Objectifs : Vous allez réaliser quelques applications client-serveur utilisant le protocole TCP. L'objectif est d'apprendre à réaliser des échanges entre un programme serveur et un ou plusieurs programmes clients et de comprendre le fonctionnement de TCP.

Il va falloir :

1. effectuer une réflexion (sur papier) pour :
   - comprendre la structure de l'application (définir combien de programmes la composent et le rôle de chacun).
   - définir un protocole d'application / d'échange.
2. implémenter les programmes de l'application
3. exécuter les programmes sur différentes machines (impératif)
4. réaliser des tests mettant en évidence quelques propriétés de TCP.

Notations et rappel :
Le protocole de transport **TCP** permet de réaliser des communications en mode connecté. Un message envoyé en **TCP** est transféré/acheminé sous forme de flux d'octets. Enfin, **TCP** gère la duplication et la remise dans l'ordre des paquets à leur réception.

1 Dialogue avec un serveur

Dans une application client-serveur, un serveur attend un message, sous forme de chaine de caractères, envoyé par un client, affiche ce message, renvoie au client la taille du message reçu (en nombre d'octets) et termine le traitement de ce client.

L'objectif de l'exercice est d'écrire uniquement un programme client et de tester sont fonctionnement. Ce client communiquera avec un serveur s'exécutant sur une machine et dont la sortie standard sera projetée pendant le TP. L'adresse de la socket d'écoute vous sera fournie.

Ecrire et tester le programme client sachant que l'objectif à l'exécution est d'avoir un affichage correct des messages envoyés au serveur et un nombre minimum d'octets échangés (réduits à des données utiles).

2 Et le serveur ?

Ecrire le programme serveur de l'exercice précédent. Nous supposons pour l'instant qu'il ne gère qu'un seul client et termine.
Ensuite, exécuter votre programme client et programme serveur sur des machines différentes et assurez vous du bon fonctionnement de votre application.
Que se passe-t-il si le client se termine avant l'envoi d'un message ? Si ce cas n'a pas été pris en compte, corriger vos programmes.

3 Les échanges en TCP

L'objectif de cet exercice est de mettre en évidence des propriétés du protocole TCP vues en cours et en TD.
Ecrire deux programmes :
- un programme client qui envoie successivement deux chaines de caractères saisies au clavier (il fait donc 2 envois) et affiche le nombre total d'octets envoyés. La taille d'une chaine de caractères saisie au clavier ne dépassera pas 32 caractères.
- un programme serveur qui reçoit une suite d'octets de taille maximum 124 et qui affiche le nombre d'octets reçus. Le serveur ne fait donc qu'une seule réception.

Exécuter les deux programmes (sur deux machines différentes) en s'assurant que les deux chaines de caractères on été toutes les deux envoyées avant que le serveur ne soit en réception. Qu'observez vous ? Corriger votre programme pour gérer ce problème ?

Ensuite :

1. modifier le programme client pour qu'il envoie en boucle une même chaine de caractères saisie au clavier (taille maximum 2000 caractères). Le nombre de chaines à envoyer (nombre d'itérations) sera un paramètre de votre programme. Enfin, le programme affichera le nombre total d'octets envoyés depuis la première émission (peut importe le type des messages).

2. modifier le programme serveur pour qu'il puisse recevoir les chaines de caractères envoyées par le client. Le serveur affichera à chaque réception le nombre total d'octets reçus par le client depuis la première réception (peu importe le type des messages).
3. Exécuter les deux programmes (sur deux machines différentes !) en faisant varier la taille du message et le nombre de messages à envoyer de quelques uns à plusieurs milliers. Qu'observez vous ? Le nombre total d'octets envoyés est-il toujours égal au nombre d'octets reçus ? Si ce n'est pas le cas, expliquer le problème et corriger votre programme. Il est rappelé qu'il n'y a pas de perte de paquets en TCP ni de duplication à la réception par la couche application.
4. Relancer votre application de manière à remplir le buffer de réception du serveur et le buffer d'envoi du client. Que se passe-t-il ?

4 Bilan TCP vs UDP

Lors du TP2, vous avez pu observer ce qui se produit lors des échanges de messages en utilisant le protocole UDP pour mettre en évidence quelques propriétés de ce protocole. Quelles sont les différences observées avec TCP ? Pour réponde à cette question, appuyez vous sur des situations concrètes étdudiées en TP, par exemple : quel est le retour des fonctions d'envoi et de réception ? Peut ont détecter dans tout les cas l'arrêt d'un des programmes communiquant ? Que se passe-t-il dans le cas où le buffer de réception est plein ? etc.