

Examen final Applications réseaux juin 2014

M1 MIAGE apprentissage (F. Sikora)

—1h30. Documents et calculettes interdits.

Barème uniquement indicatif pouvant évoluer.

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

► Exercice 1. (3 points)

On considère un serveur et un client séparés par un routeur. La bande passante entre le serveur et le routeur est R_s , celle entre le routeur et le client est R_c . Supposons que le serveur envoie au client deux paquets à la suite, chacun de taille L bits. On considère aucun autre trafic sur ces liaisons. La distance entre le serveur et le routeur, et entre le routeur et le client est de m mètres. La vitesse de propagation v est la même sur les deux liaisons.

1. Supposons que $R_s < R_c$. Combien de temps y a-t-il entre le moment où le dernier bit du premier paquet arrive chez le client et le moment où le dernier bit du deuxième paquet arrive chez le client ?
2. Supposons maintenant jusqu'à la fin que $R_c < R_s$. Est-il possible que le second paquet attende au niveau du routeur ? Expliquer et justifier.
3. Supposons que le serveur envoie le second paquet T secondes après avoir envoyé le premier paquet. Quelle doit-être la valeur de T pour assurer l'absence de délai d'attente au niveau du routeur ? Expliquer et justifier.

► Exercice 2. (1.5 points)

1. Si un serveur Web attend des clients TCP sur le port 80, est-il possible d'installer en même temps sur la même machine un serveur en UDP qui écoute sur le port 80 ? Justifier.
2. Deux sockets TCP peuvent-elles être attachées au même numéro de port de la même machine ? Si oui, dans quel cas ?
3. Deux sockets UCP peuvent-elles être attachées au même numéro de port de la même machine ? Si oui, dans quel cas ?

► Exercice 3. (5 points)

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

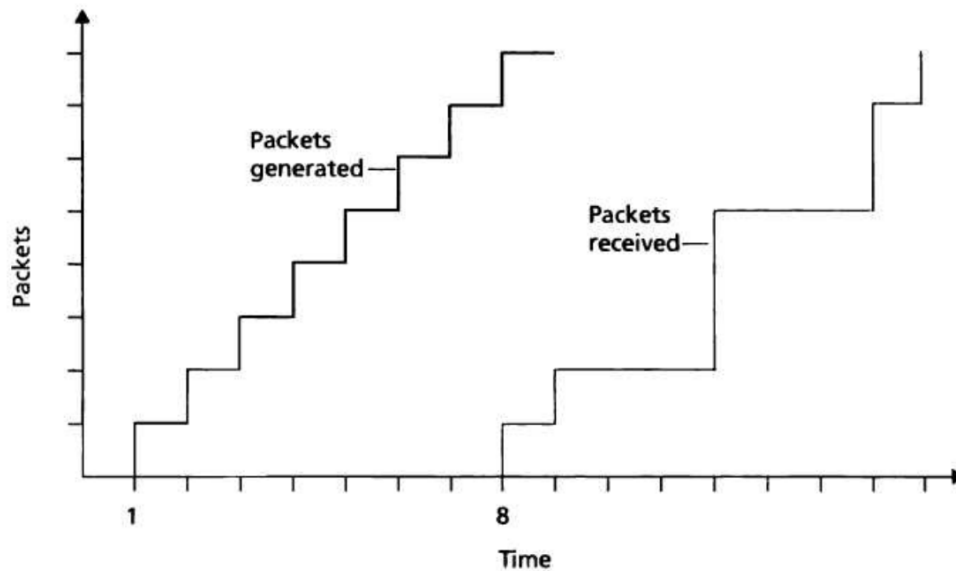
No.	Time	Source	Destination	Protocol	Length	Info
1	0.0000	Dell_5b:f5:f0	Broadcast	ARP	42	Who has 10.1.3.253? Tell 10.1.3.106
2	0.0005	Xensourc_15:27:bf	Dell_5b:f5:f0	ARP	60	10.1.3.253 is at 00:16:3e:15:27:bf
3	0.0005	10.1.3.106	10.1.3.253	DNS	80	Standard query 0xf3e A www.mido.dauphine.fr

4	0.0014	10.1.3.253	10.1.3.106	DNS	283	Standard query response 0x6f3e CNAME rweb-departements.dauphine.fr A 193.49.168.124
5	0.0015	10.1.3.106	193.49.168.124	TCP	74	57740 > http [SYN] Seq=0 Win=14600 Len=0
6	0.0023	193.49.168.124	10.1.3.106	TCP	74	http > 57740 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0
7	0.0024	10.1.3.106	193.49.168.124	TCP	66	57740 > http [ACK] Seq=1 Ack=1 Win=14720 Len=0
8	0.0066	10.1.3.106	193.49.168.124	HTTP	617	GET /fr/miage.html HTTP/1.1
9	0.0101	193.49.168.124	10.1.3.106	TCP	66	http > 57740 [ACK] Seq=1 Ack=552 Win=6912 Len=0

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

► **Exercice 4.** (2 points)

On considère la figure suivante :



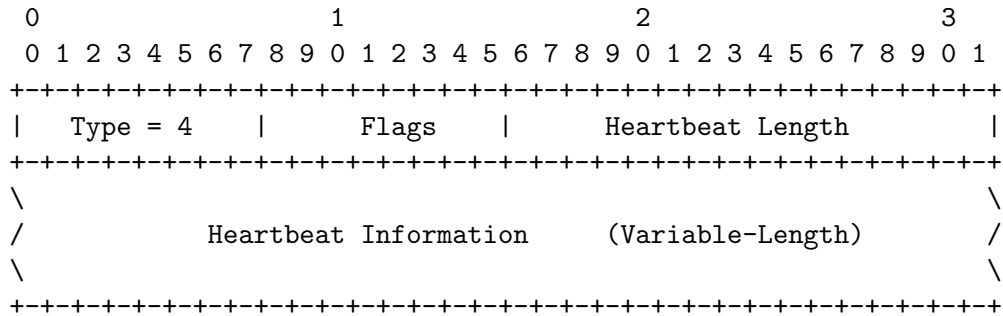
Un émetteur envoie un signal audio paquet par paquet de manière périodique à partir de $t = 1$. Le premier paquet arrive chez le destinataire à $t = 8$.

1. Quels sont les délais (de l'émetteur au destinataire) des paquets 2 à 8 ? (les lignes verticales et horizontales ont des longueurs de 1, 2 ou 3 unités de temps)
2. Si la lecture commence chez le destinataire à $t = 8$, dès le premier paquet arrivé, quels paquets seront manqués à la lecture ?
3. Même question si la lecture commence à $t = 9$.
4. Quel est le délai minimal pour que les 8 paquets envoyés soient lus à temps ?

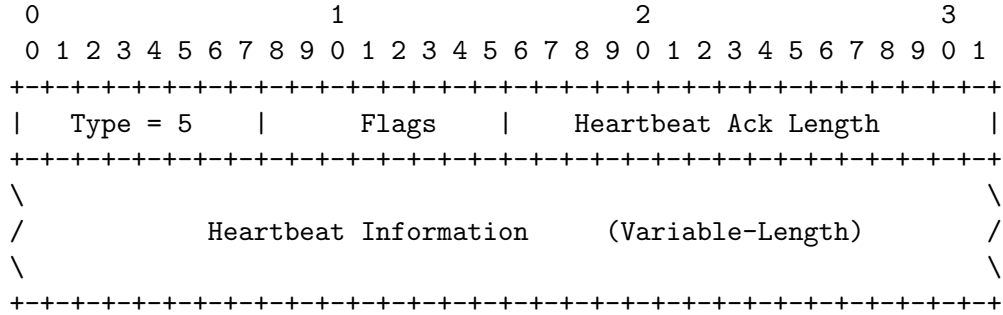
► **Exercice 5.** (1.5 points)

Heartbeat est une extension d'OpenSSL (une implémentation de SSL/TLS), permettant de garder une connexion ouverte. Elle fonctionne de la manière suivante : un des côtés de la connexion envoie un "payload" (des données) de taille arbitraire à l'autre côté, qui lui renvoie une copie de ces données pour prouver que tout fonctionne.

Voici un extrait de la RFC décrivant le protocole. Le format du message de requête est le suivant :



Le format du message de réponse est le suivant :



Dans les deux cas, il y a 1 octet utilisé pour le type de message, 1 octet pour des flags inutiles ici, 2 octets pour un champ length, et une information de taille variable.

Un problème a été détecté dans le scénario suivant : le message de requête envoie un payload de 1 octet et indique la valeur 65535 dans le champ length. Le message de réponse envoie un payload de 65535 octets et indique la valeur 65535 dans le champ length.

1. En quoi est-ce un problème ?
2. Que s'est-il passé et que faudrait-il faire pour le résoudre ?

► **Exercice 6.** (8 points)

On désire créer un proxy sous la forme d'une application Java, lancé avec une commande similaire à `java Proxy proxyPort server serverPort`. On souhaite que ce proxy puisse gérer plusieurs clients à la fois, c'est-à-dire que plusieurs clients puissent communiquer avec le port `serverPort` de la machine `server` via le proxy (leur message est transmis au serveur, les réponses du serveur leurs sont transmises). On ne suppose rien sur les informations contenues dans les paquets, ni sur l'ordre et le rythme des échanges.

1. Si l'on désire un proxy TCP, quel serait le principe de fonctionnement du proxy ? Décrire en français.

2. On suppose jusqu'à la fin de l'exercice qu'on désire maintenant un proxy gérant UDP. On ne sait toujours rien sur les communications avec le serveur, à part qu'elles ne dépassent pas 1024 octets par datagramme. Si le proxy devait gérer un unique client, quel en serait le principe de fonctionnement ?
3. On désire maintenant un proxy UDP devant gérer plusieurs clients. Quels sont les différents problèmes à prendre en compte, comment peuvent-ils être résolus (ou non), les inconvénients des solutions choisies ? (Vous pouvez utiliser plusieurs sockets).
4. Coder en Java un tel proxy UDP multclients.