

Série 13 : Solutions

1 Couches et adresses

a) Les trois adresses correspondent à trois couches différentes du modèle TCP/IP :

Adresse	Exemple	Couche	Portée
f8:f2:1e:aa:cd:c0	adresse MAC	2 – Liaison	Locale (réseau local)
172.16.254.1	adresse IP	3 – Réseau	Globale (bout en bout)
443	numéro de port	4 – Transport	Machine (processus)

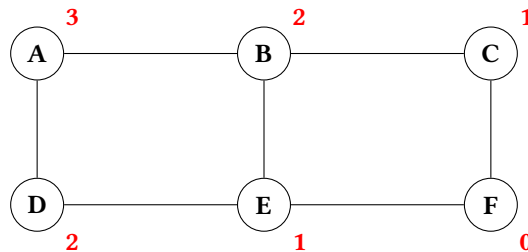
b) Pour chaque opération :

1. Requête HTTP GET / : couche **Application** (5).
2. Numéro de séquence et ACK : couche **Transport** (4) – c'est typiquement le rôle de TCP.
3. Routage de proche en proche selon l'adresse IP : couche **Réseau** (3) – protocole IP.
4. Gestion des collisions et réémission locale sur le Wi-Fi : couche **Liaison** (2).
5. Transmission des bits sous forme d'ondes radio : couche **Physique** (1).

c) Un routeur intermédiaire n'a besoin de remonter que jusqu'à la **couche Réseau** (couche 3). En effet, son rôle est uniquement de lire l'**adresse IP de destination** contenue dans l'en-tête IP afin de décider vers quel voisin envoyer le paquet. Il n'a donc pas à savoir s'il s'agit d'un paquet TCP ou UDP (couche 4), ni quelle application est concernée (couche 5). Cela permet de garder les routeurs *simples et rapides*, et de préserver la confidentialité du contenu transporté.

2 Routage et algorithme BFS

a) L'algorithme BFS depuis F : on commence par marquer F à la distance 0, puis on marque ses voisins directs (E et C) à la distance 1, leurs voisins non encore marqués (D, B) à la distance 2, et enfin A à la distance 3 :



b) Pour atteindre une destination, le nœud A relaie le paquet à son voisin direct dont la distance à la destination est la plus petite. En refaisant l'analyse BFS pour chaque destination, on obtient :

A		
dest	dir	dist
B	B	1
C	B	2
D	D	1
E	B ou D	2
F	B ou D	3

(Pour E, on peut passer aussi bien par B (A–B–E) que par D (A–D–E). Même chose pour F : A–B–C–F, A–B–E–F ou A–D–E–F sont tous de longueur 3.)

c) Si le lien B–E disparaît, les chemins via B–E ne sont plus disponibles. On refait BFS dans le graphe modifié :

- La distance de A à B reste 1 (lien direct A–B).
- La distance de A à C reste 2 (via B).
- La distance de A à D reste 1 (lien direct A–D).

- Pour E, le seul chemin court restant est A-D-E : distance 2, direction **D uniquement**.
- Pour F, on peut passer par A-D-E-F ou par A-B-C-F : distance 3, directions **B ou D**.

Concrètement, la seule entrée qui change est celle de E : direction **D** au lieu de « B ou D ». Pour F, les deux directions restent valides.

3 Protocole TCP : ACK, time-out et retransmission

a) Les trois scénarios sont :

1. Le paquet envoyé par A s'est **perdu ou détérioré** en chemin (et n'est jamais arrivé à B).
2. Le paquet est arrivé correctement à B, mais l'**ACK** envoyé par B en réponse s'est *lui aussi* perdu ou détérioré.
3. Le paquet (ou l'ACK) est resté **bloqué** dans le réseau à cause d'un problème de **congestion** et arrive trop tard, c'est-à-dire après l'échéance du time-out T .

Dans tous ces cas, A ne reçoit pas la notification à temps et conclut (à tort ou à raison) qu'il faut retransmettre.

b) Compromis sur le choix de T :

- Si T est *trop grand* : la plupart des ACK arrivent avant l'expiration, ce qui évite des retransmissions inutiles, mais A **attend trop longtemps** entre deux envois. Résultat : le **réseau est sous-utilisé**, le débit effectif est faible.
- Si T est *trop petit* : A envoie de nouveaux paquets très rapidement, sans avoir le temps de recevoir les ACK. Beaucoup de retransmissions *inutiles* surviennent (les paquets étaient en réalité en route), ce qui **aggrave la congestion** et peut conduire à un effondrement du réseau.

C'est précisément pour trouver un bon compromis qu'on utilise un algorithme de type AIMD, qui ajuste dynamiquement T (ou de manière équivalente le débit $W = 1/T$) en fonction des conditions du réseau.

c) Chaque paquet TCP porte dans son en-tête un **numéro de séquence**. Si B reçoit deux fois un paquet portant le même numéro, il sait qu'il s'agit d'un doublon et peut l'ignorer (après avoir tout de même renvoyé un ACK, pour signaler à A que la réception a bien eu lieu).

4 Algorithme AIMD et autres

L'idée, dans chaque cas, est de suivre l'évolution du point (W_1, W_2) dans le plan : une **répartition équitable** correspond à un point situé sur la diagonale $W_1 = W_2$. On part d'un point initial légèrement en-dessous de cette diagonale (par exemple, A_1 a un léger avantage). Les phases sans problème correspondent à une « flèche verte » d'augmentation, les phases avec problème à une « flèche rouge » de retrait.

a) **AIAD** : L'augmentation est $(W_1, W_2) \rightarrow (W_1 + a, W_2 + a)$: le point se déplace parallèlement à la diagonale à 45° , donc s'éloigne ou se rapproche de la diagonale *de la même distance euclidienne*. Le retrait $W \rightarrow W - \min(W, b)$ se fait également symétriquement sur les deux coordonnées. Au final, la **distance à la diagonale ne diminue pas**, et A_1 et A_2 restent figés dans leurs proportions initiales.

b) **MIMD** : Augmentation $(W_1, W_2) \rightarrow (aW_1, aW_2)$ et retrait $(W_1, W_2) \rightarrow (bW_1, bW_2)$: dans les deux cas, le point se déplace *le long de la droite passant par l'origine et le point courant*. Le rapport W_2/W_1 ne change **jamais**, donc la répartition initiale (inégle) est gelée pour toujours.

c) **MIAD** : Augmentation multiplicative (le long de la droite vers l'origine, donc le point s'éloigne de l'origine en *maintenant* le même rapport W_2/W_1), suivie d'un retrait additif (qui retire la même quantité sur les deux axes). Comme l'étape multiplicative *amplifie* l'écart entre W_1 et W_2 (en valeur absolue : si $W_1 > W_2$, alors $aW_1 - aW_2 = a(W_1 - W_2) > W_1 - W_2$) et que le retrait additif ne réduit pas cet écart, le point **s'éloigne progressivement de la diagonale**. La situation est même pire qu'au départ : l'utilisateur déjà avantageé accapare de plus en plus la bande passante.

Conclusion : seul AIMD garantit la convergence vers la diagonale $W_1 = W_2$. L'augmentation *additive* (qui se fait « à 45° ») combinée au retrait *multiplicatif* (qui ramène le point *vers l'origine* le long de la droite courante, et donc *vers la diagonale*) est l'ingrédient clé pour obtenir l'équité *et* l'efficacité.

5 Questions d'examens passés

a)

Concernant le modèle en couches TCP/IP, quelles affirmations sont **vraies** ?

- ✓ Chaque couche ajoute (à l'envoi) son propre en-tête au paquet, et le retire (à la réception) pour transmettre la charge utile à la couche supérieure.
- C'est la couche Transport qui décide du chemin pris par les paquets à travers le réseau.
- ✓ L'adresse MAC est utilisée par la couche Liaison et n'a de sens qu'à l'échelle locale (au sein d'un même réseau).
- Un routeur intermédiaire dépaquette le message jusqu'à la couche Application pour pouvoir décider de la direction à emprunter.

Solution

1. **Vrai.** C'est le principe même de l'encapsulation : à l'émission, chaque couche enveloppe la charge utile reçue de la couche supérieure avec un en-tête qui lui est propre ; à la réception, le même en-tête est retiré par la couche correspondante.
2. **Faux.** C'est la couche **Réseau** (couche 3, protocole IP) qui s'occupe du routage des paquets. La couche Transport s'occupe de la communication *de bout en bout* entre processus (TCP, UDP).
3. **Vrai.** L'adresse MAC ne sert qu'au sein d'un réseau local. Pour acheminer un paquet entre deux réseaux différents, on utilise l'adresse IP, qui a une portée globale.
4. **Faux.** Un routeur ne remonte que jusqu'à la couche **Réseau** (couche 3) pour lire l'adresse IP de destination ; il n'a pas besoin (et ne doit pas avoir besoin) de regarder le contenu transporté.

b)

Dans l'architecture TCP/IP, c'est la couche Transport qui décide du chemin pris par les paquets à travers le réseau.

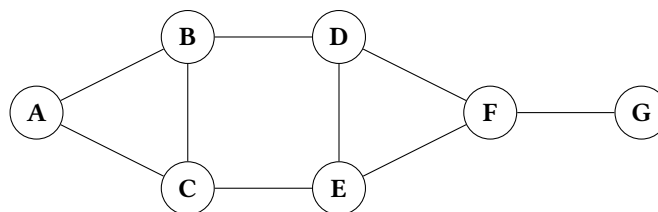
- VRAI
- ✓ FAUX

Solution

FAUX. C'est la couche **Réseau** (couche 3, protocole IP) qui s'occupe du routage en lisant l'adresse IP de destination dans l'en-tête IP et en consultant la table de routage. La couche Transport (TCP, UDP) gère quant à elle la communication *de processus à processus* (numéros de port, ACK, retransmission, contrôle de congestion).

c)

On considère le réseau suivant, dans lequel deux nœuds reliés par une arête sont séparés d'une distance 1 :



Parmi les affirmations suivantes, lesquelles sont **vraies** ?

- ✓ La distance de A à G dans ce réseau vaut 4.
- Dans la table de routage de A, l'entrée pour la destination D admet deux directions possibles (B ou C).
- ✓ Si le nœud F tombe en panne, il n'existe plus aucune route pour atteindre G depuis le reste du réseau.
- ✓ Si on ajoute un lien direct entre C et F, la distance de A à G dans la nouvelle table de routage devient 3.

Solution

1. **Vrai.** Plus court chemin : A-B-D-F-G (ou A-C-E-F-G), de longueur 4.
2. **Faux.** Le seul chemin de longueur minimale de A à D est A-B-D (longueur 2). Passer par C donnerait au mieux A-C-B-D ou A-C-E-D, soit une longueur 3. Donc dans la table de routage de A, l'entrée pour D a une **unique** direction, qui est B.
3. **Vrai.** G n'est relié qu'à F dans ce réseau. Si F tombe, G est isolé.
4. **Vrai.** Avec le lien C-F, le chemin A-C-F-G a longueur 3, et c'est le plus court possible.

d)

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

A		
dest	dir	dist
C	B	2
F	F	1

C		
dest	dir	dist
E	D	2
A	B	2

F		
dest	dir	dist
B	A	2
A	A	1

Déduisez les valeurs de x et y dans la table de routage de E :

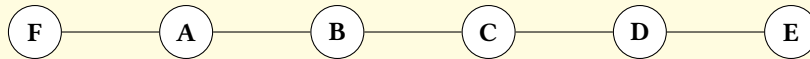
E		
dest	dir	dist
A	x	y

Solution

A. On déduit les liens du réseau à partir des tables :

- Table de A : « pour aller à C, prendre B, distance 2 » \Rightarrow A est relié à B, et B à C (en revanche, A n'est pas directement relié à C). « Pour aller à F, prendre F, distance 1 » \Rightarrow A est directement relié à F.
- Table de C : « pour aller à E, prendre D, distance 2 » \Rightarrow C est relié à D, et D à E (mais C n'est pas directement relié à E). « Pour aller à A, prendre B, distance 2 » confirme A-B-C.
- Table de F : « pour aller à B, prendre A, distance 2 » \Rightarrow F-A-B (cohérent avec ce qu'on a déjà).

Le réseau le plus simple compatible avec toutes ces tables est donc le chemin :



B. Dans ce graphe linéaire, pour aller de E à A, on doit traverser E-D-C-B-A. Donc :

$$x = D, \quad y = 4.$$