Série 14

1 Algorithme AIMD et autres

Au cours, nous avons vu que l'algorithme AIMD (augmentation additive - retrait multiplicatif; "additive increase - multiplicative decrease" en anglais) permet de garantir un partage équitable du réseau lorsque deux utilisateurs A_1 et A_2 cherchent à communiquer de manière concurrente sur le réseau. Pour rappel, cet algorithme consiste, pour chaque utilisateur, à mettre à jour W , le nombre de paquets envoyés par unité de temps, de la manière suivante:

$W \rightarrow W + a$ tant que tout se passe bien (avec $a > 0$ un paramètre fixé);

$W \rightarrow bW$ dès qu'un problème survient (avec $0 < b < 1$ un autre paramètre fixé).

En représentant l'évolution de W_1 et W_2 sur un même graphique (avec W_1 en abscisse et W_2 en ordonnée), nous avons vu en particulier que cet algorithme a pour effet, à terme, de rapprocher les valeurs de W_1 et W_2 , et donc de permettre un partage équitable du réseau.

En réutilisant le même graphique avec W_1 en abscisse et W_2 en ordonnée, expliquez pourquoi *aucun* des trois algorithmes suivants ne permet d'atteindre ce but:

a) Algorithme AIAD (augmentation additive, retrait additif):

$W \rightarrow W + a$ tant que tout se passe bien (avec $a > 0$ un paramètre fixé);

$W \rightarrow W - \min(W, b)$ dès qu'un problème survient (avec $b > 0$ un autre paramètre fixé).

b) Algorithme MIMD (augmentation multiplicative, retrait multiplicatif):

$W \rightarrow aW$ tant que tout se passe bien (avec $a > 1$ un paramètre fixé);

$W \rightarrow bW$ dès qu'un problème survient (avec $0 < b < 1$ un autre paramètre fixé).

c) Algorithme MIAD (augmentation multiplicative, retrait additif):

$W \rightarrow aW$ tant que tout se passe bien (avec $a > 1$ un paramètre fixé);

$W \rightarrow W - \min(W, b)$ dès qu'un problème survient (avec $b > 0$ un autre paramètre fixé).

Note: Lors d'un retrait additif, on ne peut pas systématiquement soustraire une valeur fixe $b > 0$, au risque d'obtenir une valeur négative de W ; c'est pourquoi on soustrait $\min(W, b)$ au lieu de b .

2 Routage

a) On considère un réseau comprenant 6 noeuds nommés A, B, C, D, E et F. On connaît (en partie) leurs tables de routage:

A		
dest	dir	dist
C	B	2
E	F	2
D	B	3

B		
dest	dir	dist
D	C	2
F	A	2
E	C	2

C		
dest	dir	dist
F	E	2
A	B	2

et

D		
dest	dir	dist
B	C	2
E	C	2
F	C	3
A	C	3

E		
dest	dir	dist
D	C	2
A	F	2
B	C	2

F		
dest	dir	dist
B	A	2
C	E	2
D	E	3

Le noeud B tombe en panne et n'est plus utilisable. Tous les autres noeuds sont avertis de cette panne et leurs tables de routage sont mises à jour de façon à éviter le noeud B.

Existe-t-il encore une route de A à D après la panne, et si oui, quelle est sa longueur?

b) On considère maintenant un autre réseau dans lequel se trouvent plusieurs noeuds A, B, C, ..., N, O. On connaît en partie les tables de routage des noeuds A, F et H, qui sont:

A		
dest	dir	dist
B	C	2
D	C	5
N	x	y

F		
dest	dir	dist
J	O	2
L	K	2
M	N	2

H		
dest	dir	dist
D	I	2
C	B	2
F	J	3

En se basant uniquement sur un strict minimum de liens entre noeuds qui *doivent* exister selon les tables ci-dessus (i.e., sans imaginer d'autres liens non-justifiés par ces tables), pouvez-vous déduire les valeurs de x et y dans la table de routage de A?

Indication (à lire que si vous ne voyez pas comment faire!): Dans cet exercice, pour construire le réseau à partir des tables de routage, vous devez d'abord vous focaliser uniquement sur les indications des noeuds qui sont à distance 2 dans les tables (puis vérifier avec les autres que le réseau que vous avez construit est cohérent).