



ESISAR

NE442 - Programmation répartie

Examen de TP – TIERS TEMPS

Table des matières

1 Consignes générales.....	1
2 Exercice 1 : Port scanning (10 points).....	2
2.1 Présentation générale du problème.....	2
2.2 Préparation de votre machine.....	2
2.3 Ecriture du programme.....	2
3 Exercice 2 : Réalisation d'un client TCP (10 points).....	3
3.1 Présentation générale du problème.....	3
3.2 Préparation de votre machine.....	4
3.3 Ecriture du programme PontTCPTaille.....	4

1 Consignes générales

Ce sujet est spécifique aux étudiants bénéficiant d'un tiers temps.

La durée de cet examen de TP est de **1H30**.

Vous devez impérativement travailler sur la station fournie (pas de PC personnel).

Attention : vous devez absolument respecter les consignes suivantes :

- un projet Eclipse par exercice
- le nom du projet Eclipse doit être **examen_exo1_[votrenom]** pour l'exercice 1, **examen_exo2_[votrenom]** pour l'exercice 2, ...
- vous devez rendre vos travaux en suivant les instructions données par le serveur d'examen dont l'URL est indiquée au tableau.

L'accès à Internet n'est pas autorisé, tous les documents sont autorisés (documents papiers + une clé USB).

2 Exercice 1 : Port scanning (10 points)

2.1 Présentation générale du problème.

Un serveur UDP est lancé sur une machine, vous savez que ce serveur écoute en UDP sur un port compris entre 30 000 et 32 000.

Ce serveur fonctionne ainsi :

- il écoute sur un port X , X étant compris entre 30 000 et 32 000
- il attend un paquet UDP contenant le texte `hello`
- quand il reçoit ce paquet avec `hello`, il répond `ok`.

L'objectif de cet exercice est de réaliser un programme capable de déterminer ce port X et de l'afficher.

2.2 Préparation de votre machine.

Pour pouvoir tester votre programme, vous allez installer sur votre machine ce serveur UDP.

Pour cela, télécharger le fichier **port-scanning.jar** disponible sur le serveur d'examen.

Lancer ce serveur UDP avec la commande

```
java -jar port-scanning.jar
```

Vous avez maintenant un serveur UDP écoutant sur un port X.

2.3 Ecriture du programme

Ecrivez maintenant une classe ClientScanner. Si on exécute le main de cette classe, on obtient l'affichage suivant :

```
Début du scanning des ports UDP de 30 000 à 32000 sur la machine  
"127.0.0.1"
```

```
Le serveur UDP écoute sur le port X = 31 500
```

```
Fin du programme
```

Cet affichage devra se faire dans un délai inférieur à 1 minute.

Dans un commentaire tout en haut de cette classe, vous indiquerez

- la valeur de X que vous avez trouvé
- la logique utilisée par votre programme

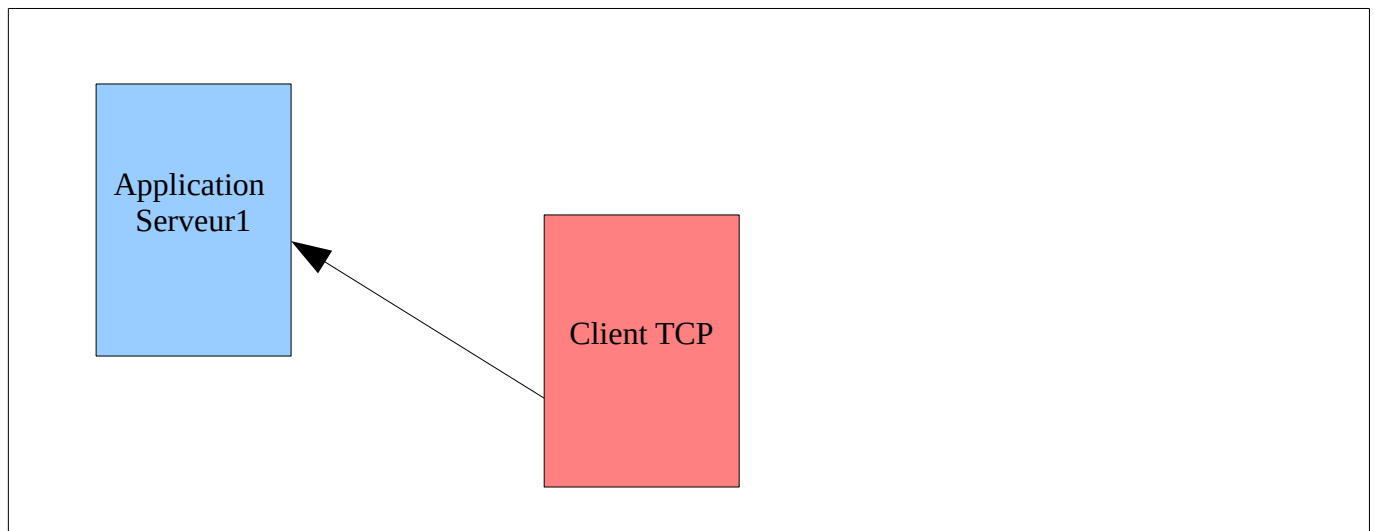
L'adresse IP de la machine scannée (ici "127.0.0.1") sera écrite en dur dans votre programme.

Attention : votre programme doit fonctionner correctement même si le serveur UDP est distant (c'est à dire si on remplace 127.0.0.1 par une autre IP).

3 Exercice 2 : Réalisation d'un client TCP (10 points)

3.1 Présentation générale du problème.

Vous allez devoir réaliser un client TCP.



L'application 1 est un serveur TCP, qui écoute sur le port 8000. Ce serveur 1 fonctionne ainsi :

- quand une demande de connexion arrive, il envoie au client un fichier d'environ 1 GigaOctet. Quand le serveur arrive à la fin du fichier, il coupe la connexion.

3.2 Préparation de votre machine.

Pour pouvoir tester votre programme, vous allez installer sur votre machine ce serveur TCP.

Pour cela, télécharger le fichier **pont-tcp.jar** disponible sur le serveur d'examen.

Lancer ce serveur avec la commande

```
java -jar pont-tcp.jar
```

Vous avez maintenant

- un serveur TCP 1 écoutant sur le port 8000 fournissant un fichier

3.3 Ecriture du programme PontTCPTaille

Ecrivez maintenant une classe PontTCPTaille. Si on exécute le main de cette classe, on obtient la taille du fichier fourni par le serveur 1.

Par exemple, si on exécute le main de cette classe, on obtient l'affichage suivant :

```
Début de la connexion au serveur 1
```

```
La taille du fichier envoyé par le serveur 1 est : 1000222000 octets
```

```
Fin du programme
```