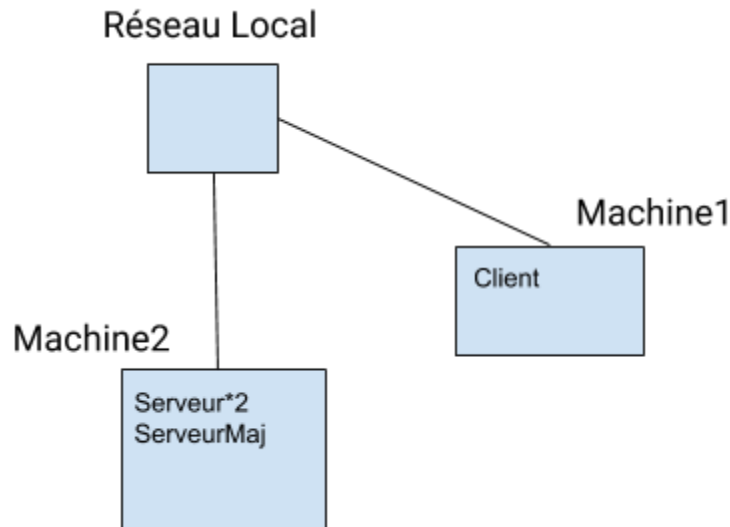


# BPO

## TP Application Client-Serveur - Partie 3



Ce schéma (déjà vu à la fin du TP précédent) fait la synthèse du travail réalisé jusqu'à maintenant.

L'utilisation des adresses IP est peu pratique. Pour y remédier, on utilise une table **DNS** qui associe un nom à une adresse IP. Ainsi, on pourra utiliser un nom de machine, plutôt qu'une adresse IP.

1) Chaque machine est enregistrée dans une table **DNS**, qui permet d'associer le nom et l'adresse IP. Écrire et tester la classe **reseau.tables.DNS**, avec les fonctions suivantes :

```
DNS()
void ajouter (String nomMachine, Adresse adrIP)
Adresse getAdresse (String nomMachine)
```

Le constructeur permet de construire une table vide. La fonction **ajouter** permet d'associer un nom de machine avec une adresse IP, par exemple :

```
DNS dns = new DNS() ;
nom = "zebulon" ;
adr = new Adresse ("192.23.23.23") ;
dns.ajouter(nom, adr);
```

2) Compléter la classe **TestReseau** avec la création de la table **DNS**. Chaque machine du réseau local doit être enregistrée dans cette table.

3) Ajouter une nouvelle machine dans le réseau local avec une application serveur **ServeurDNS** capable de fournir l'adresse IP correspondant à un nom, en consultant la table **DNS**. Choisir un numéro de port pour cette application serveur. Regardez le diagramme de classes partiel en fin de sujet.

Cette classe a pour constructeur :

```
public ServeurDNS (int port, Machine m, Application clientDNS, DNS dns)
```

Ajouter dans la classe **TestReseau** l'envoi d'un message vers ce nouveau serveur, pour récupérer l'adresse IP d'une machine de nom donné.

4) Écrire la classe **ClientDNS** pour décrire l'application cliente qui sait interroger le serveur DNS pour retrouver l'adresse IP correspondant à un nom. Cette classe a pour constructeur :

```
public ClientDNS (int port, Machine m, Adresse adr)
```

où adr est l'adresse IP du serveurDNS.

Cette classe doit redéfinir la fonction **sendMessage** pour envoyer le message au serveurDNS dont on connaît l'adresse IP, le message étant le nom de la machine dont on demande à connaître l'adresse IP.

```
void sendMessage(String nomDest, int portDest, Message message) {
```

Créer une application de ce type sur chaque machine.

5) Chaque application de chaque machine du réseau connaît cette application **ClientDNS**, passée en paramètre lors de la création.

```
Application appliClientNum = new ClientNumerique (portClient, machine, clientDNS)
```

6) Un client peut maintenant envoyer un message à un serveur, sans connaître son adresse IP, mais juste le nom de la machine sur laquelle se trouve le serveur. Il faut pour cela ajouter une fonction **sendMessage** dans la classe **Application**, avec un paramétrage différent (cf appel ci-dessous). Cette fonction commence par récupérer l'adresse IP du destinataire en s'adressant au serveur **DNS**, avant de faire l'envoi avec l'adresse IP.

```
appliClientNum.sendMessage("zebulon", portFoisDeux, message)
```

