

Exercice 1

Question 1: dans l'algorithme RIP, quelles sont les informations présentes dans les vecteurs de distance qu'échangent les routeurs ?

Corrigé: pour chaque destination, le routeur stocke 3 informations: (destination, distance, next hop). Il en transmet deux dans son vecteur de distance (destination, distance)

Question 2: on considère les deux réseaux suivants. On supposera que les machines émettent leur vecteur de distance dans l'ordre suivant : A émet en premier puis C émet puis B émet. Appliquez l'algorithme RIP sur les deux réseaux :

A ----- B ----- C (on suppose chaque liaison de poids 4).

corrigé:

étape 0:

De/ vers ->	A	B	C
A	(A, 0)	X	X
B	X	(B, 0)	X
C	X	X	(C, 0)

Étape 1 (après les envois de A, C et B)

De/ vers ->	A	B	C
A	(A, 0)	(B, 1)	(B, 2)
B	(A, 1)	(B, 0)	(C, 1)
C	(B, 2)	(B, 1)	(C, 0)

De/ vers ->	A	B	C
A	(A, 0)	(B, 1)	(B, 2)
B	(A, 1)	(B, 0)	(C, 1)
C	(B, 2)	(B, 1)	(C, 0)

Le tableau ne change pas. On a convergé

A ----- B ----- C on suppose que tous les liens ont un poids de 1.

```

|           |
|           |
D-----E

```

De/Vers =>	A	B	C	D	E
A	(A, 0)				
B		(B, 0)			
C			(C, 0)		
D				(D, 0)	
E					(E, 0)

De/Vers =>	A	B	C	D	E
A	(A, 0)	(B, 1)	(B, 2)	(D, 1)	
B	(A, 1)	(B, 0)	(C, 1)		
C	(B, 2)	(B, 1)	(C, 0)	(E, 2)	(E, 1)
D	(A, 1)		(E, 2)	(D, 0)	(E, 1)
E	(D, 2)		(C, 1)	(D, 1)	(E, 0)

De/Vers =>	A	B	C	D	E
A	(A, 0)	(B, 1)	(B, 2)	(D, 1)	(B, 3) (D, 2)
B	(A, 1)	(B, 0)	(C, 1)	(A, 2)	(C, 2)
C	(B, 2)	(B, 1)	(C, 0)	(E, 2)	(E, 1)
D	(A, 1)	(A, 2)	(E, 2)	(D, 0)	(E, 1)
E	(D, 2)	(C, 2)	(C, 1)	(D, 1)	(E, 0)

De/Vers =>	A	B	C	D	E
A	(A, 0)	(B, 1)	(B, 2)	(D, 1)	(D, 2)
B	(A, 1)	(B, 0)	(C, 1)	(A, 2)	(C, 2)
C	(B, 2)	(B, 1)	(C, 0)	(E, 2)	(E, 1)
D	(A, 1)	(A, 2)	(E, 2)	(D, 0)	(E, 1)
E	(D, 2)	(C, 2)	(C, 1)	(D, 1)	(E, 0)

L'application de l'algo ne change pas le tableau donc on a convergé.
Le lien entre B et C est coupé :

De/Vers =>	A	B	C	D	E
A	(A, 0)	(B, 1)	(B, 2)	(D, 1)	(D, 2)
B	(A, 1)	(B, 0)	(C, inf.)	(A, 2)	(C, inf.)
C	(B, inf.)	(B, inf.)	(C, 0)	(E, 2)	(E, 1)
D	(A, 1)	(A, 2)	(E, 2)	(D, 0)	(E, 1)
E	(D, 2)	(C, 2)	(C, 1)	(D, 1)	(E, 0)

De/Vers =>	A	B	C	D	E
A	(A, 0)	(B, 1)	(B, 2)	(D, 1)	(D, 2)
B	(A, 1)	(B, 0)	(A, 3)	(A, 2)	(A, 3)
C	(B, inf.)	(B, inf.)	(C, 0)	(E, 2)	(E, 1)
D	(A, 1)	(A, 2)	(E, 2)	(D, 0)	(E, 1)
E	(D, 2)	(C, 2)	(C, 1)	(D, 1)	(E, 0)

De/Vers =>	A	B	C	D	E
A	(A, 0)	(B, 1)		(D, 1)	
B	(A, 1)	(B, 0)	(C, 1)		
C	(B, 2)	(B, 1)	(C, 0)	(E, 2)	(E, 1)
D	(A, 1)	(E, 3)	(E, 2)	(D, 0)	(E, 1)
E	(C, 3) puis (D, 2)	(C, 2)	(C, 1)	(D, 1)	(E, 0)

De/Vers =>	A	B	C	D	E
A	(A, 0)	(B, 1)	(B, 2)	(D, 1)	(D, 2)
B	(A, 1)	(B, 0)	(C, 1)	(A, 2)	(C, 2)
C	(B, 2)	(B, 1)	(C, 0)	(E, 2)	(E, 1)
D	(A, 1)	(A, 2)	(E, 2)	(D, 0)	(E, 1)
E	(D, 2)	(C, 2)	(C, 1)	(D, 1)	(E, 0)

De/Vers =>	A	B	C	D	E
A	(A, 0)	(B, 1)	(B, 2)	(D, 1)	(D, 2)
B	(A, 1)	(B, 0)	(C, 1)	(A, 2)	(C, 2)
C	(B, 2)	(B, 1)	(C, 0)	(E, 2)	(E, 1)
D	(A, 1)	(A, 2)	(E, 2)	(D, 0)	(E, 1)
E	(D, 2)	(C, 2)	(C, 1)	(D, 1)	(E, 0)

Le tableau ne change pas. On a convergé

Question 3: dans le réseau précédent, le lien entre B et C est coupé. On suppose que A émet son vecteur de distance après que B ait mis à jour ses tables mais avant qu'il ait émis le vecteur de distance correspondant.

Corrigé:

De/ vers ->	A	B	C
A	(A, 0)	(B, 1)	(B, 2)
B	(A, 1)	(B, 0)	(C, 16)
C	(B, 16)	(B, 16)	(C, 0)

De/ vers ->	A	B	C
A	(A, 0)	(B, 1)	(B, 4)
B	(A, 1)	(B, 0)	(A, 3)
C	(B, 16)	(B, 16)	(C, 0)

IMPORTANT : La modification à la hausse sur A est due au fait que B est le next hop : toute annonce venant du next hop pour une destination met à jour la distance.

De/ vers ->	A	B	C
A	(A, 0)	(B, 1)	(B, 6)
B	(A, 1)	(B, 0)	(A, 5)
C	(B, 16)	(B, 16)	(C, 0)

Le même mécanisme se poursuit. Il s'arrêtera quand la distance de A vers C en passage par B atteindra la valeur infinie (=16).

De/Vers =>	A	B	C	D	E
A	(A, 0)	(B, 1)	(B, 2)	(D, 1)	(D, 2)
B	(A, 1)	(B, 0)	(C, 16)	(A, 2)	(C, 16)
C	(B, 16)	(B, 16)	(C, 0)	(E, 2)	(E, 1)
D	(A, 1)	(A, 2)	(E, 2)	(D, 0)	(E, 1)
E	(D, 2)	(C, 2)	(C, 1)	(D, 1)	(E, 0)

Ordre des envois (choisi arbitrairement pour l'exemple) : A puis B puis C puis D puis E

De/Vers =>	A	B	C	D	E
A	(A, 0)	(B, 1)	(B, 4) puis (D, 3)	(D, 1)	(D, 2)
B	(A, 1)	(B, 0)	(A, 3)	(A, 2)	(A, 3)
C	(E, 3)	(E, 4)	(C, 0)	(E, 2)	(E, 1)
D	(A, 1)	(A, 2)	(E, 2)	(D, 0)	(E, 1)
E	(D, 2)	(C, 16) puis (D, 3)	(C, 1)	(D, 1)	(E, 0)

De/Vers =>	A	B	C	D	E
A	(A, 0)	(B, 1)	(D, 3)	(D, 1)	(D, 2)
B	(A, 1)	(B, 0)	(A, 4)	(A, 2)	(A, 3)
C	(E, 3)	(E, 4)	(C, 0)	(E, 2)	(E, 1)
D	(A, 1)	(A, 2)	(E, 2)	(D, 0)	(E, 1)
E	(D, 2)	(D, 3)	(C, 1)	(D, 1)	(E, 0)

De/Vers =>	A	B	C	D	E
A	(A, 0)	(B, 1)	(D, 3)	(D, 1)	(D, 2)
B	(A, 1)	(B, 0)	(A, 4)	(A, 2)	(A, 3)
C	(E, 3)	(E, 4)	(C, 0)	(E, 2)	(E, 1)
D	(A, 1)	(A, 2)	(E, 2)	(D, 0)	(E, 1)
E	(D, 2)	(D, 3)	(C, 1)	(D, 1)	(E, 0)

On a convergé.