

Protocoles à vecteur de distance : RIP

1. Pourquoi automatiser le routage.

L1 : Introduction - le routage interne



2. Le routage à vecteur de distance : le protocole RIP

D'après https://www.lecluse.fr/nsi/NSI_T/archi/routage/RIP/.

2.1. Le principe

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), et ce protocole cherche à minimiser le nombre de routeurs (c'est-à-dire la distance) à traverser pour atteindre la destination (on cherche ici à minimiser le nombre de sauts).

L2 : le routage à vecteur de distance



Chaque routeur reçoit en permanence (toutes les 30 secondes environ) de ses voisins les informations de routage qu'ils possèdent. Il va alors exploiter ces informations pour 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. C'est l'algorithme de [Bellman-Ford](#), un des algorithmes de recherche de plus court chemin dans un graphe, que nous étudierons plus en détail dans le chapitre sur les graphes.

A l'issue de quelques étapes, les tables de tous les routeurs se stabilisent et le routage est pleinement opérationnel. Le temps nécessaire à la stabilisation des tables est proportionnel au diamètre du réseau, c'est-à-dire à la plus grande route possible entre

deux routeurs - c'est ce qui est appelé **vitesse de convergence** dans la vidéo de l'[Institut Mines-Télécom](#).

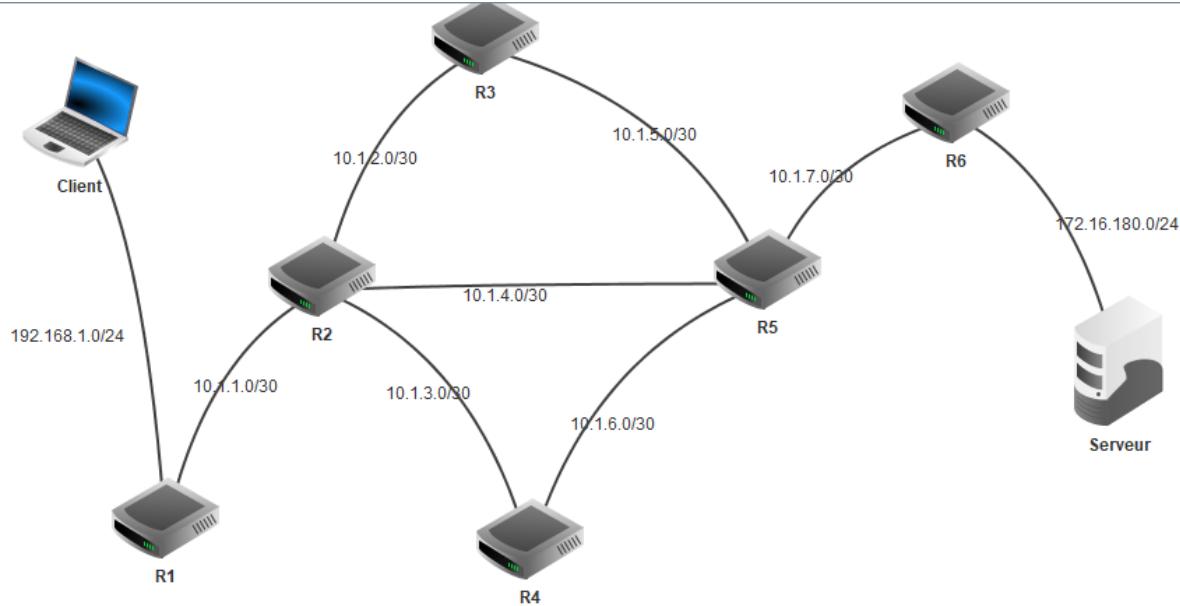
2.2. Débranche tout !

Chacun d'entre-vous se verra remettre un numéro de routeur, ainsi que la connaissance de ses voisins proches. Nous allons simuler l'exécution de l'algorithme de Bellman-Ford, en construisant petit-à-petit vos tables de routages.

2.3. Un exemple d'application

Considérons le réseau suivant, pour lequel on admettra la norme suivante :

- le poste client et le poste serveur se voient attribués respectivement la première adresse de la plage de leur réseau (soit respectivement 192.168.0.1 et 172.16.180.1) ;
- les routeurs d'accès R1 et R6 ont sur leur interface réseau les dernières adresses IP de la plage de leur réseau (soit respectivement 192.168.0.254 et 172.16.180.251) ;
- Entre deux interfaces internes, le routeur de plus bas indice possède la première adresse et le routeur de dernier indice la seconde adresse : par exemple entre R2 et R5, les interfaces sont connectées par le réseau 10.1.4.0/30 , donc l'interface de R2 est 10.1.4.1 et celle de R5 est 10.1.4.2 ;
- tous les routeurs suivent le protocole RIP.



Initialisation

Au démarrage, les routeurs ne connaissent que leur voisin immédiat. Les tables de routages pour les routeurs R1, R2 et R3 sont donc :

| Routeur R1 | | | | |
|----------------|-----------------|------------|---------------|----------|
| IP destination | Masque | Passerelle | Interface | Distance |
| 192.168.1.0 | 255.255.255.0 | | 192.168.1.254 | 1 |
| 10.1.1.0 | 255.255.255.252 | | 10.1.1.1 | 1 |
| | | | | |
| | | | | |

| Routeur R2 | | | | |
|----------------|-----------------|------------|-----------|----------|
| IP destination | Masque | Passerelle | Interface | Distance |
| 10.1.1.0 | 255.255.255.252 | | 10.1.1.1 | 1 |
| 10.1.2.0 | 255.255.255.252 | | 10.1.2.1 | 1 |
| 10.1.3.0 | 255.255.255.252 | | 10.1.3.1 | 1 |
| 10.1.4.0 | 255.255.255.252 | | 10.1.4.1 | 1 |

| Routeur R3 | | | | |
|----------------|-----------------|------------|-----------|----------|
| IP destination | Masque | Passerelle | Interface | Distance |
| 10.1.2.0 | 255.255.255.252 | | 10.1.2.2 | 1 |
| 10.1.5.0 | 255.255.255.252 | | 10.1.5.1 | 1 |
| | | | | |
| | | | | |

i | Principe de l'algorithme

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.

Quand il mets à jour les distances reçues, il **ajoute 1** à celles-ci, pour prendre en compte le *hop* (saut) supplémentaire effectué.

Etape 1

- R1 reçoit les routes de R2. Sa table de routage devient :

| Routeur R1 | | | | |
|----------------|-----------------|------------|---------------|----------|
| IP destination | Masque | Passerelle | Interface | Distance |
| 192.168.1.0 | 255.255.255.0 | | 192.168.1.254 | 1 |
| 10.1.1.0 | 255.255.255.252 | | 10.1.1.1 | 1 |
| 10.1.2.0 | 255.255.255.252 | 10.1.1.2 | 10.1.1.1 | 2 |
| 10.1.3.0 | 255.255.255.252 | 10.1.1.2 | 10.1.1.1 | 2 |
| 10.1.4.0 | 255.255.255.252 | 10.1.1.2 | 10.1.1.1 | 2 |

- R3 reçoit les routes de R2, et celles de R5. On peut constater qu'on peut atteindre de deux manière différentes les routeurs R2 et R5, car R3 ne réalise pas que les interfaces 10.1.2.1 et 10.1.4.1 sont toutes les deux deux interfaces de R2. De même il sait rejoindre de deux manière ayant la même métrique le réseau 10.1.4.0/30, donc il garde la première arrivée (ici on suppose d'il a reçu les informations de R2 avant celles de R5)

| Routeur R2 | | | | |
|----------------|-----------------|------------|-----------|----------|
| IP destination | Masque | Passerelle | Interface | Distance |
| 10.1.1.0 | 255.255.255.252 | | 10.1.1.1 | 1 |
| 10.1.2.0 | 255.255.255.252 | | 10.1.2.1 | 1 |
| 10.1.3.0 | 255.255.255.252 | | 10.1.3.1 | 1 |
| 10.1.4.0 | 255.255.255.252 | | 10.1.4.1 | 1 |
| 192.168.1.0 | 255.255.255.0 | 10.1.1.1 | 10.1.1.2 | 2 |
| 10.1.5.0 | 255.255.255.252 | 10.1.2.2 | 10.1.2.1 | 2 |

- R2 reçoit les routes de R1, puis de R3, mais aussi de R4 et de R5 :

| Routeur R3 | | | | |
|----------------|-----------------|------------|-----------|----------|
| IP destination | Masque | Passerelle | Interface | Distance |
| 10.1.2.0 | 255.255.255.252 | | 10.1.2.2 | 1 |
| 10.1.5.0 | 255.255.255.252 | | 10.1.5.1 | 1 |
| 10.1.1.0 | 255.255.255.252 | 10.1.2.1 | 10.1.2.2 | 2 |
| 10.1.3.0 | 255.255.255.252 | 10.1.2.1 | 10.1.2.2 | 2 |
| 10.1.4.0 | 255.255.255.252 | 10.1.2.1 | 10.1.2.2 | 2 |
| 10.1.7.0 | 255.255.255.252 | 10.1.5.2 | 10.1.5.1 | 2 |
| 10.1.6.0 | 255.255.255.252 | 10.1.5.2 | 10.1.2.2 | 2 |

Etape 2

- R1 reçoit les nouvelles routes de R2 :

| Routeur R1 | | | | |
|----------------|-----------------|------------|---------------|----------|
| IP destination | Masque | Passerelle | Interface | Distance |
| 192.168.1.0 | 255.255.255.0 | | 192.168.1.254 | 1 |
| 10.1.1.0 | 255.255.255.252 | | 10.1.1.1 | 1 |
| 10.1.2.0 | 255.255.255.252 | 10.1.1.2 | 10.1.1.1 | 2 |
| 10.1.3.0 | 255.255.255.252 | 10.1.1.2 | 10.1.1.1 | 2 |
| 10.1.4.0 | 255.255.255.252 | 10.1.1.2 | 10.1.1.1 | 2 |
| 10.1.5.0 | 255.255.255.252 | 10.1.1.2 | 10.1.1.1 | 3 |
| 10.1.6.0 | 255.255.255.252 | 10.1.1.2 | 10.1.1.1 | 3 |

- R2 reçoit les nouvelles routes de R3 et de R5 :

| Routeur R3 | | | | |
|----------------|-----------------|------------|-----------|----------|
| IP destination | Masque | Passerelle | Interface | Distance |
| 10.1.2.0 | 255.255.255.252 | | 10.1.2.2 | 1 |
| 10.1.5.0 | 255.255.255.252 | | 10.1.5.1 | 1 |
| 10.1.1.0 | 255.255.255.252 | 10.1.2.1 | 10.1.2.2 | 2 |
| 10.1.3.0 | 255.255.255.252 | 10.1.2.1 | 10.1.2.2 | 2 |
| 10.1.4.0 | 255.255.255.252 | 10.1.2.1 | 10.1.2.2 | 2 |
| 10.1.7.0 | 255.255.255.252 | 10.1.5.2 | 10.1.5.1 | 2 |
| 10.1.6.0 | 255.255.255.252 | 10.1.5.2 | 10.1.2.2 | 2 |
| 192.168.1.0 | 255.255.255.0 | 10.1.2.1 | 10.1.2.2 | 3 |
| 172.16.180.0 | 255.255.255.0 | 10.1.5.2 | 10.1.5.1 | 3 |

- R2 reçoit les nouvelles routes de R1 (qui ne lui apportent rien), de R3, de R4, de R5.... Très rapidement, la construction de ces routes devient absolument ingérable à la main... Mais voici quand même la table de routage :

| Routeur R2 | | | | |
|----------------|-----------------|------------|-----------|----------|
| IP destination | Masque | Passerelle | Interface | Distance |
| 10.1.1.0 | 255.255.255.252 | | 10.1.1.1 | 1 |
| 10.1.2.0 | 255.255.255.252 | | 10.1.2.1 | 1 |
| 10.1.3.0 | 255.255.255.252 | | 10.1.3.1 | 1 |
| 10.1.4.0 | 255.255.255.252 | | 10.1.4.1 | 1 |
| 192.168.1.0 | 255.255.255.0 | 10.1.1.1 | 10.1.1.2 | 2 |
| 10.1.5.0 | 255.255.255.252 | 10.1.2.2 | 10.1.2.1 | 2 |
| 10.1.6.0 | 255.255.255.252 | 10.1.3.2 | 10.1.3.1 | 2 |
| 10.1.7.0 | 255.255.255.252 | 10.1.4.2 | 10.1.4.1 | 2 |
| 172.16.180.0 | 255.255.255.0 | 10.1.4.2 | 10.1.4.2 | 3 |

Etape 3

Cette dernière table de R2 sera transmise à l'étape suivante à R1, et une liaison sera alors possible dès l'étape 3 entre le Client et le Serveur. Dans ce mini-réseau, les tables sont alors toutes stabilisées dès la troisième étape, et le réseau est totalement opérationnel.

3. Quelques points importants

Un des inconvénients du protocole RIP est qu'il génère un traffic important entre voisins, chacun envoyant sa table de routage à tous ses voisins à intervalles réguliers.

En conséquences, le protocole RIP est configuré pour se limiter à un **nombre maximum de 15 sauts**, ce qui le réserve à des réseaux de petites taille.

L'avantage de cette limite est qu'elle permet d'éviter les **éventuelles boucles de routage** (comme R2 => R3 => R5 => R2), en utilisant le TTL qui ne doit pas dépasser 15, afin qu'un paquet qui tourne en rond soit détruit.

De même, si un routeur ne reçoit pas d'informations d'un de ses voisins directs au bout d'un ordre de temps de trois minutes, il considère que ce routeur est hors-service et lui affecte une distance de 16, qui est alors considérée dans ce protocole comme une distance infinie. En conséquence tous ses voisins recalculeront leurs vecteurs en évitant le lien qui est tombé.