disparaissent).

5 La table de routage

Pour choisir le bon chemin, le routeur s'appuie sur une **table de routage** : c'est une table donnant pour chaque destination connue par le routeur la porte de sortie à emprunter ainsi que l'efficacité de cette route.

Cette efficacité sera exploitée dans les protocoles que nous détaillerons par la suite.

Voici un exemple :

Total number of IP routes: 687				
Destination	NetMask	Gateway	Port	Cost
137.194.2.0	255.255.254.0	137.194.4.254	v10	2
137.194.4.0	255.255.255.248	137.194.4.253	v10	2
137.194.4.8	255.255.255.248	137.194.4.251	v10	11
137.194.4.192	255.255.255.192	0.0.0.0	v10	1
137.194.6.0	255.255.254.0	137.194.4.254	v10	2
137.194.8.0	255.255.248.0	137.194.4.251	v10	20
137.194.16.0	255.255.255.128	137.194.160.230	v160	11
137.194.16.128	255.255.255.128	137.194.192.102	v192	11
137.194.16.144	255.255.255.240	137.194.192.102	v192	11
137.194.16.176	255.255.255.240	137.194.192.103	v192	20
137.194.17.0	255.255.255.0	137.194.192.103	v192	2

Par exemple, ici, on voit que pour joindre le réseau 137.194.2.0, il faut passer par le port v10, c'est-à-dire sur le lien de communication qui le relie au routeur dont l'adresse IP est 137.194.4.254.

Pour éviter que la table ne grossisse trop, on ne stocke pas l'adresse de chaque machine, mais des adresses de réseaux. Ici la dernière ligne de la table est valable pour toutes les machines dont les 24 premiers bits forment le préfixe 137.194.17.XXX, car le masque de sous-réseau est : 255.255.255.0.

A retenir !

Lorsqu'un routeur reçoit un paquet celui-ci comporte une adresse IP de destination. Pour savoir dans quelle direction envoyer le paquet, chaque routeur possède une table de routage.

Une table de routage est un tableau indiquant pour chaque réseau de destination (connu à partir de l'IP de la destination), quelle interface réseau on peut utiliser en lui associant également une métrique qui mesure la «distance» séparant le routeur du réseau de destination.

Pour afficher une table de routage :

- Sous *Windows* : `route print` ou `netsh int ipv4/ipv6 sh route`
- Sous *Unix/Mac Os* : `netstat -rn`
- Sous *Linux* : `ip -4/-6 route`

6 Types de routage

Il existe deux types de routage :

- ▷ Le **routage statique** qui consiste à indiquer l'adresse IP des réseaux que l'on cherche à atteindre. Les administrateurs vont configurer les routeurs un à un au sein du réseau afin d'y saisir les routes (Ils vont renseigner manuellement pour chaque adresse le nom de l'interface du routeur ou l'adresse IP du routeur voisin) à emprunter pour aller sur tel ou tel réseau. Chaque routeur ainsi paramétré permettra de faire le lien entre deux réseaux.

Si le réseau global est complexe, la configuration peut être fastidieuse et source d'erreurs.

De plus si le réseau évolue, il faudra le mettre à jour manuellement. Ce type de routage peut-être intéressant pour des raisons de sécurité si on veut maîtriser la route des paquets.

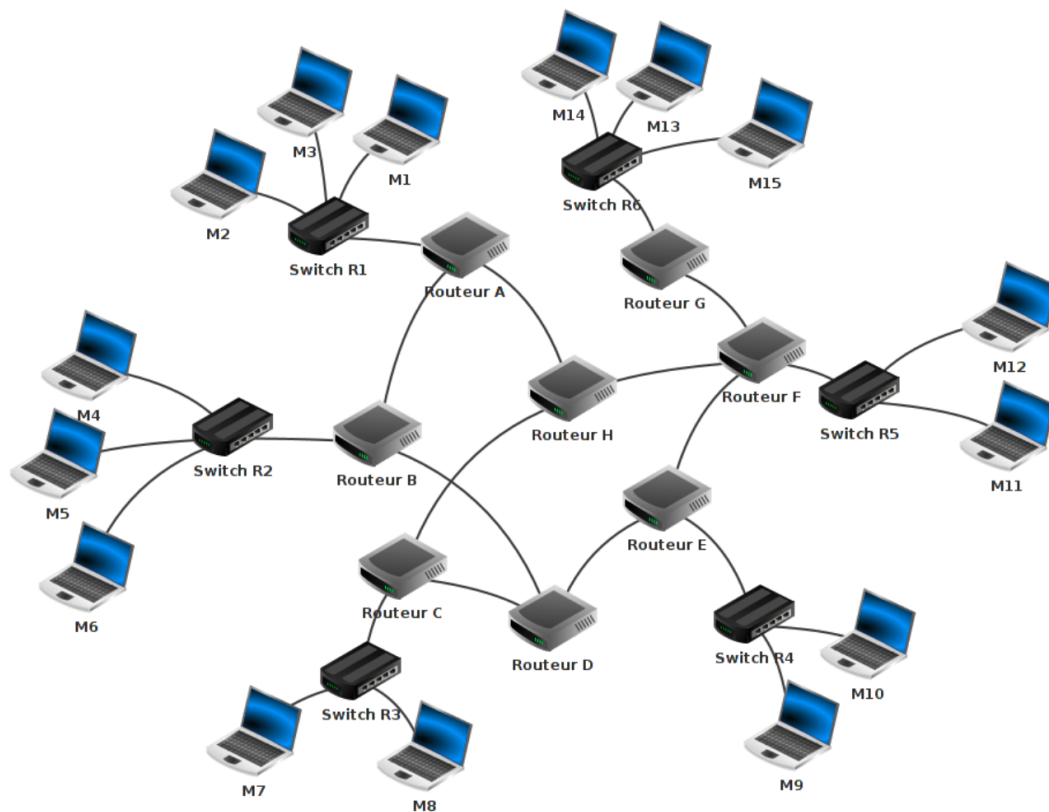
Ce type de routage reste très localisé.

- ▷ Le **routage dynamique** permet quant à lui de se mettre à jour de façon automatique.
- ▷ Un protocole de routage va permettre aux différents routeurs de se comprendre et d'échanger des informations de façon périodique ou événementielle afin que chaque routeur soit au courant des évolutions du réseau sans aucune intervention de l'administrateur du réseau.

7 Cas concret

Nous avons sur le schéma ci-dessous les éléments suivants :

- 15 ordinateurs : M1 à M15
- 6 switches : R1 à R6
- 8 routeurs : A, B, C, D, E, F, G et H.



Nous avons 6 réseaux locaux, chaque réseau local possède son propre switch.

Les ordinateurs M1, M2 et M3 appartiennent au réseau local 1.

Les ordinateurs M4, M5 et M6 appartiennent au réseau local 2.

Nous pouvons synthétiser tout cela comme suit :

Réseau local 1 : M1, M2 et M3	Réseau local 4 : M9 et M10
Réseau local 2 :	Réseau local 5 :
Réseau local 3 :	Réseau local 6 :

1. Compléter le tableau ci-dessus avec les réseaux locaux 2, 3, 5 et 6.

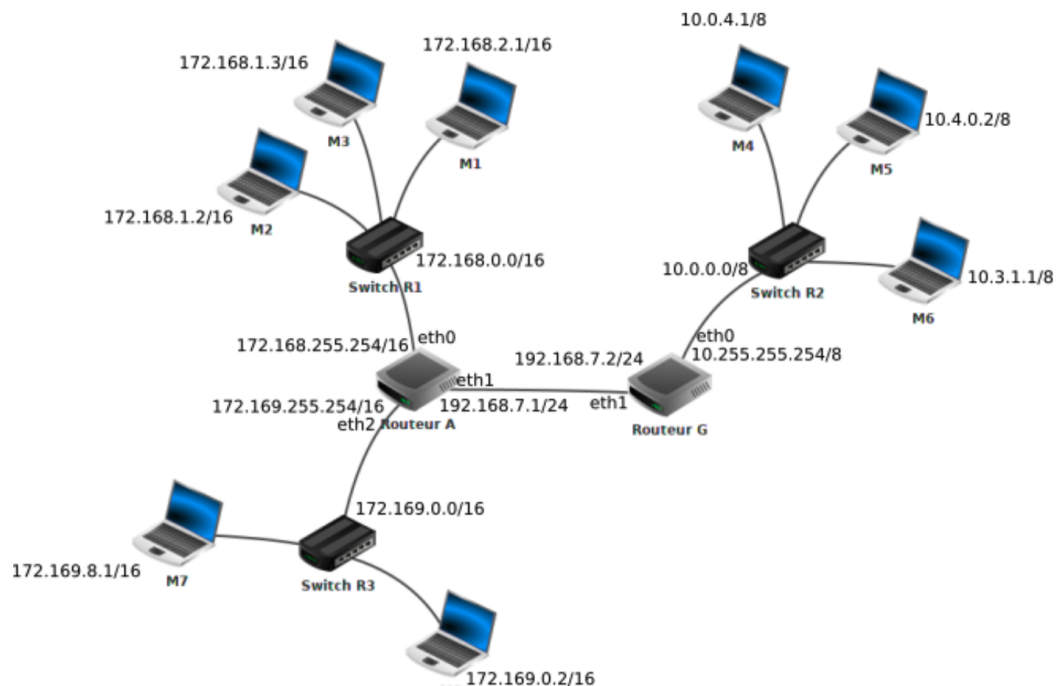
2. Observer les exemples de communications suivants :

- M1 veut communiquer avec M3
- M1 veut communiquer avec M6
- M1 veut communiquer avec M9
- M13 veut communiquer avec M9

3. Déterminer un chemin possible entre M4 et M14.

7.1 Comment les routeurs procèdent-ils pour amener les paquets à bon port ?

Soit le schéma suivant :



Dans le schéma ci-dessus M1 et M4 n'ont pas la même adresse réseau (car elles n'appartiennent pas au même réseau local).

Si M1 cherche à entrer en communication avec M4, le switch R1 va constater que M4 n'appartient pas au réseau local (grâce à son adresse IP), R1 va donc envoyer le paquet de données vers le **routeur A**.

Cela sera donc au **routeur A** de gérer le "problème".

Comment atteindre M4 ?

Chaque routeur possède une **table de routage**.

7.2 Comment fonctionne une table de routage ?

Étudier attentivement le schéma ci-dessus. Le choix des adresses IP des machines a été fait au "hasard". En revanche, vérifiez que tout est cohérent : adresses machines avec adresses réseaux (les adresses réseaux sont notées à côté des différents switch (par exemple le switch R1 est utilisé dans le réseau d'adresse 172.168.0.0/16)).

Voici les informations présentes dans la table de routage de A :

- le routeur A est directement relié au réseau 172.168.0.0/16 par l'intermédiaire de son interface **eth0**
- le routeur A est directement relié au réseau 172.169.0.0/16 par l'intermédiaire de son interface **eth2**
- le routeur A est directement relié au réseau 192.168.7.0/24 par l'intermédiaire de son interface **eth1** (le réseau 192.168.7.0/24 est un peu particulier car il est uniquement composé des **routeurs A et G**)
- le routeur A n'est pas directement relié au réseau 10.0.0.0/8 mais par contre il "sait" que les paquets à destination de ce réseau doivent être envoyés à la machine d'adresse IP 192.168.7.2/24 (c'est à dire le **routeur G** qui lui est directement relié au réseau 10.0.0.0/8)

On peut résumer tout cela avec le tableau suivant (table de routage simplifiée de A)

Réseau	Passerelle	Interface	Nombre de sauts *
172.168.0.0/16		eth0	0
192.168.7.0/24		eth1	0
172.169.0.0/16		eth2	0
10.0.0.0/8	192.168.7.2/24	eth1	

* Le nombre de saut permet de trouver le chemin le plus court : plus la valeur est petite plus le chemin est court. Un réseau directement lié à un routeur aura un nombre de saut de 0.

7.3 Exercice d'application

Déterminer la table de routage de G.

Réseau	Passerelle	Interface	Nombre de sauts

7.4 En résumé

Plusieurs algorithmes sont possibles pour compléter les tables de routage.

Pour le routage réseau, des contraintes particulières liées au fonctionnement des réseaux sont à prendre en compte :

- ▷ Les routeurs n'ont pas connaissance de la topologie globale du réseau, **ils ne communiquent qu'avec leurs voisins immédiats**.
- ▷ L'algorithme est **distribué** : il n'y a pas de centre de gestion central. Chaque routeur se constitue sa propre table à partir des informations communiquées par ses voisins, puis il transmet à ses voisins les informations en sa possession.
- ▷ L'algorithme est **itératif** : il est exécuté en permanence et ne s'arrête jamais.

Lorsqu'une modification est faite sur le réseau, celle-ci se propage de proche en proche à chaque routeur qui adapte alors sa table de routage.

A retenir !

A partir de là, deux grandes familles de protocole de routage se dégagent :

- ▷ le routage **à vecteur de distance** : il consiste à compter le **nombre d'étapes nécessaires** pour atteindre l'objectif.
- ▷ le routage **à état de lien** : il consiste à prendre en compte **la qualité des liaisons** pour optimiser le débit ou la latence de la liaison.

8 Le protocole RIP

8.1 Principe général de l'algorithme

A retenir !

Le protocole RIP (*Routing Information Protocol*) rentre dans la catégorie des protocoles à vecteur de distance. Un vecteur de distance est un couple (adresse, distance).

Le principe simplifié de ce protocole est de **chercher à minimiser le nombre de routeurs à traverser pour atteindre la destination** (on minimise le nombre de sauts).

Il est utilisé pour des réseaux de petite taille (jusqu'à 15 machines au maximum).

Chaque routeur reçoit en permanence de ses voisins les informations de routage qu'ils possèdent. Il va alors exploiter ces informations pour se construire lui-même sa table de routage en ne retenant que les informations les plus pertinentes : une simple comparaison permet de ne garder que le chemin le plus avantageux. Il transmettra à son tour ces informations à ses voisins et ainsi de suite.

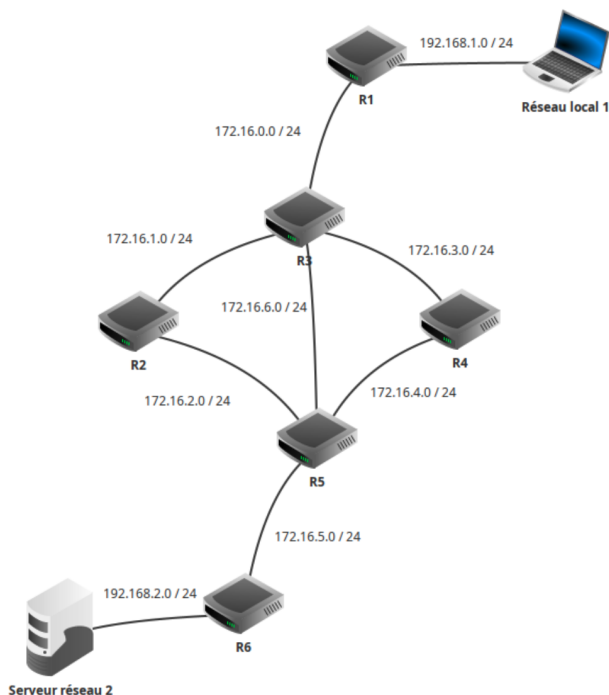
L'algorithme de *Bellman-Ford* est un algorithme de recherche de plus court chemin dans un graphe.

A l'issue de quelques étapes, les tables se stabilisent et le routage est pleinement opérationnel. Le temps nécessaire à la stabilisation des tables est proportionnel au diamètre du graphe modélisant le réseau (c'est-à-dire au nombre maximal d'étapes nécessaires pour relier deux points quelconques du réseau).

Regarder la vidéo suivante sur **le principe du routage à vecteur de distance**.

8.2 Exemple

Considérons le réseau suivant qui relie deux réseaux d’une entreprise :



Le “réseau local 1” contient des postes de travail dans un bureau.

Le “serveur réseau 2” contient un serveur dans un centre de données.

Les routeurs R1 et R6 permettent d’accéder au réseau de l’entreprise.

R2, R3, R4 et R5 sont des routeurs internes au réseau.

Nous allons nous intéresser à l’évolution des tables de routage des routeurs R1 et R3 sur lesquelles on a activé le protocole RIP.

8.2.1 Etape 0

- Au démarrage, les routeurs R1 et R3 ne connaissent que leurs voisins proches.
- Au départ, R1 ne peut atteindre que ses voisins immédiats (nb Sauts vaut 1).
- Aucune passerelle n’est nécessaire puisque la communication est directe.
- Chaque sous réseau utilise une interface spécifique. Le *réseau local 1* contenant les postes de travail est accessible en wifi.

Voici à quoi peut ressembler sa table de routage au démarrage :

Destination	Passerelle	Interface	Nb sauts	Remarques
192.168.1.0		wifi	1	⇒ vers les postes de travail
172.16.0.0		eth0-1	1	⇒ vers R3

Pour ce qui concerne R3, celui-ci possède 4 interfaces réseau filaires, que nous nommerons **eth0-3** qui permettent d’atteindre les routeurs immédiats (R1, R2, R4 et R5).

Voici à quoi peut ressembler sa table de routage au démarrage :

Destination	Passerelle	Interface	Nb sauts	Remarques
172.16.0.0		eth0-3	1	
172.16.1.0		eth1-3	1	
172.16.6.0		eth2-3	1	
172.16.3.0		eth3-3	1	

8.2.2 Etape 1

Au bout de 30 secondes, un premier échange intervient avec les voisins immédiats des routeurs.

Le principe de l'algorithme

Lorsqu'un routeur reçoit une nouvelle route de la part d'un voisin, 4 cas sont envisageables :

1. Il découvre une route vers un nouveau réseau inconnu.
⇒ **Il l'ajoute à sa table.**
2. Il découvre une route vers un réseau connu, plus courte que celle qu'il possède dans sa table.
⇒ **Il actualise sa table.**
3. Il découvre une route vers un réseau connu, plus longue que celle qu'il possède dans sa table.
⇒ **Il ignore cette route.**
4. 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.**

En appliquant ces règles, voici la table de routage de R1 après une étape :

Destination	Passerelle	Interface	Nb sauts	Remarques
192.168.1.0		wifi0	1	⇒ vers les postes de travail
172.16.0.0		eth0-1	1	⇒ vers R3
172.16.1.0	172.16.0.3	eth0-1	2	Ces 3 routes proviennent de R3
172.16.6.0	172.16.0.3	eth0-1	2	Ces 3 routes proviennent de R3
172.16.3.0	172.16.0.3	eth0-1	2	Ces 3 routes proviennent de R3

172.16.0.3 est l'adresse IP du routeur R3.

- On ajoute à la table précédente les réseaux atteignables par R3.
- On pense cependant à ajouter 1 au nombre de sauts !
- Si R1 veut atteindre le réseau 172.16.3.0, il s'adressera à R3 et atteindra le réseau cible en 2 sauts

Voici la table de R3 qui s'enrichit des informations envoyées par R1 afin d'atteindre le réseau local, mais aussi des informations en provenance de R2, R4 et R5.

Il découvre ainsi 4 nouveaux réseaux.

Destination	Passerelle	Interface	Nb sauts	Remarques
172.16.0.0		eth0-3	1	
172.16.1.0		eth1-3	1	
172.16.6.0		eth2-3	1	
172.16.3.0		eth3-3	1	
192.168.1.0	172.16.0.1	eth0-3	2	Reçu de R1
172.16.2.0	172.16.1.2	eth1-3	2	Reçu de R2
172.16.5.0	172.16.6.5	eth2-3	2	Reçu de R5
172.16.4.0	172.16.3.4	eth3-3	2	Reçu de R4

8.2.3 Etape 3

Comme vous le voyez, les tables deviennent vite longues et énumérer dans le détail chacune d'elle est trop long.

On va donc passer directement à l'étape finale : l'étape 3.

Voici ce que contient la table de routage de R1 :

Destination	Passerelle	Interface	Nb sauts	Remarques
192.168.1.0		wifi0	1	⇒ vers les postes de travail
172.16.0.0		eth0-1	1	⇒ vers R3
172.16.1.0	172.16.0.3	eth0-1	2	
172.16.6.0	172.16.0.3	eth0-1	2	
172.16.3.0	172.16.0.3	eth0-1	2	
172.16.5.0	172.16.0.3	eth0-1	3	Obtenu à l'étape 2
192.168.2.0	172.16.0.3	eth0-1	4	Obtenu à l'étape 3

Comme vous le voyez, le routeur R1 est à présent en capacité d'acheminer un paquet du poste de travail du réseau 1 vers le serveur se trouvant dans le réseau 2.

8.3 Détection des pannes

Le protocole RIP est en mesure de détecter des pannes :

Si un routeur ne reçoit pas d'information de la part d'un de ses voisins au bout d'un temps de l'ordre de 3 minutes (configurable), il va considérer que ce lien est mort et en informer ses voisins en indiquant un nombre de sauts égal à 16.

Puisque RIP ne gère que 15 sauts au maximum, 16 peut être considéré comme une distance infinie. De cette manière, les voisins vont pouvoir recalculer leurs routes en conséquence en évitant le lien qui est tombé.

8.4 Détection des boucles

RIP implémente d'autres mécanismes pour empêcher que se forment des boucles de routage.

Une boucle est par exemple une route du type $R2 \Rightarrow R3 \Rightarrow R5 \Rightarrow R2$.

Des exemples de tels mécanismes sont :

- ▷ une durée de vie limitée sur les paquets (TTL) afin qu'un paquet qui tourne en rond soit détruit
- ▷ ne pas renvoyer une information vers un routeur si celle-ci est déjà passée par ce routeur

Exercice 1 : *A vous de jouer !*

Élaborer au fil du temps la table de routage du routeur R4 de manière similaire à ce que l'on vient de faire.

Destination	Passerelle	Interface	Nb sauts	Remarques

8.5 Conclusion

Le protocole RIP est en général utilisé sur de **petits réseaux** : il est en effet limité à **15 sauts** et de plus, il génère un trafic important.

Pour des structures plus importantes, on va lui préférer le *protocole OSPF*.

9 Le protocole OSPF

Le **protocole OSPF** (*Open Shortest Path First*) est dans la catégorie des protocoles **à état de lien**.

Dans le protocole à vecteur de distance que nous venons de voir, on cherche à minimiser le nombre de sauts, mais sans aucune garantie que le chemin emprunté soit en réalité le plus performant (en termes de débit par exemple).

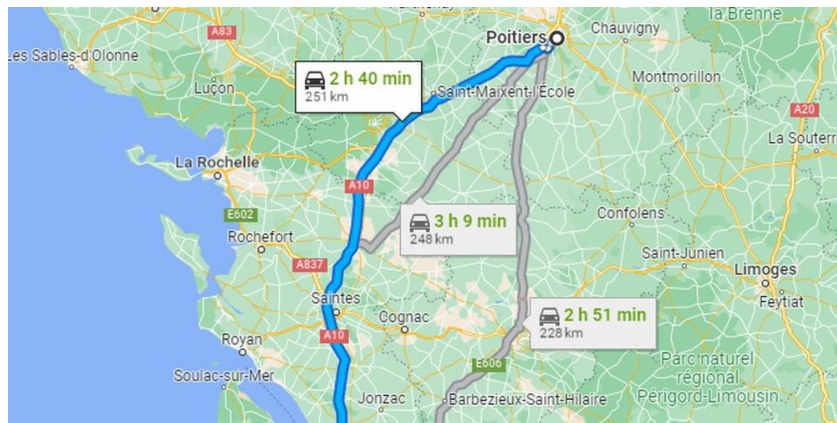
9.1 Principe général de l'algorithme

A retenir !

Le *protocole OSPF* propose une approche tout à fait différente : au lieu de s'intéresser au nombre de sauts, on va chercher à **optimiser le débit des liaisons empruntées**.

Pour cela, chaque routeur va devoir **connaître l'intégralité du réseau avec le débit associé à chaque lien** afin d'appliquer un algorithme de recherche de chemin optimal.

On peut faire un parallèle entre le fonctionnement d'OSPF et celui de nos logiciels de guidage par GPS.



En effet, dans ce type de logiciels :

- ▷ l'ensemble de la carte de France et de ses routes est connue du logiciel
- ▷ le type de chaque route est renseigné ainsi que la vitesse autorisée sur la route
- ▷ le calcul d'itinéraire va permettre le calcul d'un chemin permettant par exemple d'emprunter les routes sur lesquelles la vitesse est la plus importante (temps le plus court).

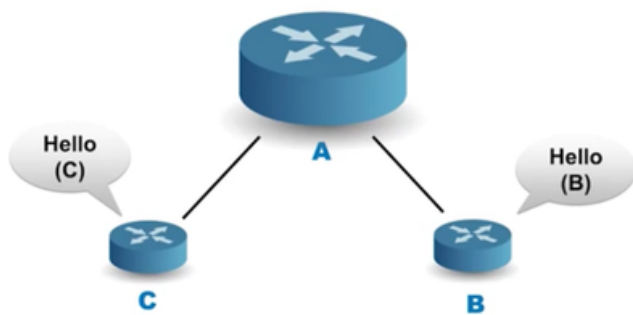
Visionner la vidéo suivante sur [le principe du routage à état de lien](#).

9.2 Découverte de la topologie du réseau

Le protocole OSPF a besoin de connaître la topologie du réseau ainsi que la qualité de chaque lien en terme de bande passante.

Pour cela, chaque routeur va fabriquer une **table de voisinage** : il s'agit d'un tableau permettant d'identifier tous les routeurs qui lui sont connectés ainsi que le débit associé à chaque lien.

Pour obtenir ces informations, le routeur échange périodiquement des messages (appelés messages *hello*) avec ses voisins.



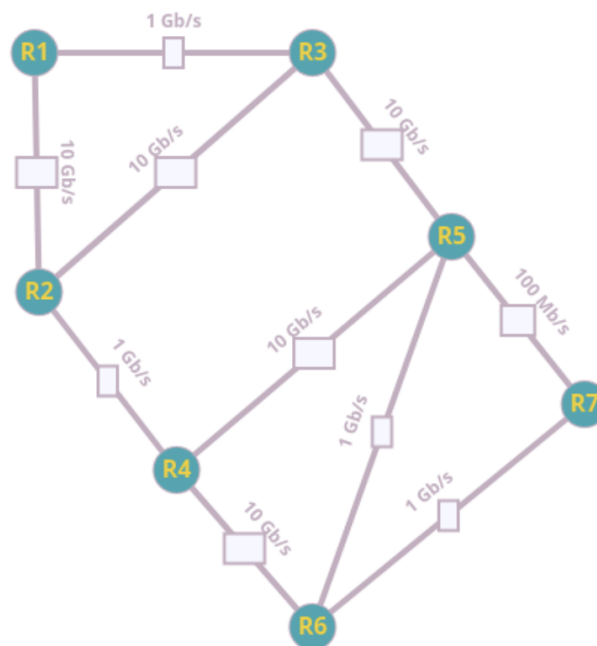
Voisin	Qualité du lien
B	1 Gb/s
C	10 Gb/s

Une fois tous ses voisins directs identifiés, le routeur va envoyer sa table de voisinage à tous les autres routeurs du réseau. Il va recevoir des autres routeurs leurs tables de voisinages et ainsi pouvoir constituer une cartographie complète du réseau.

Nous allons utiliser l'**algorithme de Dijkstra** datant de 1959 qui permet de trouver **le chemin le plus court sur un graphe**.

9.3 Exemple d'application

Considérons le réseau suivant. Après échanges de messages *hello*, la cartographie suivante du réseau a été constituée :



Nous cherchons à déterminer le chemin le plus rapide entre R1 et R7. L'outil graphonline vous permet de le faire visuellement via le menu Algorithmes / plus court chemin avec l'algorithme de Dijkstra.

Contrairement à RIP, le chemin qu'OSPF nous indiquera sera $R1 \Rightarrow R2 \Rightarrow R3 \Rightarrow R5 \Rightarrow R4 \Rightarrow R6 \Rightarrow R7$.

Ce chemin n'est clairement pas le plus efficace en termes de sauts mais c'est le plus rapide en terme de débit car il n'exploite pratiquement que des liaisons à 10 Gb/s.

9.4 Notion de coût

Video - algorithme de Dijkstra

L'algorithme de Dijkstra permet de minimiser la longueur d'un chemin, or nous souhaitons maximiser le débit sur nos liaisons. Nous allons donc considérer l'inverse de la bande passante de nos liens pour appliquer Dijkstra : maximiser le débit revient à minimiser l'inverse des débits.

La métrique utilisée n'est donc plus le nombre de sauts, mais le **coût de la liaison**. Celui-ci tient compte du **débit binaire de chaque liaison en bits par seconde**.

On définit le coût par l'expression suivante :

$$\text{coût} = \frac{10^8}{\text{débit en Mbit/s}}$$

- ▷ 10 Gb/s sera affecté au poids 0.1
- ▷ 1 Gb/s au poids 1
- ▷ 100 Mb/s = 0,1 Gb/s au poids 10

9.5 Application de l'algorithme

Nous allons ensuite constituer notre tableau. A chaque nouvelle ligne :

- ▷ on calcule les distances totales vers les destinations possibles
- ▷ on ne retient que la plus petite (en gras) que l'on marque sur une nouvelle ligne
- ▷ pour empêcher les retours, une fois une destination choisie (en gras), on désactive tout le reste de la colonne (avec des ×).

Tableau pour la liaison de R1 à R7 :

R1	R2	R3	R4	R5	R6	R7

Dans le tableau, on indique des **couples distance - origine** :

Ainsi 0.5 - R4 dans la colonne R6 signifie que R6 est à une distance minimum de 0.5 du départ en provenance de R4.

On peut ainsi reconstituer l'itinéraire optimal en partant de R7 et en remontant à l'envers en utilisant le champ origine :

R1 ⇒ R2 ⇒ R3 ⇒ R5 ⇒ R4 ⇒ R6 ⇒ R7 avec un poids total minimum de 1.5.

Exercice 2 : A vous de jouer !

On supprime la liaison R4-R5. Appliquer l'algorithme de Dijkstra pour déterminer un chemin optimal entre R1 et R7.

R1	R2	R3	R4	R5	R6	R7

9.6 Conclusion

OSPF peut s'adapter à la qualité des liens mais dans une certaine mesure uniquement : si un lien à 10Gb/s est saturé, il vaut mieux emprunter un lien à 1Gb/s sous utilisé, mais OSPF n'en a pas connaissance.

Il n'y a pas dans l'absolu de meilleur algorithme de routage, tout dépend du réseau auquel on a affaire. Un protocole sera plus réactif face aux changements de topologie mais au prix d'un plus gros volume échangé. Un autre sera plus efficace si les liaisons au sein du réseau sont très hétérogènes.

Regarder la [vidéo bilan sur les protocoles RIP et OSPF](#).

10 Exercices

Exercice 3 : *IPv4 ou IPv6*

1. Combien y a-t-il d'adresses IP disponibles avec le protocole *IPv4* ?
2. Combien y a-t-il de personnes connectées à Internet en 2023 ?
3. Comment est codée une adresse IP selon le nouveau protocole *IPv6* ?
4. Combien y a-t-il d'adresses IP disponibles avec le protocole *IPv6* ?
5. Depuis quand fonctionne le protocole *IPv6* ?

Exercice 4 : *Adresse MAC*

1. Ouvrir un terminal (Taper `cmd` ou `powershell` dans la barre de recherche *Windows*).
2. Trouver l'adresse MAC de votre carte réseau (avec la commande `ipconfig /all`).
3. A l'aide de la commande `ping` suivie de l'adresse Internet du site, trouver l'adresse IP des sites `www.twitter.com` et `www.google.fr`. Y a-t-il une différence ?
4. A l'aide de la commande `tracert`, suivie de l'adresse IP du site, trouver les routeurs rencontrés pour accéder au site `www.twitter.com`.

Exercice 5 : *QCM de révision*

Répondre aux questions du QCM sur le [site suivant](#).

Exercice 6 : *Exercices d'entraînement*

Faire les [exercices en ligne suivants](#).