



ESISAR

NE441 - Utilisation des couches transport

Examen blanc

Table des matières

1	Consignes générales.....	1
2	Exercice 1 : Ping Pong TCP (20 points).....	2
2.1	Présentation générale du problème.....	2
2.2	Préparation de votre machine.....	3
2.3	Ecriture du programme PingPongTCP.....	3
2.4	Test de votre programme PingPongTCP.....	4

1 Consignes générales

La durée de cet examen blanc de TP est de 50 **minutes**

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

Attention : vous devez absolument respecter les consignes suivantes :

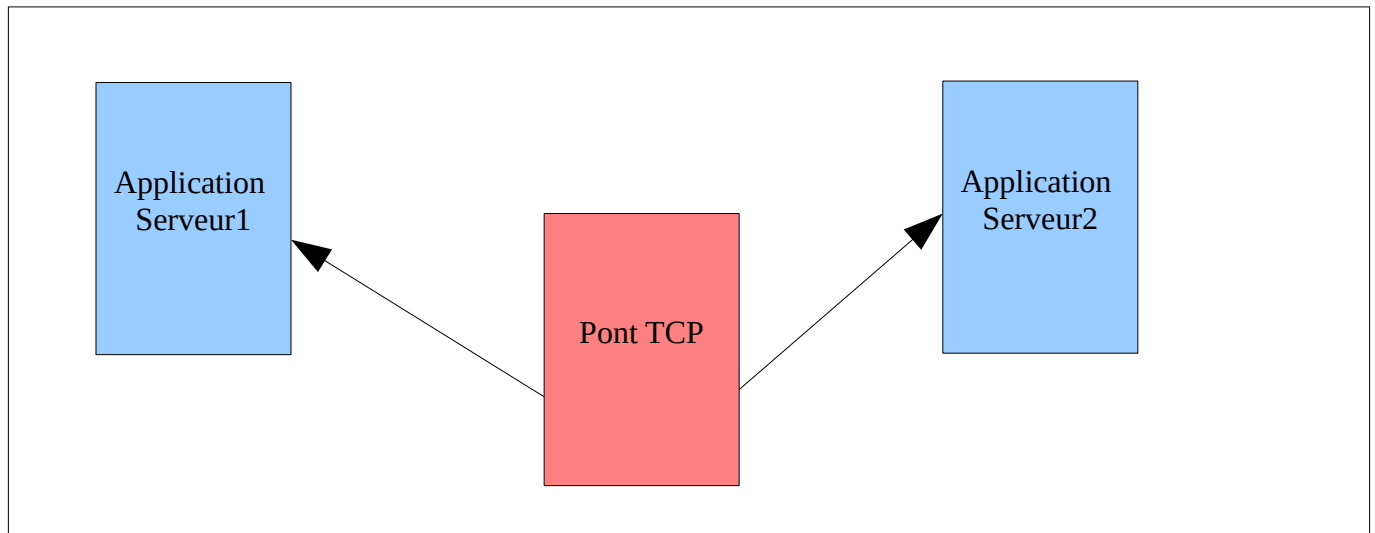
- un projet Eclipse par exercice
- le nom du projet Eclipse doit être **examen_exo1_[votrenom]** pour l'exercice 1, **examen_exo2_[votrenom]** pour l'exercice 2, ...
- vous devez rendre vos travaux en suivant les instructions données par le serveur d'examen dont l'URL est indiquée au tableau.

L'accès à Internet n'est pas autorisé, aucun document autorisé.

2 Exercice 1 : Ping Pong TCP (20 points)

2.1 Présentation générale du problème.

Vous allez devoir réaliser un pont TCP entre deux applications serveurs.



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 sous la forme d'un flux d'octets (avec uniquement le contenu du fichier , sans le nom du fichier). Le fichier a une taille d'environ 500 Mo.
- Pour cacher les données contenues dans le fichier, le programmeur a offusqué le fichier de la façon suivante :
 - le premier octet envoyé est correct
 - le deuxième octet envoyé est un octet aléatoire
 - le troisième octet envoyé est correct et correspond au deuxième octet du fichier
 - le quatrième octet envoyé est un octet aléatoire
 - le 5ème octet envoyé est correct et correspond au 3ème octet du fichier
 - et ainsi de suite
- Le flux d'octets envoyé par le serveur 1 a donc une taille d'environ 1 Go.
- Quand le serveur arrive à la fin du flux d'octet, il coupe la connexion.

Exemple : le fichier contient le mot FROID, le serveur 1 peut envoyer FXRXOXIXDX.

L'application 2 est un serveur TCP, qui écoute sur le port 8200. Ce serveur attend le contenu du fichier **en clair**. Ce serveur 2 fonctionne ainsi :

- il attend une demande de connexion
- quand la connexion est réalisée, il attend que le client lui envoie le contenu du fichier en clair (toujours sous la forme d'un flux d'octets).
- si le contenu du fichier envoyé par le client correspond bien au fichier en clair : le serveur 2 répond « Le mot de passe est 1234 » puis ferme la connexion.
- si le contenu du fichier par le client ne correspond pas au fichier en clair envoyé par le serveur 1 : le serveur 2 ferme la connexion

A noter : la vérification du fichier n'est pas faite à la fin de l'envoi complet. Le serveur 1 envoie une partie du fichier, le serveur 2 vérifie qu'il a bien reçu cette partie du fichier en clair, le serveur 1 continue d'envoyer des données, le serveur 2 fait la vérification sur la partie suivante, et ainsi de suite.

Le serveur 2 vérifie aussi le temps mis pour le décodage du fichier : le décodage du fichier doit être fait en moins de **5 secondes**.

2.2 Préparation de votre machine.

Pour pouvoir tester votre programme, vous allez installer sur votre machine ces 2 serveurs TCP.

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

Lancer ces 2 serveurs avec la commande

```
java -jar ping-pong-tcp.jar
```

Il suffit d'exécuter une seule fois cette commande, et les deux serveurs seront actifs.

Vous avez maintenant

- un serveur TCP 1 écoutant sur le port 8000 fournissant un fichier offusqué
- un serveur TCP 2 écoutant sur le port 8200 attendant un fichier en clair

2.3 Ecriture du programme PingPongTCP

Ecrivez maintenant une classe PingPongTCP.

Le code de cette classe va permettre de réaliser un premier client qui va se connecter au serveur 1 et un deuxième client qui va se connecter au serveur 2.

Le premier client va aller chercher le fichier du serveur 1, le mettre en clair et le deuxième client va prendre ce fichier en clair pour l'envoyer au serveur 2.

Etant donné que le flux d'octets est très important en taille (plus de 1Go) et que la vérification faite au fur à mesure, il ne sera pas possible d'aller chercher le flux d'octets en entier avant de l'envoyer au serveur 2. Il est nécessaire d'aller chercher le flux partie par partie et de l'envoyer partie par partie en clair au serveur 2.

Ensuite, votre programme doit lire le mot de passe qui est envoyé par le serveur 2 et l'afficher.

Si on exécute le main de cette classe, on doit obtenir l'affichage suivant :

```
Début du programme
Début du transfert des données depuis serveur 1 vers serveur 2
Fin du transfert des données.
Le mot de passe est 8205
Fin du programme.
```

Votre programme doit s'exécuter en moins de 5 secondes.

La consommation mémoire de votre programme devra rester raisonnable (c'est à dire être inférieure à 50 Mo). Vous ne devez pas stocker tout le flux inutilement.

L'utilisation de la fonction Thread.sleep() est interdite dans votre programme (ou tout autre fonction d'attente).

2.4 Test de votre programme PingPongTCP

Faites fonctionner votre programme PingPongTCP.

Dans un commentaire en haut de votre classe PingPongTCP, mettez le mot de passe que vous avez obtenu.