



2005-2006

# **MASTER IGLII**

Ingénierie de la Langue  
pour la gestion intelligente de l'information

**INTRANET EXTRANET**  
**TD 5 : Programmation réseau 1**  
**Les sockets**

---

## 1. QUELQUES ELEMENTS DE COURS

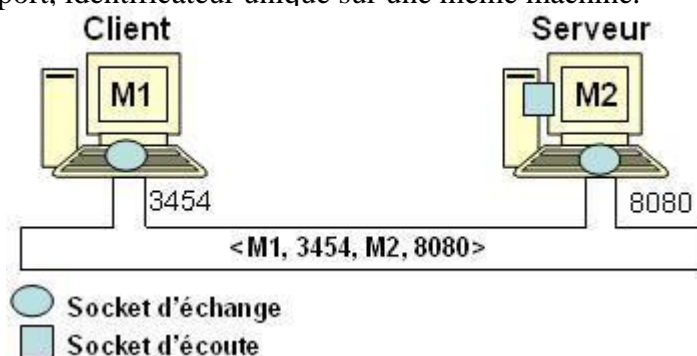
### 1.1. Rappels

#### Les sockets

Les données transitent sur l'Internet dans des paquets d'octets de taille limitée appelés datagrammes, formés d'un entête (adresse, numéro de port de l'expéditeur et du destinataire,...) et d'une charge utile composée des données. Un transfert unique peut nécessiter plusieurs paquets ce qui implique des difficultés de réorganisation de l'information à l'arrivée. Les sockets permettent la gestion d'un transfert de données comme un flux ordinaire (les paquets sont triés à l'arrivée).

Une socket est décrite par :

- une adresse de machine (adresse IP);
- un numéro de port, identificateur unique sur une même machine.



#### Modes de communication

La communication par socket est souvent comparée aux communications humaines. On distingue ainsi deux modes de communication :

- **Le mode connecté** (comparable à une communication téléphonique), utilisant le protocole TCP. Dans ce mode de communication, une connexion durable est établie entre les deux processus, de telle façon que l'adresse de destination n'est pas nécessaire à chaque envoi de données.
- **Le mode non connecté** (analogue à une communication par courrier), utilisant le protocole UDP. Ce mode nécessite l'adresse de destination à chaque envoi.

#### Les classes

Plusieurs classes interviennent lors de la réalisation d'une communication par sockets. La classe `java.net.InetAddress` permet de manipuler des adresses IP. La classe `java.net.SocketServer` permet de programmer l'interface côté serveur en mode connecté. La classe `java.net.Socket` permet de programmer l'interface côté client et la communication effective par flot via les sockets. Les classes `java.net.DatagramSocket` et `java.net.DatagramPacket` permettent de programmer la communication en mode datagramme.

#### 7 opérations élémentaires sont assurées par une Socket :

- Connexion à une machine distante (Socket),
- Envoi de données (Socket),
- Réception de données (Socket),
- Fermeture d'une connexion (Socket),
- Attachement à un port (ServerSocket),
- Attente de demande de connexion provenant de machines distantes (ServerSocket),
- acceptation de demandes de connexion sur le port local (ServerSocket).

---

**Mise en œuvre :**

- La Socket est créée sur le client.
- Le ServerSocket fonctionne sur le serveur en attente de connexion (méthode accept()) ;
- La nouvelle Socket tente de se connecter sur le serveur distant.
- Après détection d'ouverture de connexion, le ServerSocket génère une Socket pour communiquer avec le client.
- Une fois la connexion établie, les systèmes locaux et distant établissent un canal de communication par flux à partir de leurs Sockets de manière à échanger des données.
- Lorsque les échanges sont terminés, l'un des interlocuteurs clôt la communication.
- Socket permet l'instanciation de sockets (sur les clients et serveurs) pour initier et gérer les transferts de données au cours d'une communication.

|                     |
|---------------------|
| <b>2. QUESTIONS</b> |
|---------------------|

1. Ecrire un programme Java qui permet de retourner l'adresse IP d'un serveur de messagerie, exemple « smtp.laposte.net »?
2. Créer une communication client/serveur entre deux machines ?
3. Créer une communication client/serveur entre trois machines ?