IUT Bordeaux 1 **Année 2012-2013**
Département Informatique **S2**

TD7 ASR2 Réseau
Deux grands principes d'algorithme de routage

Routage à vecteur de distance

Technique fondée sur l'algorithme de Bellman-Ford, fonctionnant ainsi :

- chaque routeur connaît le temps d'acheminement avec ses voisins
- périodiquement, chaque routeur transmet la liste des temps estimés vers chaque destination et reçoit de ses voisins leur propre liste
- en fonction de ces informations, modifier les tables de routage.

Technique utilisée à l'origine dans le réseau **ARPANET** *et par le protocole* **RIP** *(Routing Information Protocol) du réseau Internet.*

Routage par information d'état des liens

Sur chaque routeur, faire :

1. découvrir ses voisins et apprendre leurs adresses
2. mesurer les coûts
3. envoyer ces informations à tous les routeurs
4. calculer les plus courts chemins par l'algorithme de Dijkstra

Technique très utilisée, notamment par le protocole **OSPF** *utilisé dans le réseau Internet et par le protocole* **IS-IS** *pour le réseau DECnet.*

1 Exercice sur le routage à vecteur de distance

Dans un réseau, le noeud A reçoit les tables de routage suivantes de ses voisins :

Dest.	Lien	Coût
A	A	3
C	A	5
D	D	2
E	D	4
F	D	11
G	D	10
H	D	7

Table de B

Dest.	Lien	Coût
A	A	6
B	A	12
D	E	9
E	E	7
F	E	15
G	H	4
H	H	1

Table de C

Dest.	Lien	Coût
A	A	2
B	D	6
C	C	3
D	D	2
F	F	8
G	C	7
H	C	4

Table de E

1. Les sommets B et D sont-ils voisins ? Les sommets E et G sont-ils voisins ?
2. Quel sont les coûts associés aux liens $E \to C$ et $D \to F$?
3. Calculer la table de routage de A, sachant que les coûts des liens entre A et ses voisins sont les suivants : $c(AB) = 8$, $c(AC) = 2$ et $c(AE) = 1$ et que pour un noeud Y quelconque, le coût du chemin entre A et Y est donné par $c(AY) = \min_X(c(AX) + c(XY))$, X étant un noeud voisin de A.

2 Exercice sur le routage par information d'état des liens

Dans un réseau, le noeud A reçoit les paquets d'information d'état des liens de chaque noeud ; il connait donc les voisins de chaque noeud ainsi que les coûts associés :

A	
B	4
E	5

B	
A	4
C	2
F	6

C	
B	2
D	7
E	1

D	
C	7
F	3

E	
A	5
C	1
F	3

F	
B	6
D	3
E	3

1. Aider A à reconstruire le réseau.
2. Calculer les tables de routage de A et D en utilisant l'algorithme de Dijkstra.

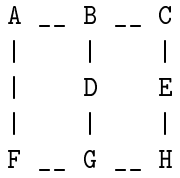
A GARDER POUR UNE AUTRE FOIS PEUT-ETRE

3 Routage "optimal"

Principe d'optimalité :
Si un noeud de routage J est sur le chemin optimal qui va de I vers K, le chemin optimal de J vers K suit la même route.

Conséquence : l'ensemble de toutes les routes optimales de toutes les sources vers une destination donnée est un arbre (appelé arbre collecteur) dont la racine est la destination. On peut donc parler de noeud en "amont" ou en "aval".

On considère le réseau suivant :



1. Construire l'arbre collecteur pour la destination H. (en terme de nombre de sauts avec comme algorithme du plus court chemin celui qui correspond à l'ordre lexicographique. Par exemple la route BCEH est plus courte que BDGH)

Chaque noeud tient à jour une table de routage avec une ligne par destination. Chaque colonne donne le nombre de sauts pour atteindre la destination via une de ses lignes de sortie (pour G, les lignes de sortie sont D,F et H). Les entrées qui font référence aux noeuds en amonts sont laissés vierges.

2. Construire la table de routage de G. Repérer la meilleur route pour chaque destination.

Regardons ce qui arrive si la ligne GH est coupée.

3. Que se passe-t-il pour la destination E ? Pour la destination H ?

4. Que se passe-t-il si le réseau est séparé en 2 parties disjointes ?