

Mise au propre des TDs de calculabilité

Yann Miguel

13 février 2020

Table des matières

1	TD1	2
1.1	Exercice 1	2
1.1.1	Question 1	2
1.1.2	Question 2	2
1.1.3	Question 3	2
1.1.4	Question 4	2
1.2	Exercice 2	2
1.2.1	Question 1	2
1.2.2	Question 2	2
1.3	Exercice 3	3
1.3.1	Question 1	3
1.3.2	Question 2	3
1.3.3	Question 3	3
1.3.4	Question 4	3
1.4	Exercice 4	3
2	TD2	4
2.1	Exercice 1	4
2.1.1	Question 1	4
2.1.2	Question 2	4
2.2	Exercice 2	4
2.2.1	Question 1	4
2.2.2	Question 2	4
2.2.3	Question 3	4
2.3	Exercice 3	4
2.3.1	Question 1	4
2.3.2	Question 2	4
2.3.3	Question 3	5
2.3.4	Question 4	5
3	TD3	6
3.0.1	Question 1	6
3.0.2	Question 2	6
3.0.3	Question 3	7
3.0.4	Question 5	7

1 TD1

1.1 Exercice 1

1.1.1 Question 1

L'intérêt du modèle en couche est la facilité d'abstraction et l'isolation des couches.

1.1.2 Question 2

Le modèle de couche TCP/IP contient 4 couches:

1. Application
2. Transport
3. Internet
4. NetworkAccess

1.1.3 Question 3

Les deux modèles sont des modèles à couches, mais le TCP/IP en contient moins.

Couches du modèle OSI:

1. Application
2. Présentation
3. Session
4. Transport
5. Network
6. DataLink
7. Physical

1.1.4 Question 4

La couche permettant de faire abstraction de la manière dont les différents réseaux locaux sont interconnectés est la couche Network pour le modèle OSI et la couche Internet pour le modèle TCP/IP.

1.2 Exercice 2

1.2.1 Question 1

Protocoles de la couche DataLink:

- WiFi
- Ethernet
- Bluetooth

1.2.2 Question 2

Le rôle de la couche DataLink est d'assurer la communication entre plusieurs machines connectées sur un même réseau local.

1.3 Exercice 3

1.3.1 Question 1

Le protocole Ip appartient à la couche Internet.

1.3.2 Question 2

Afin de connaître la destination d'un paquet, on regarde l'adresse dans la paquet.

1.3.3 Question 3

La principale amélioration de L'IPv6 sur l'IPv4 est la quantité d'adresses disponibles.

1.3.4 Question 4

On a choisi des adresses de 128 bits pour deux choses:

1. La quantité d'adresses disponibles.
2. La généralisation des adresses à partir de l'adresse Mac.

1.4 Exercice 4

$$\frac{vc}{d} < t \Rightarrow \frac{36}{d} < 1,2 \Rightarrow d < \frac{5 \times 16000}{2,4} \Rightarrow d < \frac{400000}{12} \Rightarrow d < 33333,333 \text{ m}$$

Cette solution prends en compte le retour du disque vers le point de départ.

Si on compte un allé simple, on fait $33,333 \times 2 = 66,67 km$, ou bien 66666,67 m.

2 TD2

2.1 Exercice 1

2.1.1 Question 1

Afin de représenter l'adresse de ce serveur, on utilise la classe `InetAddress` de Java.

2.1.2 Question 2

```
public static InetAddress getInetAddressByName(String hostname){
    return InetAddress.getByname(hostname);
}
```

2.2 Exercice 2

2.2.1 Question 1

Les classes importantes pour UDP sont `DatagramSocket` et `DatagramPacket`.

2.2.2 Question 2

Afin de créer un `DatagramPacket`, un client à besoin de:

- Le buffer.
- La taille du buffer.
- L'adresse du serveur.
- Le port d'écriture.

2.2.3 Question 3

```
public static void receiveTextFromUdp(){
    DatagramSocket s = new DatagramSocket(ssss);
    byte msg[] = new byte[10];
    DatagramPacket n = new DatagramPacket(msg, 40);
    s.receive(n);
    System.out.println(new String(msg));
    s.close();
}
```

2.3 Exercice 3

2.3.1 Question 1

La différence entre les transmission UDP et TCP est que UDP envoie un flux unidirectionnel et plus rapides, mais a des pertes, alors que le TCP est plus lent mais n'a aucune perte.

2.3.2 Question 2

Les données UDP se transmettent plus vite.

2.3.3 Question 3

```
Socket s = new Socket(adresse, port);
ObjectOutputStream output = new ObjectOutputStream(s.getOutputStream());
ObjectInputStream input = new ObjectInputStream(s.getInputStream());
input.read();
output.write();
s.close();
```

2.3.4 Question 4

```
public static void receiveMessageTCP(){
    Socket s = new Socket(myServer, ssss);
    InputStream in = s.getInputStream();
    BufferedReader b = new BufferedReader(new InputStreamReader(in));
    System.out.println(b.readLine());
    s.close();
}
```

3 TD3

3.0.1 Question 1

L'unité consommée n'est pas nécessaire pour le SYN, on le fait plus par habitude, mais elle l'est pour le FIN.

3.0.2 Question 2

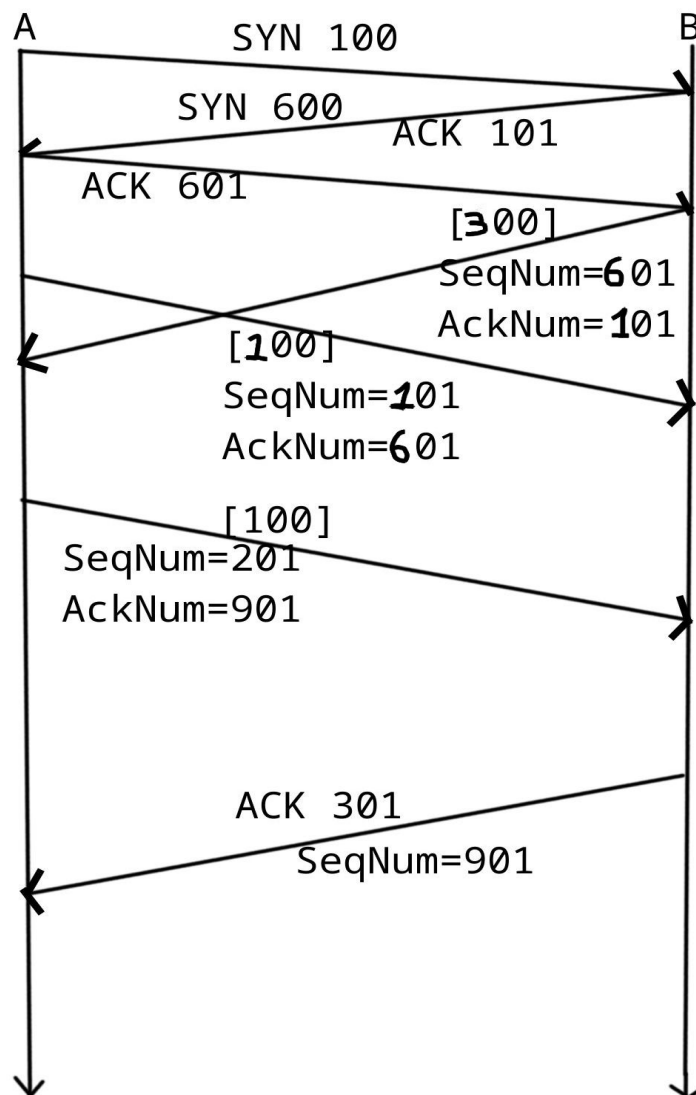


FIGURE 1 – Chronogramme d'échange entre A et B

Si le premier segment de données de A se perd, alors l'AckNum de B contiendra 100 de moins que le SeqNum de A, ce qui pousse B à envoyer un ACK pour indiquer qu'il manque le packet à partir de 101.

3.0.3 Question 3

Résumé du chronogramme:

- Envoie d'abord de 'a', attente d'ACK/tampon plein.
- Envoie "bcde", attente d'ACK/tampon plein.
- Envoie "fghi".

Si la connection est de type "TelNet full duplex", les observations sont les mêmes.

3.0.4 Question 5

Elle va grandir exponentiellement jusqu'à la moitié de la taille de la fenêtre avant le crash, 9 Ko dans ce cas.

Elle sera donc de la taille du seuil au bout de 4 transmissions acquittées, si tout se passe bien.

Après le seuil, au lieu de faire une croissance exponentielle, on incrémente seulement la quantité d'octets qui passent.