Université Paris Diderot – Paris 7 **Programmation réseau**
L3 Informatique **Anne 2010-2011, 2ème semestre**

TP n°5

Client et serveur UDP en Java

1 Un protocole similaire à SNTP

Dans ce TP, le but est de programmer un client et un serveur UDP en Java similaire au protocole NTP vu au TP précédent.

1.1 Présentation du protocole

Le client de notre protocole envoie au serveyr une requête sous forme d'un paquet UDP qui contient l'adresse IP du client ainsi que la date à laquelle le message est transmis. Le serveyr quant à lui renvoie une réponse qui contient trois dates :

- la date à laquelle la requête a été transmise (selon le client) ;
- la data à laquelle la requête a été reçue (selon le serveur) ;
- la date à laquelle la réponse a été transmise (selon le serveur) ;

On remarquera que la réponse contient la date que le client avait incluse dans la requête ; ceci permet au client d'identifier une réponse comme allant de pair avec une requête donnée, et donc de contourner les problèmes dûs à la nature non-fiable du transport.

1.2 Format des données

Les dates transmises sont codés sur 8 octets (6 bits) et correspondent au temps en miliseconde écoulé depuis le er Janvier 970.

Les requêtes et les réponses de notre protocole ont alors le format suivant :

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
Client Identifier
Server Identifier
Originate Timestamp (6)
Receive Timestamp (6)
Transmit Timestamp (6)

Les champs d'un tel message sont définis comme suit :

- *Client identifier* contient l'adresse IP du client ;
- *Server identifier* vaut 0 pour une requête et contient l'adresse IP du serveur pour une réponse ;
- *Originate Timestamp* vaut 0 pour une requête, et, pour une réponse, est la copie du *Transmit Timestamp* de la requête correspondante ;
- *Receive Timestamp* vaut 0 pour une requête, et pour une réponse, est la date de réception de la requête correspondate ;
- *Transmit Timestamp* est la date de transmission de ce paquet.

2 Exercices

Exercice 1 [Implémentation du protocole]
Écrivez en Java un client et un serveur pour ce protocole et testez-les.

2.1 Quelques indications

Voilà quelques pistes en vrac pour vous aidez à faire ce tp.

1. Pour la programmation UDP en Java vous trouverez les informations utiles dans le polycopié disponible sur la page web du cours.
2. Pour les dates, vous pouvez utiliser la classe `java.util.GregorianCalendar`
3. Voilà un exemple de code pour imprimer une date proprement avec en premier le jour, le mois et l'année et ensuite l'heure, les minutes, secondes et milisecondes :

```
import java.util.*;
import java.text.*;

public static void int main(String []argv) {
  GregorianCalendar gc= new GregorianCalendar();
  SimpleDateFormat sf=new SimpleDateFormat("dd.MM.yyyy HH:mm:ss:SS");
  System.out.println(sf.format(gc.getTime()));
}
```

4. Voilà un exemple de code permettant de traduire un entier long dans un tableau de 8 octets et un tableau de 8 octets dans un entier long :

```
public static void int main(String []argv) {
  long a=5' ;
  byte []tab=new tab[8];
  tab[7]=(byte)(l>>0);
  tab[6]=(byte)(l>>8);
  tab[5]=(byte)(l>>'6);
```

```
    tab[ ]=(byte)(l>>  );
    tab[ ]=(byte)(l>>  );
    tab[ ]=(byte)(l>> 0);
    tab[ ]=(byte)(l>> 8);
    tab[0]=(byte)(l>>56);

    long c=
      (long)(0xff & tab[0]) << 56 |
      (long)(0xff & tab[ ]) <<  8 |
      (long)(0xff & tab[ ]) <<  0 |
      (long)(0xff & tab[ ]) <<  |
      (long)(0xff & tab[ ]) <<  |
      (long)(0xff & tab[5]) << 6 |
      (long)(0xff & tab[6]) << 8  |
      (long)(0xff &mtab[7]) << 0;
}
```