*Université Paris Diderot*
M1 informatique
Protocoles réseaux 2013-2014

Examen:
Mardi 7 janvier 2014
15h30-18h30

*Aucun document n'est autorisé. Tous les exercices sont indépendants.*

**Exercice 1.**— Répondre briévement aux questions suivantes

1. Donner un exemple de protocole de la couche liaison de données.
2. Donner un exemple de protocole la couche réseau.
3. Donner un exemple de protocole la couche transport.
4. Vous avez un ordinateur portable que vous utilisez à l'université et chez vous. A-t-il toujours la même adresse ethernet? A-t-il toujours la même adresse internet?
5. Une machine $A$ a pour adresse 130.10.1.1. Si $B$ a pour adresse 130.10.1.3, quels protocoles sont utilisés quand $A$ veut envoyer une trame IP à la machine $B$ dont il ne connait que l'adresse Internet.
6. Une machine $A$ a pour adresse 130.10.1.1. Si $B$ a pour adresse 131.11.1.3. Quels protocoles sont utilisés quand $A$ veut envoyer une trame IP à la machine $B$ dont il connait l'adresse Internet.
7. Une machine $A$ envoie un datagramme IP à une machine $B$ d'une taille inférieure à la taille maximale d'un datagramme IP. Ce datagramme peut il être fragmenté? A quel moment ? Quand ce datagramme est-il réassemblé? Quels sont les éléments qui permettent de le reassembler?
8. Lors d'une connexion TCP, tous les segments suivent-ils le même chemin physique entre la source et la destination?
9. Le champ *checksum* de l'en-tête du segment TCP permet-il de corriger des erreurs de transmission?
10. Dans quels cas l'émetteur réémet il un segment TCP?
11. Lors d'un échange entre deux sites par UDP, tous les segments suivent-ils le même chemin physique entre la source et la destination?
12. Dans quel cas l'émetteur réémet il un segment UDP?

**Exercice 2.**— On considère un alphabet $A$ et un alphabet $B$. Soit $C$ une application qui associe à chaque lettre de $A$ un mot sur $B$. On étend $C$ naturellement aux mots sur $A$.

1. Soit $m$ un mot sur $B$. Si pour les lettres $x$ de $A$, $C(x)$ est de longeur 3, est il possible de trouver deux mots différents $\alpha$ et $\beta$ sur $A$ tels que $C(\alpha) = C(\beta) = m$
2. Pour $A = \{a, b, c, d, e\}$ et $B = \{0, 1\}$, donnez un exemple de $C$ et de mots $m$ et $m'$ sur $B$ tels que:
   (a) il n'existe aucun mot $\alpha$ sur $A$ tel que $C(\alpha) = m$.
   (b) il existe deux mots différents $\alpha$ et $\beta$ sur $A$ tels que $C(\alpha) = C(\beta) = m'$.
3. Connaissez vous un algorithme qui permettent de déterminer si $C$ est un code? L'appliquer à votre exemple de la question précédente.

4. Pour $A = \{a, b, c, d, e\}$ et $B = \{0, 1\}$, donnez un exemple où $C$ est un code de longueur variable i.e. il y a au moins 2 lettres de $A$, $x$ et $y$, pour lesquelles la longueur de $C(x)$ est différente de celle de $C(y)$ (vous devez justifier le fait que votre exemple donne un code).

**Exercice 3.**— On considére un réseau représenté par le graphe $N =< S, A >$, $S$ étant l'ensemble des noeuds du réseau et $A$ l'ensemble des arêtes ($(x, y) \in A$ si et seulement si les noeuds $x$ et $y$ sont voisins).

Soit $c(x, y)$ le coût associé à l'arête $(x, y)$ dans le réseau. Par hypothése on suppose que (a) si $x$ et $y$ ne sont pas voisins, $c(x, y) = \infty$ (b) si $x \neq y$ alors $c(x, y) > 0$ et (c) $c(x, y) = c(y, x)$.

Soit un chemin $C$ de $x_0$ à $x_n$: $C = (x_0, x_1)(x_1, x_2) \dots (x_{n-1}, x_n)$ ($C$ est un chemin d'arêtes successives $(x_i, x_{i+1})$), le coût du chemin $C$ est $c(C) = \Sigma_{i=0}^{i=n-1} c(x_i, x_{i+1})$.

On note $d(x, y) = \min\{c(C) | C \text{ chemin de } x \text{ à } y\}$ la distance de $x$ à $y$. On rappelle que l'on a (Bellman-Ford):

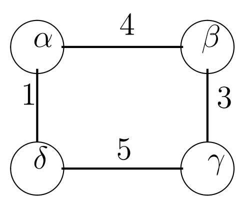
$$d(x, y) = \min\{c(x, z) + d(z, y) | z \text{ voisin de } x\} \tag{1}$$

L'algorithme suivant calcule les valeurs de $d(x, y)$ en utilisant l'équation (1). Dans cet algorithme, chaque noeud $x$ connaît $c(x, y)$ et maintient une variable (tableau) $D_x$ telle que $D_x(y)$ contient la valeur estimée de $d(x, y)$. Dans cet algorithme chaque noeud $x$ exécute:

- Initialement, $D_x(y) = c(x, y)$ et envoie $D_x$ à tous ses voisins.
- à chaque changement de $D_x$, chaque $x$ envoie à tous ses voisins la nouvelle valeur de $D_x$.
- quand $x$ reçoit de son voisin $k$ un nouveau $D_k$ ou que $c(x, k)$ est modifié, $x$ met à jour $D_x$ comme suit:

$$\forall y : D_x(y) := \min\{c(x, z) + d(z, y) | z \text{ voisin de } x\}$$

1. Soit le réseau:




   $c(\alpha, \beta) = 4$, $c(\beta, \gamma) = 3$, $c(\delta, \gamma) = 5$ et $c(\delta, \alpha) = 1$. Décrire un comportement possible de l'algorithme jusqu'à ce qu'il se stabilise (plus aucune variable n'est modifiée).

2. Cet algorithme se stabilise-t-il toujours ( donnez un contre exemple ou justifiez briévement)?

3. On veut transformer l'algorithme précédent en un algorithme qui construit une table de routage. Dans une table de routage, chaque noeud $x$ doit construire une variable (tableau) $R_x$ telle que $R_x(y)$ est un voisin de $x$ par lequel passe un plus court chemin de $x$ à $y$. Comment modifier l'algorithme précédent de façon à ce qu'il construise une table de routage?

4. En reprenant le premier exemple ci dessus, si après la stabilisation, le coût $c(\alpha, \delta)$ change et $c(\alpha, \delta) = 40$, est ce que l'algorithme se re-stabilise vers les bonnes valeurs? Combien d'itérations cela peut-il prendre?

5. Proposez un nouvel algorithme qui permet de stabiliser en moins d'itérations.

6. L'algorithme que vous avez proposé permet-il d'éliminer tous les cas de convergence lente (donnez un contre exemple ou justifiez brièvement)?