

Examen final Applications réseaux 2015

M1 MIAGE apprentissage (F. Sikora)

—2h. Documents et calculettes interdits.

Barème uniquement indicatif pouvant évoluer.

Certains exercices sont d'après le livre de Kurose & Ross, d'E. Duris ou M. Chilowicz.—

► Exercice 1. (3 points)

Anaïs (A) communique avec Benoît (B) en VoIP en temps réel. La machine de A convertit à la volée le signal analogique de la voix d'Anaïs en un flux numérique de 8 kbit/s. La machine de A regroupe ces bits dans des paquets audio de 50 octets. Dès que le paquet est rempli, il est envoyé à B. Dès que la machine de B a reçu le paquet, il est converti en analogique et est lu.

Les paquets partant de A vers B doivent emprunter une route entrecoupée de 2 routeurs (donc constituée de 3 liaisons). On note d_i , v_i et R_i la longueur, la vitesse de propagation et le taux de transmission de la liaison i , pour $i = 1, 2, 3$. On suppose qu'il n'y a pas d'autre trafic sur la liaison, donc qu'il n'y a pas de délai d'attente. Chaque routeur utilise un temps d_{proc} pour chaque paquet.

1. Expliquer (en moins de 4 lignes) comment le débit d'un flux numérique audio joue sur la qualité du son.
2. Expliquer à quoi peut être dû d_{proc} .
3. En supposant que les routeurs sont en store-and-forward, exprimer le délai total pour l'envoi d'un paquet de taille L de A vers B, en terme de d_{proc} , L , d_i , v_i et R_i .
4. Combien de temps s'écoule t-il entre le moment où le premier bit est créé chez A et est converti chez B? On suppose que $v_1 = v_2 = v_3 = 2,5 \cdot 10^8$ m/s, que $R_1 = R_2 = R_3 = 0,2$ Mbit/s, que $d_{proc} = 3$ ms et que $d_1 = 5000$ km, $d_2 = 2500$ km et $d_3 = 5000$ km. Les fractions, étant d'un niveau école primaire, sont à résoudre.
5. Même question mais maintenant $d_{proc} = 0$ et les routeurs ne sont plus en store-and-forward.

► Exercice 2. (4 points)

Voici un extrait d'une capture sur le réseau via wireshark.

No.	Time	Source	Destination	Protocol	Info
1	1.13	10.1.3.106	10.1.3.253	DNS	Standard query 0xa95b A smtp.dauphine.fr
2	1.13	10.1.3.253	10.1.3.106	DNS	Standard query response 0xa95b CNAME mta.dauphine.fr A 193.49.168.87
3	1.13	10.1.3.106	193.49.168.87	TCP	55335 > smtp [SYN] Seq=0 Win=29200 Len=0

```

4  1.13  193.49.168.87  10.1.3.106  TCP      smtp > 55335 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0
5  1.13  10.1.3.106        193.49.168.87 TCP      55335 > smtp [ACK] Seq=1 Ack=1 Win=29312 Len=0
6  1.13  193.49.168.87    10.1.3.106  SMTP    S: 220 mta.dauphine.fr ESMTP Sendmail 8.13.6 Seq=1 Ack=1 Len=43
7  1.13  10.1.3.106        193.49.168.87 TCP      55335 > smtp [ACK] Seq=1 Ack=44 Win=29312 Len=0
8  2.19  10.1.3.106        193.49.168.87 SMTP    C: HELO 10.1.3.106 Seq=1 Len=17
9  2.20  193.49.168.87    10.1.3.106  TCP      smtp > 55335 [ACK] Seq=44 Ack=18 Win=5888 Len=0
10 2.20  193.49.168.87    10.1.3.106  SMTP    S: 250 mta1.dauphine.fr Seq=44 Len=22
11 2.20  10.1.3.106        193.49.168.87 TCP      55335 > smtp [ACK] Seq=18 Ack=66 Win=29312 Len=0

```

Expliquez ce qu'il s'est passé ainsi que chaque paquet.

► **Exercice 3.** (3.5 points)

1. Rappeler ce qu'est HTTP et quel protocole de transport il utilise.
2. Quels étaient les noms des 3 versions d'HTTP jusqu'en 2014 ?
3. Quels sont les 2 types de connexions (dans le sens du champ Connection du header) pour HTTP et qu'est ce que cela implique ?
4. Quelle est le nom de la future version du protocole HTTP ?
5. Quelles peuvent être les moyens d'accélérer le protocole HTTP et qui seront utilisés dans cette future version ?

► **Exercice 4.** (1.5 points)

(Exercice du TD 2 qu'on aurait du corriger si...)

1. Quels sont les différents éléments d'une URL ?
2. Est-ce que "www" est obligatoire pour un serveur web (pourquoi / pourquoi pas) ?

► **Exercice 5.** (4.5 points)

On souhaite un petit programme qui scanne les ports UDP d'une machine donnée. Pour cela, on suppose que cette machine héberge des serveurs UDP sur des ports inconnus : l'objectif est de découvrir lesquels. Pour cela, on va envoyer vers chaque port un message UDP et attendre une réponse potentielle.

```

1 public static void main(String[] args) {
2     byte[] m = "A".getBytes("UTF-8");
3     InetAddress ia = InetAddress.getByname(args[0]);
4     DatagramSocket ds = new DatagramSocket();
5     DatagramPacket dp = new DatagramPacket(m, m.length);
6     dp.setAddress(ia);
7     for(int i=1 ; i<=N ; i++) {
8         dp.setPort(i);
9         ds.send(dp);
10    }
11    ds.close();
12 }

```

1. Expliquer ce que fait le code proposé. Par quelle valeur faut-il remplacer N ?
2. Expliquer en détail ce qu'effectue le constructeur sans argument `DatagramSocket()` (interface d'attache, port d'origine des paquets UDP...). Comparer avec le constructeur `DatagramSocket(int, InetAddress)`.
3. Décrire les possibilités d'exceptions de ce code.
4. Ajoutez à ce code la tentative de réception d'un message réponse du serveur. On suppose que le serveur répond avec un message texte en UTF-8 : on affiche le message de réponse avec le port du serveur sur la sortie standard. On introduira une temporisation pour la réception du paquet au delà de laquelle on considère qu'il n'y a pas de serveur sur ce port.

► **Exercice 6.** (5 points)

Le but de l'exercice est de proposer un serveur TCP de sondage pouvant gérer plusieurs clients à la fois. Des clients se connectent au serveur de sondage. Pour simplifier, on suppose que le serveur ne pose qu'une seule question concernant l'âge du client.

Le serveur choisit de stocker les réponses de tous les clients à la suite dans un unique fichier (dont le chemin est donné en argument de la méthode `main`). Les réponses sont sauvegardées ligne par ligne en mode texte en utilisant l'encodage UTF-8. On pourra utiliser `new OutputStreamWriter(new FileOutputStream(chemin, true), "UTF-8")` pour ouvrir le fichier en mode ajout et y ajouter une nouvelle ligne de réponse.

La NSA souhaite également pouvoir lire en temps réel les réponses. Pour cela, à l'aide d'un mot clé (que vous êtes libre de choisir) en réponse à la question, un client accède à ce mode. Lorsque ce mode est activé, le client reçoit en temps réel une copie des réponses fournies par les clients connectés au serveur (jusqu'à fermeture de la connexion). Bien sûr, la NSA possède plusieurs employés voulant utiliser ce mode simultanément.

1. Proposer un protocole décrivant les messages échangés entre les clients et le serveur.
2. Écrire en Java le code pour le serveur pouvant gérer plusieurs clients à la fois (on ne demande pas de limite sur le nombre de clients traités simultanément) et possédant le mode satisfaisant la NSA.