

TD 8

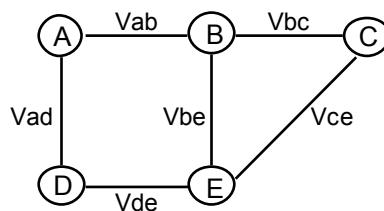
ROUTAGE

1. ALGORITHMES A VECTEURS DE DISTANCE

L'algorithme de base est dû à Bellman-Ford. Chaque nœud est supposé connaître la « distance » (ou le « coût ») qui le sépare de chacun de ses voisins. Périodiquement, chaque nœud envoie à chacun de ses voisins la liste des distances estimées vers chaque nœud du réseau : c'est le *vecteur de distance*. Sur réception d'un vecteur de distance, chaque nœud met à jour sa propre table de routage.

Exercice 1.1

Soit le réseau composé des 5 nœuds A, B, C, D et E, et des 6 liaisons V_{ab} , V_{ad} , V_{bc} , V_{be} , V_{ce} et V_{de} . A chaque liaison, supposée symétrique, est associée une distance égale à 1. L'algorithme utilisé par le protocole de routage est de type Bellman-Ford.



- On supposera que le réseau vient d'être mis en service et que chaque nœud n'a qu'une connaissance locale de la topologie du réseau (il ne connaît que ses voisins). Donner les tables de routage initiales des différents nœuds.

| A | B | C | D | E | | | | | | | |
|------|------|------|------|------|------|------|------|------|------|------|------|
| dest | suiv | dist |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |
| | | | | | | | | | | | |

- On considérera la séquence d'échange de vecteurs de distance suivante :

- T₁ B, D reçoivent VA (vecteur de distance de A)
- T₂ A, C, E reçoivent VB
- T₃ A, E reçoivent VD
- T₄ B, D reçoivent VA, VE
- T₅ B, E reçoivent VC
- T₆ A reçoit VB
- T₇ C, D reçoivent VE

Donnez la table de routage de chaque nœud, obtenue une fois que l'algorithme de routage a convergé.

3. La liaison Vab est rompue. Montrez comment les tables de routage de chaque nœud sont mises à jour. Que remarquez-vous à l'issue de la séquence d'échanges des vecteurs de distance suivante ?

- T₁ A et B détectent que Vab est rompue
- T₂ D reçoit VA ; C, E reçoivent VB
- T₃ E reçoit VD
- T₄ B, C, D reçoivent VE
- T₅ A reçoit VD

Exercice 1.2

On considère le même réseau que dans l'exercice précédent, excepté que la liaison Vce a un coût de 10 (les autres liaisons gardant un coût unitaire). On suppose qu'après convergence des algorithmes de routage, les tables obtenues sont les suivantes :

| A | | | B | | | C | | | D | | | E | | |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| dest | suiv | dist |
| B | B | 1 | A | A | 1 | A | B | 2 | A | A | 1 | A | B | 2 |
| C | B | 2 | C | C | 1 | B | B | 1 | B | A | 2 | B | B | 1 |
| D | D | 1 | D | A | 2 | D | B | 3 | C | A | 3 | C | B | 2 |
| E | B | 2 | E | E | 1 | E | B | 2 | E | E | 1 | D | D | 1 |

La liaison Vbc est alors rompue. B détecte la rupture, mais avant qu'il n'ait eu le temps d'envoyer son vecteur de distance, A a déjà diffusé le sien. La séquence d'échange est donc la suivante :

- T₁ B détecte que Vbc est rompue
- T₂ B, D reçoivent VA
- T₃ A, E reçoivent VB
- T₄ B, D reçoivent VA

1. Que se passe-t-il ?
2. A quel moment apparaît une information erronée dans les tables de routage ?
3. Quelle est l'origine de ce phénomène ?
4. On suppose que B envoie son vecteur de distance à A juste après avoir détecté la rupture, mais que, dans un même temps, A envoie spontanément son vecteur de distance à B (avant d'avoir reçu celui de B). Que se passe-t-il alors ?

Ce phénomène, appelé *comptage à l'infini*, peut être atténué par les méthodes suivantes :

- **Le vecteur de chemin** : chaque entrée du vecteur de distance contient le chemin complet associé à la valeur ; C'est une technique utilisée dans BGP, mais qui présente l'inconvénient de générer des tables volumineuses.
- **L'horizon partagé (split horizon)** : un routeur ne communique jamais le coût vers une destination à son voisin N, si N est le prochain nœud vers cette destination.

- **L'horizon partagé avec antidote (split horizon with poisonous reverse)** : Un routeur communique toujours un coût infini vers une destination à son voisin N, dès l'instant où N est le prochain nœud vers cette destination. Cela se traduit par une modification mineure du protocole de vecteurs de distance ; au lieu de diffuser le même vecteur sur toutes leurs liaisons, les nœuds devront en composer des versions différentes, pour tenir compte des destinations qui sont atteintes via chacune de ces liaisons. C'est la technique utilisée dans RIP.

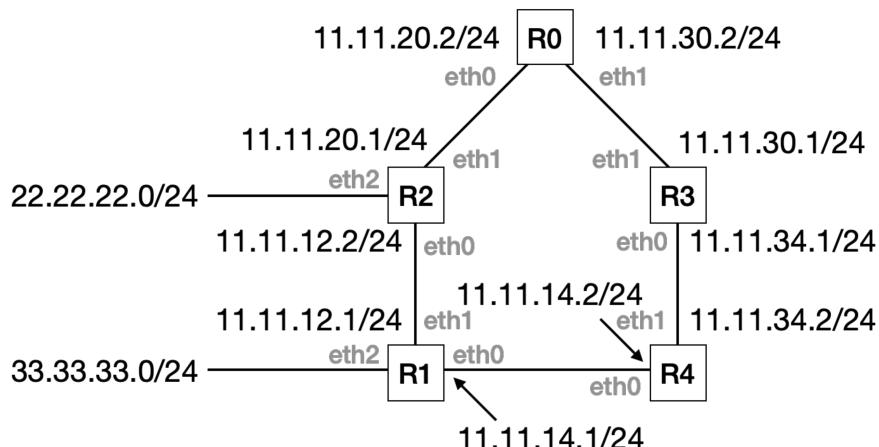
5. Qu'induisent ces différentes solutions sur l'exemple précédent ?

2. PROTOCOLE DE ROUTAGE RIP

RIP (*Routing Information Protocol*) est un protocole de routage IP de type vecteurs de distance s'appuyant sur l'algorithme décentralisé de Bellman-Ford.

Exercice 2.1

La figure ci-dessous représente un réseau où R0 et R4 sont des routeurs qui exécutent RIP. Le coût de tous les liens est égal à 1 et le masque de sous-réseau est 255.255.255.0 pour tous les sous-réseaux.



1. Appliquer les principes des algorithmes à vecteurs de distance pour donner la table de routage symbolique du routeur R4 à un instant T_1 postérieur à la convergence de l'algorithme.

| R4 | | |
|------|------|------|
| dest | suiv | dist |
| R0 | | |
| R1 | | |
| R2 | | |
| R3 | | |

Dans une table de routage RIP, la distance vers un sous-réseau est obtenue en prenant la distance symbolique conduisant au routeur le plus proche auquel le sous-réseau est directement connecté, et en l'incrémentant de 1. Ainsi, tous les sous-réseaux directement connectés à un routeur sont inscrits dans sa table de routage avec une distance de 1.

2. Déduire de la table symbolique précédente, la table de routage IP de R4.

| Destination | Mask | Gateway | Interface | Distance |
|-------------|------|---------|-----------|----------|
| 11.11.12.0 | | | | |
| 11.11.14.0 | | | | |
| 11.11.20.0 | | | | |
| 11.11.30.0 | | | | |
| 11.11.34.0 | | | | |
| 22.22.22.0 | | | | |
| 33.33.33.0 | | | | |

A l'instant $T_2 > T_1$, le lien ($R1, R2$) tombe en panne. $R1$ s'en rend compte, met à jour sa table de routage et envoie son vecteur de distance à $R4$.

3. Donner le vecteur de distance envoyé par $R1$ à $R4$ si la technique de l'horizon partagé n'est pas utilisée. Même question si elle est utilisée.

A l'instant $T_3 > T_2$, $R4$ reçoit le vecteur de distance de $R1$. Aucun autre message RIP n'a été émis ailleurs dans le réseau entre T_1 et T_3 . Sur réception de ce vecteur de distance, $R4$ met à jour sa table de routage.

4. Compléter la table de routage de $R4$ après mise à jour.

| Destination | Mask | Gateway | Interface | Distance |
|-------------|------|---------|-----------|----------|
| 11.11.12.0 | | | | |
| 11.11.14.0 | | | | |
| 11.11.20.0 | | | | |
| 11.11.30.0 | | | | |
| 11.11.34.0 | | | | |
| 22.22.22.0 | | | | |
| 33.33.33.0 | | | | |

Tous les routeurs continuent à s'échanger leurs vecteurs de distance. A l'instant $T_4 > T_3$, on suppose que l'algorithme a convergé.

5. Donner la table de routage de $R4$ après convergence et disparition des entrées obsolètes.

| Destination | Mask | Gateway | Interface | Distance |
|-------------|------|---------|-----------|----------|
| 11.11.14.0 | | | | |
| 11.11.20.0 | | | | |
| 11.11.30.0 | | | | |
| 11.11.34.0 | | | | |
| 22.22.22.0 | | | | |
| 33.33.33.0 | | | | |

3. ALGORITHMES A ETATS DE LIEN

Dans cette approche, un routeur communique à tous les autres nœuds du réseau sa distance avec ses voisins. L'idée est en effet de distribuer la topologie du réseau et le coût de chaque lien à tous les routeurs. Chaque routeur peut alors calculer, de façon autonome, son chemin optimum pour chaque destination. Il utilise pour cela l'algorithme de Dijkstra :

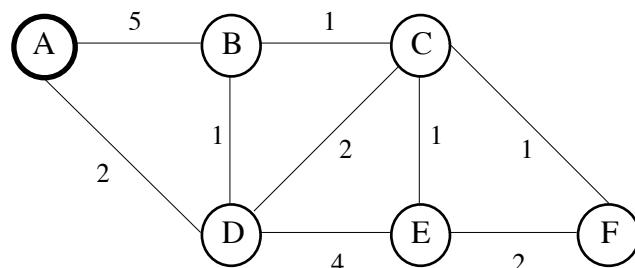
```

DIJKSTRA (G, s)
pour chaque sommet v ∈ S faire
    d(v) = ∞
    pred(v) = nil
fin pour
d(s) = 0
E = ∅
R = S
tant que R ≠ ∅ faire
    u = sommet avec l'estimation de pondération minimum de R
    E = E ∪ {u}
    pour chaque sommet v ∈ voisin[u] faire
        si d(v) > d(u) + c(u,v) alors
            d(v) = d(u) + c(u,v)
            pred(v) = u
    R = R - {u}
fin tant que

```

Exercice 1.1

On considère le réseau suivant composé des 6 nœuds :



Les nœuds appliquent le principe de l'algorithme de Dijkstra pour évaluer les plus courts chemins vers les autres nœuds du réseau et calculer leur table de routage. On suppose que tous les nœuds connaissent la topologie complète du réseau (représentée sur la figure).

1. Appliquer l'algorithme de Dijkstra au niveau du nœud A.
2. Donner l'arbre des plus courts chemins (partant de A) issu de l'exécution de l'algorithme.
3. En déduire la table de routage de A.