M1 Informatique # Protocoles réseaux

Algorithme de routage à vecteur de distance

- Les réseaux informatiques emploient généralement des algorithmes de routage dynamique plus complexes que l'inondation, mais plus efficaces car ils trouvent les chemins les plus courts correspondant à la topologie actuelle.
- Les deux algorithmes dynamiques les plus connus sont celui à **vecteur de distance** et celui par informations d'état de lien.

- Alors que l'algorithme d'état de lien repose sur une connaissance de la topologie globale du réseau, l'algorithme à **vecteur de distance** (DV, *Distance Vector*) est de nature ***itérative***, ***asynchrone*** et ***distribuée***.

  - **Distribuée** : parce que les calculs se font au niveau de chaque nœud individuel, à partir des informations fournies par les *nœuds voisins directs*, et parce que les résultats sont partagés de la même manière.
  - **Itérative** : car ce processus se répète jusqu'à ce qu'il n'y ait plus d'informations à échanger entre nœuds voisins. De fait, cet algorithme s'interrompt de lui-même.
  - **Asynchrone** : dans la mesure où il n'impose pas à tous les nœuds de travailler ensemble.

  - Un algorithme distribué, asynchrone, itératif et à interruption automatique est plus intéressant qu'un algorithme centralisé.

- Avec les algorithmes de **routage par vecteur de distance** (*distance vector routing*), chaque routeur maintient sa propre **table de routage** : La principale structure de données dans un algorithme DV.
- C'est une table de distances tenue à jour au niveau de chaque nœud.

- La table de distances d'un nœud donné contient une ligne pour chaque destination possible au sein du réseau et une colonne pour chacun des voisins directs d'un nœud donné.

- **Un exemple** permet d'appréhender le contenu d'une table de distances. La topologie d'un réseau est illustrée à la figure 1, avec la table de distances présentée à droite, correspondant au nœud **E** après convergence de l'algorithme.

- Il met régulièrement à jour sa table de routage avec les informations reçues des routeurs voisins. Finalement, chaque routeur connait le meilleur lien à utiliser pour atteindre chaque destination.

- L'établissement des routes vers les meilleurs chemins est appelé **convergence**.

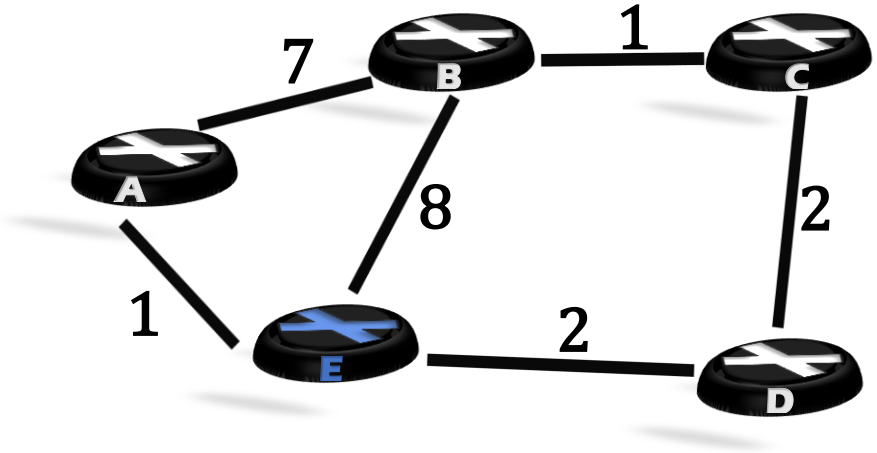


Figure 1 · Table de distances du nœud d'origine E

Cout du parcours vers la destination via (le nœud suivant sur le parcours)

Destination	$D^E(\ )$	A	B	D
	A	(1)	14	5
	B	7	8	(5)
	C	6	9	(4)
	D	4	11	(2)

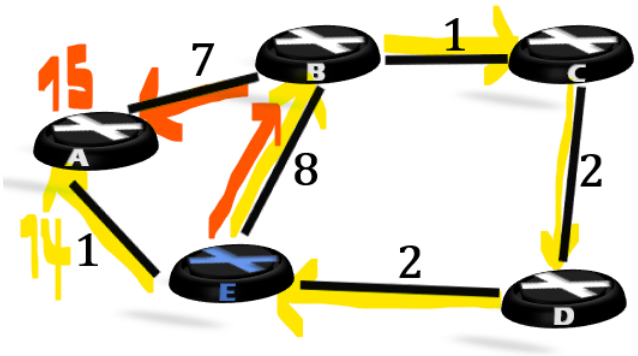
Les cercles autour de certaines valeurs de la table de distances signalent les parcours les moins onéreux en direction du nœud correspondant à la ligne dans laquelle elles se trouvent.

Intéressons-nous à la ligne concernant le destinataire A :

$D^E(\ )$	A	B	D
A	1	14	5

- Le cout du trajet direct de **E** vers A présente une valeur de 1.
  Par conséquent, $\mathbf{D^E(A, A) = 1}$

- La valeur de $\mathbf{D^E(A, D)}$ représente le cout du parcours conduisant de **E** à A en passant par D (**routeur du premier bond**).
    - L'entrée dans la table correspond à la somme de la valeur de la liaison reliant **E** et D (2) et de celle du parcours le moins onéreux entre D et A.
    - Celle-ci est de 3 bien que le parcours en question repasse par E ! Le parcours le moins onéreux entre E et A et passant par D présente une valeur totale de 5.
    - Il est difficile cependant de penser que le parcours le moins onéreux repassant par l'origine peut générer certains problèmes en aval **(et c'est pourtant le cas !)**.
- De la même manière, on observe que l'entrée dans la table de distances du parcours le moins onéreux passant par B et $\mathbf{D^E(A, B) = 14}$, et non pas 15 !



- Cette technique est également souvent appelée algorithme de routage de **Bellman-Ford**, en hommage aux chercheurs qui l'ont développée. C'est l'algorithme original du réseau ARPAnet, qui a également été utilisé par le protocole RIP (*Routing Information Protocol*) sur l'Internet.
    - Il est utilisé par de nombreux protocoles de routage supplémentaires, parmi lesquels les protocoles RIP et BGP de l'internet, IRDP de l'ISO, IPX de Novell.

- Dans cette technique chaque routeur maintient une table de routage indexée contenant une entrée par routeur. Chaque entrée se compose de deux parties, la ligne de sortie préférée vers la destination en question et une estimation du temps ou de la distance séparant cette destination du routeur.

- Ainsi, la **table d'acheminement** d'un nœud (qui indique la liaison de sortie appropriée pour transmettre des paquets vers une destination donnée) peut aisément être obtenue à partir de sa table de distances.

- La distance est mesurée en nombre de sauts ou toute autre métrique.

- Le routeur est supposé connaitre la distance vers chacun des routeurs adjacents. Si la métrique est le nombre de sauts, cette distance est de 1.

- Si la métrique est le délai de propagation, le routeur peut le mesurer directement en envoyant des paquets ECHO spéciaux que le destinataire retourne le plus vite possible avec des informations de temps.

- **Par exemple,** Imaginez un nœud X souhaitant acheminer un paquet vers une destination Y *via* son voisin direct Z.
  - L'entrée de la table de distances du nœud X, $D^x(Y,Z)$ correspond à la somme du cout de la liaison directe entre X et Z, $c(X,Z)$ et de la valeur du parcours le moins onéreux actuel entre Z et Y, ce qui conduit à l'expression :

$$D^x(Y,Z) = c(X,Z) + min_w\{D^z(Y,w)\}$$

dans laquelle le terme min_w est choisi parmi tous les voisins directs de Z (y compris X).

- Cette équation donne une idée du type de communication de voisinage qui s'établit avec l'algorithme DV, dans lequel chaque nœud doit connaitre la valeur du parcours le moins onéreux de tous ses voisins pour toutes les destinations possibles. Dès qu'un nœud trouve un nouveau parcours moins onéreux (couteux) vers une destination donnée, il est chargé d'en informer tous ses voisins.

- **Par exemple,** imaginons que la métrique utilisée soit le délai d'acheminement et que chaque routeur connaisse le temps qu'il met pour atteindre chacun de ses routeurs voisins.
  - Toutes les T ms, chaque routeur envoie à chacun des routeurs contigus une liste des délais estimés pour atteindre chaque destination proposée.
  - Supposons que l'une de ces tables vienne d'arriver de la part du routeur voisin X, avec X_i le délai estimé pour atteindre le routeur i.
  - Si le routeur récepteur sait que le délai qui le sépare de X est m ms, il sait aussi qu'il peut atteindre i par l'intermédiaire de X avec un délai de $X_i + m$ ms.
  - En effectuant ce calcul pour chaque routeur adjacent, un routeur peut déterminer le délai estimé qui lui semble le meilleur et le porter dans sa table de routage en y associant le lien de sortie appropriée.
  - Notez que l'ancienne table de routage n'est pas utilisée dans le calcul.

- **Le processus de mise à jour** est illustré à la figure 2. La partie (a) illustre un réseau.

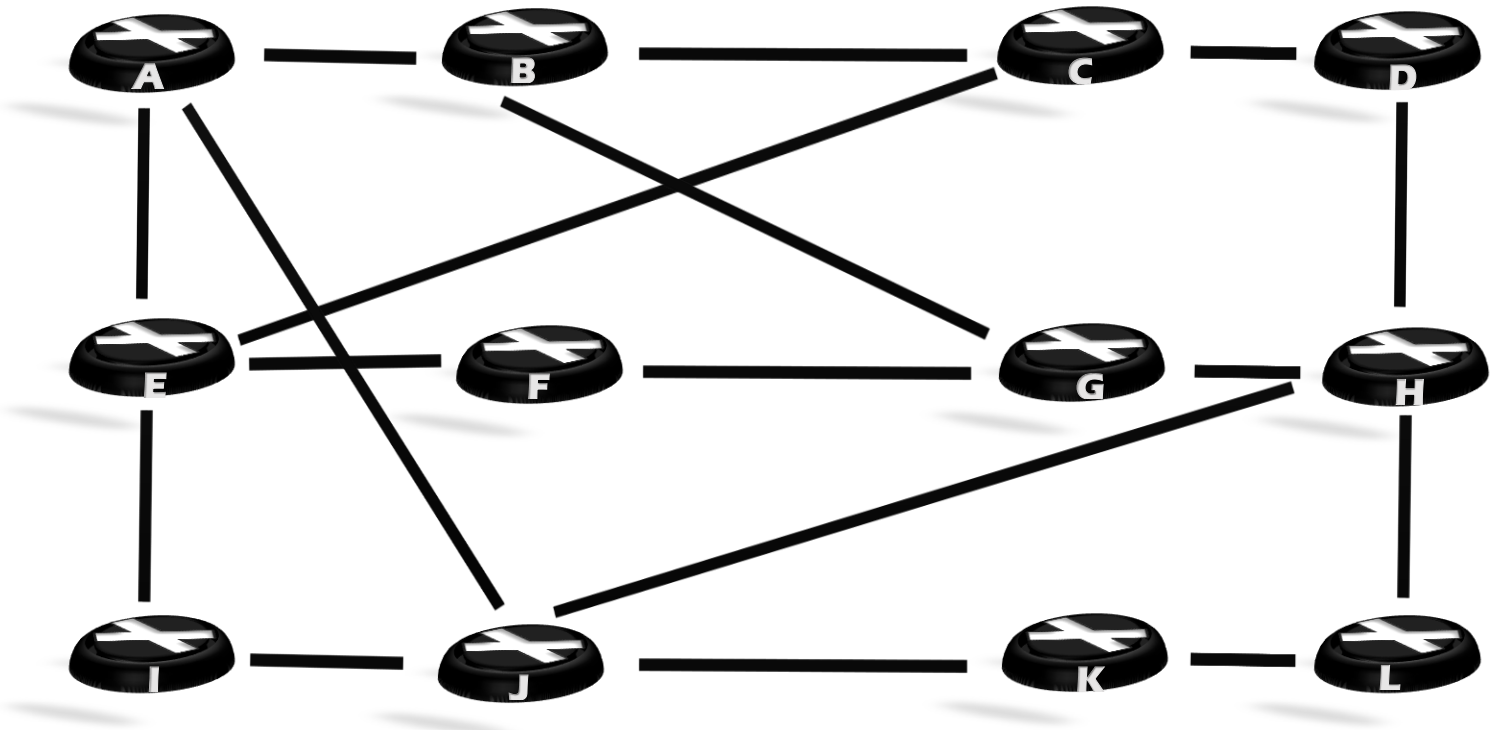
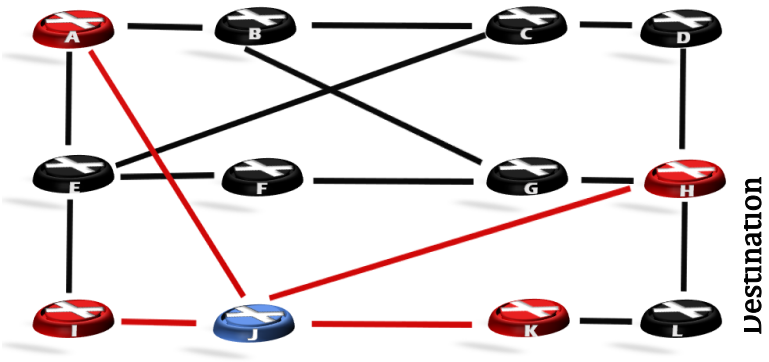


Figure 2 · (a) Un réseau

- Supposons que ***J*** ait calculé ou estimé que les délais vers ***A***, ***I***, ***H*** et ***K*** étaient respectivement de 8, 10, 12 et 6 ms.



J	
	Distance
J	0 ms
A	8 ms
H	12 ms
I	10 ms
K	6 ms

- Les vecteurs de délai reçus des voisins directs du routeur ***J***.
  Le routeur ***A*** prétend atteindre ***B*** en 12 ms, ***C*** en 25 ms, ***D*** en 40 ms, etc.

A	
A	0
B	12
C	25
D	40
E	14
F	23
G	18
H	17
I	21
J	9
K	24
L	29

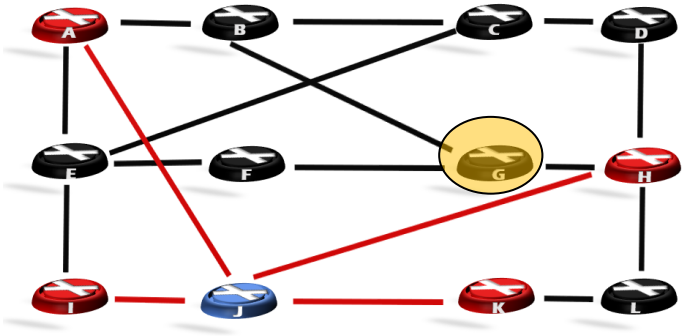
I	
A	24
B	36
C	18
D	27
E	7
F	20
G	31
H	20
I	0
J	11
K	22
L	33

K	
A	21
B	28
C	36
D	24
E	22
F	40
G	31
H	19
I	22
J	10
K	0
L	9

H	
A	20
B	31
C	19
D	8
E	30
F	19
G	6
H	0
I	14
J	7
K	22
L	9

- Examinons de quelle façon ***J*** calcule sa nouvelle route vers ***G***. Il sait qu'il peut atteindre ***A*** en 8 ms. De plus, ***A*** prétend pouvoir atteindre ***G*** en 18 ms. Par conséquent, ***J*** estime qu'il peut envoyer un paquet se destinant à ***G*** en 26 ms s'il le fait passer par ***A***.

  Il effectue les mêmes calculs pour chacun de ses voisins. ***I***, ***H***, et ***K***, et il obtient des estimations de 41 (30 + 10), 18 (6 + 12) et 37 (31 + 6), respectivement. La meilleure valeur étant 18, il crée une entrée dans sa table de routage en indiquant que le délai vers ***G*** est de 18 ms et que la route à emprunter est celle qui passe par ***H***.



J		
Destination	Distance	nh
J	0 ms	-
A	8 ms	A
G	18 ms	H
H	12 ms	H
I	10 ms	I
K	6 ms	K

- Le même calcul est réalisé pour toutes les destinations, et ***J*** obtient la nouvelle table de routage c'est-à-dire un vecteur lui indiquant la meilleure distance vers chaque destination et le lien à utiliser dans chaque cas.

Destination	Distance	nh
A	8 ms	A
B	20	A
C	28	I
D	20	H
E	17	I
F	30	I
G	18 ms	H
H	12 ms	H
I	10 ms	I
J	0	-
K	6 ms	K
L	15	K

Algorithme à vecteur de distance (DV)

A chaque nœud X :

// Initialisation

Pour tous les nœuds adjacents v :

$D^x(*, v) = ?$ // Le signe * signifie ''pour toutes les lignes''

$D^x(v, v) = c(X,v)$

Pour toutes les destinations y :

envoyer $\min_w D(y, w)$ à chaque voisin // w à tous les voisins de X

Effectuer une boucle

Attendre jusqu'à constater un changement de cout de ligne v *OU*
jusqu'à recevoir une mise à jour venant de v

Si (c(x,v) devient d) // Note : d peut être positif ou négatif

// Changer les couts de toutes les destination *via* v par d

Pour toutes les destinations y :

$D^x(y, v) = D^x(y, v) + d$

Ou bien

// mise à jour reçue de v wrt à destination de y : le chemin le plus court de v à y a changé

// v a transmis une nouvelle valeur pour : $\min_w D^v(y, w)$

// Cette nouvelle valeur est : ''newval''

Pour la destination y :

$D^x(y, v) = c(X, v)$ 1 newval

Si nouveau $\min_w D^X(y, w)$ pour toute destination y

envoyer une nouvelle valeur de $\min_w D^X(y, w)$ à tous les voisins

Sans cesse

Sources :

Andrew Tanenbaum, David Wetherall. Computer Networks.. (Edition française)

James W. Kurose, Keith W. Ross. Computer Networking, A Top-Down Approach. (Edition française)