

# Introduction Internet et Réseaux TD

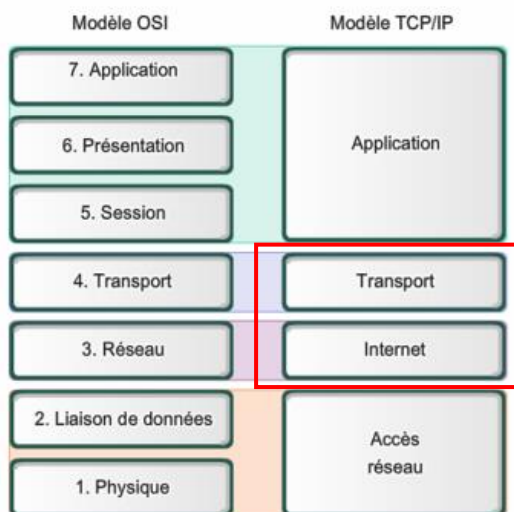
## Programmation Socket TCP/IP en C#

Le but de ce TD est de se familiariser avec l'utilisation d'internet et des réseaux au travers des programmes en C#. Nous utiliserons pour cela, le namespace System.Net.Socket du .Net Framework. Ce TD se déroulera sur vos PCs portables, sous Windows avec Visual Studio.

**ATTENTION : Vous n'êtes bien sûr pas familier avec toutes les notions ci-dessous, en particulier celles qui gère le mail. Les TDs encadré par l'enseignant sont prévu pour cela...**

## 1 Rappel sur l'API des sockets sous le .Net Framework en C#

### 1.1 Rappel sur les principes d'Internet et en particulier TCP/IP :



La figure ci-joint rappelle la projection de la pile TCP/IP dans la pile OSI/ISO.

- *Internet* appartient donc à la couche réseau. C'est un standard qui permet d'acheminer des paquets d'une machine d'adresse IP<sub>1</sub> à une machine d'adresse IP<sub>2</sub> en passant par des machines et des équipements intermédiaires. Tout ceci est orchestré grâce à des tables de routage qui en chaque point du réseau (IP<sub>x</sub>), définisse le point suivant à atteindre (IP<sub>y</sub>) pour in fine atteindre IP<sub>2</sub>.
- TCP (Transmission control protocol) introduit quant à lui les notions de couche de transport et de port pour permettre à une machine IP<sub>1</sub> de dialoguer simultanément avec plusieurs autres machines du réseau internet. Les ports peuvent alors être vus comme des boîtes aux lettres. Ils se caractérisent par un identifiant sur 2 octets. Dès lors un programme de la machine A qui veut

dialoguer avec un programme de la machine B, établira une communication entre deux « EndPoints » : IP<sub>a</sub>,Port<sub>a</sub> et IP<sub>b</sub>,Port<sub>b</sub>.

- TCP présente aussi l'avantage de gérer la longueur des messages et l'ordre dans lequel ils sont envoyés. Ainsi vous pouvez envoyer un flux de caractères depuis le programme de la machine A vers le programme de la machine B sans vous soucier de sa fragmentation sur le réseau.
- Pour cela il vous faut gérer une connexion TCP, c'est-à-dire convenir entre les deux programmes concernés d'un début et d'une fin d'échange de messages.

Nous allons étudier les codes C# pour la mise en œuvre d'un client et d'un serveur TCP/IP que vous trouverez sur le site du Cours (Cf. [www.tigli.fr](http://www.tigli.fr)).

## 2 Une première mise en œuvre

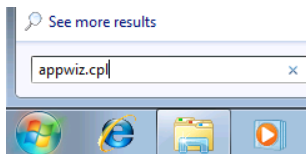
## Introduction Internet et Réseaux TD

### Programmation Socket TCP/IP en C#

Nous allons avoir dans la suite, besoin du programme « Telnet ». Le programme "telnet" se lance avec la commande « telnet <nom de la machine ou son adresse IP > <numéro du port sur lequel on veut se connecter> » (ex. telnet localhost 2245).

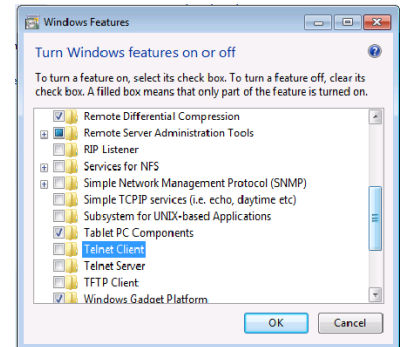
Après connexion, le programme «telnet» envoie tous les caractères que vous tapez sur la console vers le serveur et affiche tous les caractères envoyés par le serveur sur la console.

Pour installer «telnet» sous windows 7, procédez de la manière suivante :



Pour activer Telnet sur Windows 7, ouvrez la « console programs » et entrez la commande **appwiz.cpl**

Une fois la fenêtre « Windows Features » affichée, cliquez sur «Turn windows features on or off », dans la liste qui apparaît cocher la case : **Telnet Client**. Puis cliquer sur OK



**Exercice 1 :** A partir du code source serveur ci-dessus, développer un « ServeurTCPIP » pour qu'il devienne un serveur d'écho, c'est-à-dire qu'il renvoie tous les caractères qu'il reçoit. Testez-le avec « telnet ».

**Exercice 2 :** \*Nous allons maintenant écrire un client Mail. Il faut savoir que quand vous envoyez un mail un certain nombre de commandes sont envoyées et reconnues par le serveur de mail distant. Il s'agit du « Simple Mail Transfer Protocol », SMTP.

Voici ci-dessous un exemple avec « telnet ». Après avoir tapé « telnet <serveur SMTP ouvert> 25 », nous obtenons (les caractères bleus sont envoyés par le serveur et les caractères noirs sont ceux que j'envoie avec telnet).

```

220 smtp1-g21.free.fr ESMTP Postfix
HELO 123
250 smtp1-g21.free.fr
MAIL FROM:<jytigli@free.fr>
250 2.1.0 Ok
RCPT TO:<tigli@unice.fr>
250 2.1.5 Ok
DATA
354 End data with <CR><LF>.<CR><LF>
Subject: Ici j'écris mon sujet
Ici j'écris mon corps de message
.
250 2.0.0 Ok: queued as 716BB940165
QUIT

221 0.0 Bye
  
```

\*Cette exercice ne peut se faire si vous n'avez pas trouvé de serveur smtp ouvert sur le port 25

**Exercice 3 :** Reprenez le code précédent pour envoyer une requête GET / à un serveur Web. Que récupérez-vous ?

Comment est-ce traité d'habitude?