

Examen final Applications réseaux 2016

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, C. Chou ou C. Fouqueré. —

► **Exercice 1.** (3 points)

Vrai ou faux (et pourquoi) :

1. Un utilisateur demande une page Web contenant du texte et 3 images. Pour cette page, il enverra 1 requête et recevra 4 réponses.
2. Ces deux pages web distinctes `http://9gag.com/trending/index.html` et `http://9gag.com/fresh/index.html` peuvent être envoyées sur la même connexion persistante.
3. La méthode `connect` de la classe `DatagramSocket` n'existe pas (si elle existe dire ce qu'elle fait, si elle n'existe pas dire pourquoi).

► **Exercice 2.** (3 points)

On considère un serveur et un client séparés par un routeur store-and-forward. 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 est de m mètres, et la distance entre le routeur et le client est de m mètres. La vitesse de propagation v est la même sur les deux liaisons.

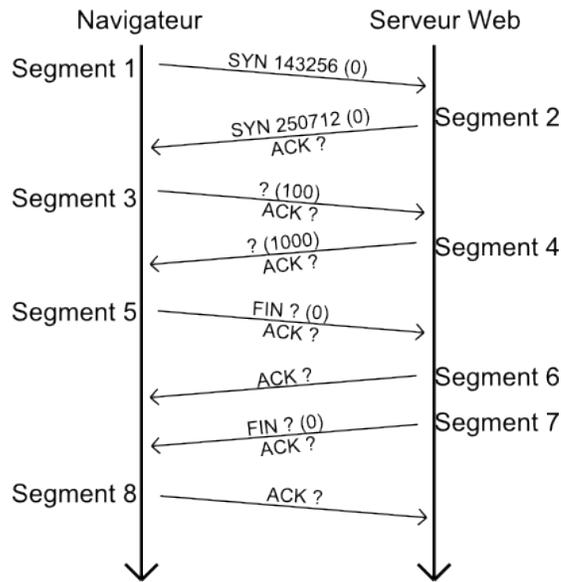
1. Supposons que $R_s < R_c$. Combien de temps y a-t-il entre le moment où le dernier bit du premier paquet arrive chez le client et le moment où le dernier bit du deuxième paquet arrive chez le client ?
2. Supposons maintenant jusqu'à la fin que $R_c < R_s$. Est-il possible que le second paquet attende au niveau du routeur ? Expliquer et prouver avec des formules.
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 3.** (3 points)

Supposons que nous avons les trois octets sur 8-bits suivants : 00101111, 10001110, 00110100.

1. Quel est l'octet stocké dans le champ checksum en UDP (ou TCP) si ces 3 octets doivent être transférés ?
2. Indiquer ce qui est effectué pour détecter une éventuelle erreur côté receveur.
3. Est-ce qu'une seule erreur est toujours détectée et pourquoi ?
4. Est-ce que deux erreurs sont toujours détectées et pourquoi ?

► **Exercice 4.** (4.5 points)

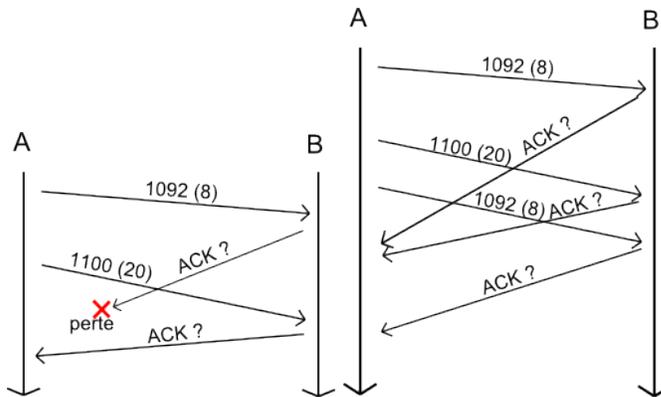


La figure ci-dessus représente les échanges lors du transfert d'une page web entre un navigateur et un serveur web. On fait l'hypothèse que la requête pour la page web fait 100 octets, que la page web fait 1000 octets, et qu'il n'y a pas d'erreur de transmission.

Pour chaque segment de données, on a indiqué si les booléens SYN, FIN ou ACK dans l'en-tête TCP étaient mis à vrai. On indique également le numéro de séquence du premier octet du segment. Le chiffre entre parenthèses correspond au nombre d'octets transmis dans le segment. Si le segment est un acquittement, ACK est mentionné suivi de la valeur du champ d'acquiescement du segment TCP.

1. Quel est le numéro de séquence du *premier* et du *dernier* octet envoyé dans le segment 3 ?
2. Donner les numéros de séquences pour chaque point d'interrogation (11 valeurs).
3. Indiquer à quoi correspondent chacun des 8 segments.

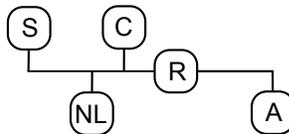
On étudie maintenant les 2 scénarios suivants entre un hôte A et un hôte B :



- 4 Dans le premier scénario, un message ACK est perdu. Donner la valeur des deux ACK envoyés par B.
- 5 Dans le deuxième scénario, le numéro de séquence pour le segment 1 et 3 de A est le même. Pourquoi, comment ?
- 6 Donner la valeur des trois ACK envoyés par B pour le scénario 2.

► **Exercice 5.** (3 points)

Dans la figure ci-dessous, A est un ordinateur de bureau, R est le routeur passerelle pour A, S est un serveur web, C est un cache web et NL est le serveur de noms local de C. L'hôte A est configuré pour utiliser C pour toutes ses requêtes.



On suppose que toutes les machines viennent juste d'être démarrées et par conséquent, tous leurs caches associés sont vides, que le contenu de `http://S/index.html` rentre dans un seul paquet, que des connexions HTTP persistantes sont utilisées (on peut considérer qu'une fois que la connexion est effectuée elle n'est jamais fermée) et que C répond aux requêtes TCP qui ont la forme de celle du paquet 2 ci-dessous (c'est-à-dire `GET http://truc/muche`).

1. L'utilisateur de A demande la page `http://S/index.html`. Le tableau ci-dessous donne certains paquets échangés pendant cette requête. Cependant, la liste est incomplète et contient des paquets qui n'ont pas été échangés. L'ordre donné est aléatoire : donner la liste ordonnée (selon l'ID) des paquets échangés sur le réseau.

ID	SRC	DST	Port SRC	Port DST	Protocole	Contenu
1	C	DNS root		DNS	UDP	Requête pour S
2	A	C		Web cache	TCP	GET http://S/index.html
3	NL	DNS root		DNS	UDP	Requête pour S
4	C	S		HTTP	TCP	SYN
5	C	S		HTTP	TCP	GET index.html
6	S	A	HTTP		TCP	index.html
7	A	Broadcast			ARP	Who is R
8	C	A	Web cache		TCP	index.html
9	NL	C	DNS		UDP	Adresse de S
10	S	C	HTTP		TCP	index.html
11	A	S		HTTP	TCP	GET index.html
12	A	S		HTTP	TCP	SYN

2. On suppose que A n'a pas de cache Web local, mais à part cela, que tout ce qui peut être mis en cache l'est (C existe toujours). Pour la même requête `http://S/index.html` de A, quels sont les messages envoyés (donner les ID) ?

► **Exercice 6.** (5 points)

On considère la classe Foo suivante, définissant un serveur. On suppose que vous disposez d'un client du type netcat utilisant l'encodage ASCII et permettant d'émuler un client (ce que vous tapez au clavier est envoyé en ASCII au serveur, et ce que le serveur vous envoie est considéré comme de l'ASCII et est affiché à l'écran).

```

1 import java.io.*;
2 import java.net.*;
3 import java.util.concurrent.*;
4
5 public class Foo {
6     private static final int MAX_THREAD = 2;
7     private static final int ASCII_CODE_FOR_B = 66;
8     private final ExecutorService executor = Executors.newFixedThreadPool(MAX_THREAD);
9     private final ServerSocket ss;
10
11     public Foo(int port) throws IOException {
12         ss = new ServerSocket(port);
13     }
14
15     public void launch() throws IOException {
16         while(true) {
17             final Socket s = ss.accept();
18             executor.submit(new Runnable() {
19                 @Override public void run() {
20                     try {
21                         BufferedReader br = new BufferedReader(new InputStreamReader(s.getInputStream(), "ASCII"));
22                         BufferedWriter bw = new BufferedWriter(new OutputStreamWriter(s.getOutputStream(), "ASCII"));
23
24                         while(!Thread.currentThread().isInterrupted()) {
25                             String line = br.readLine();
26                             if(line==null)
27                                 break;
28                             try {
29                                 int i = Integer.parseInt(line);
30                                 bw.write(i);
31                                 if (i==ASCII_CODE_FOR_B)
32                                     executor.shutdown();
33                             } catch(NumberFormatException nfe) {
34                                 String msg = nfe.toString();

```

```

35         bw.write(msg);
36     }
37     bw.newLine();
38     bw.flush();
39 }
40 } catch (IOException e) {
41     e.printStackTrace(System.err);
42 } finally {
43     try {
44         s.close();
45     } catch (IOException e) {
46         // Do nothing
47     }
48 }
49 }
50 });
51 }
52 }
53 public static void main(String[] args) throws NumberFormatException, IOException {
54     Foo server = new Foo(Integer.parseInt(args[0]));
55     server.launch();
56 }
57 }

```

1. Quel protocole de la couche transport utilise ce serveur et pourquoi pouvez-vous l'affirmer ?
 2. Supposez que vous lancez un premier client appelé C1 (comme netcat par exemple, ou nc) pour interroger ce serveur. Que se passe-t-il lorsque vous tapez "lolilol" dans ce client suivi d'un retour chariot ? Qu'affiche le client C1 ?
 3. Supposez que dans le même client C1, vous tapez ensuite "65" suivi d'un retour chariot. Que se passe-t-il ? Qu'affiche le client C1 ? (on rappelle que les codes ASCII se suivent entre A et Z)
 4. Supposez que dans le même client C1, vous tapez ensuite "B" suivi d'un retour chariot. Que se passe-t-il ? Qu'affiche le client C1 ?
 5. Supposez que vous démarrez un second client C2. Que se passe-t-il au niveau du serveur ? C2 fonctionne-t-il comme le premier client ?
 6. Supposez que vous démarrez un troisième client C3. Que se passe-t-il au niveau du serveur ? C3 fonctionne-t-il comme les deux premiers clients ?
 7. Supposez que vous fermez le flot de lecture clavier du client C1 (par Ctrl-D par exemple). Que se passe-t-il au niveau du serveur ? Que se passe-t-il au niveau du client C1 ? Que se passe-t-il au niveau des clients C2 et C3 ?
 8. Supposez que dans le client C2, vous tapez maintenant "66" suivi d'un retour chariot. Que se passe-t-il au niveau du serveur ? Que se passe-t-il au niveau des clients C2 et C3 ?
 9. Supposez que vous démarrez un quatrième client C4. Que se passe-t-il au niveau du serveur ? Que se passe-t-il au niveau des clients C2, C3 et C4 ?
-
-

Annexes

Extraits de la RFC de TCP

2.6. Reliable Communication

A stream of data sent on a TCP connection is delivered reliably and in order at the destination.

Transmission is made reliable via the use of sequence numbers and acknowledgments. Conceptually, each octet of data is assigned a sequence number. The sequence number of the first octet of data in a segment is transmitted with that segment and is called the segment sequence number. Segments also carry an acknowledgment number which is the sequence number of the next expected data octet of transmissions in the reverse direction. When the TCP transmits a segment containing data, it puts a copy on a retransmission queue and starts a timer ; when the acknowledgment for that data is received, the segment is deleted from the queue. If the acknowledgment is not received before the timer runs out, the segment is retransmitted.

Source Port : 16 bits

The source port number.

Destination Port : 16 bits

The destination port number.

Sequence Number : 32 bits

The sequence number of the first data octet in this segment (except when SYN is present). If SYN is present the sequence number is the initial sequence number (ISN) and the first data octet is ISN+1.

Acknowledgment Number : 32 bits

If the ACK control bit is set this field contains the value of the next sequence number the sender of the segment is expecting to receive. Once a connection is established this is always sent.

Control Bits : 6 bits (from left to right) :

URG : Urgent Pointer field significant ACK : Acknowledgment field significant PSH : Push Function RST : Reset the connection SYN : Synchronize sequence numbers FIN : No more data from sender

Extraits de Javadoc

ExecutorService

An ExecutorService can be shut down, which will cause it to reject new tasks. Two different methods are provided for shutting down an ExecutorService. The `shutdown()` method will allow previously submitted tasks to execute before terminating, while the `shutdownNow()` method prevents waiting tasks from starting and attempts to stop currently executing tasks.

BufferedWriter

```
public class BufferedWriter extends Writer
```

Writes text to a character-output stream, buffering characters so as to provide for the efficient writing of single characters, arrays, and strings.

```
public void write(int c) throws IOException
```

Write a single character.

Parameters : c - int specifying a character to be written.

```
public void write(String str) throws IOException
```

Write a string.

Parameters : str - String to be written