

Journées d'Immersion

MakerSpace - Introduction aux réseaux Pouloulou

24 et 25 Février 2022



Table des matières

| | | |
|----------|---|----------|
| 1 | Introduction | 3 |
| 1.1 | Qu'est-ce qu'un réseau ? | 3 |
| 1.2 | Bref historique | 3 |
| 2 | Relier les Equipements | 4 |
| 2.1 | Prenez des équipements | 4 |
| 2.1.1 | Switchs | 4 |
| 2.1.2 | Routeurs | 4 |
| 2.1.3 | Serveurs | 4 |
| 2.1.4 | Les équipements clients | 4 |
| 2.2 | ... Un peu de topologie | 5 |
| 2.3 | ... et ajoutez de l'adressage | 5 |
| 2.3.1 | Adresses IPv4 | 5 |
| 2.3.2 | Gestion des ports | 5 |
| 3 | Créer une connexion | 6 |
| 3.1 | Step 1 | 6 |
| 3.2 | Step 2 | 6 |
| 3.3 | Pour aller plus loin... | 6 |
| 4 | Communiquer à travers la connexion | 7 |
| 5 | Avant de partir | 7 |

1 Introduction

1.1 Qu'est-ce qu'un réseau ?

Un réseau informatique (network) est un ensemble d'équipements reliés entre eux pour échanger des informations. On appelle nœud l'extrémité d'une connexion, qui peut être une intersection de plusieurs connexions ou équipements (un ordinateur, un routeur, un switch, un serveur).

De façon verticale, un réseau est une strate de trois couches : les infrastructures physiques, les protocoles de communications (TCP/IP, UDP, HTTP) et les données applicatives.¹

On peut également classer les réseaux en fonction de leur portée : réseau local, réseau d'accès et réseau d'interconnexion mondial (The Internet).²

1.2 Bref historique

Dans les années 1960, les premiers réseaux informatiques étaient de portée limitée (quelques dizaines de mètres) et servaient à la communication entre micro-ordinateurs et des instruments de mesures ou des périphériques (imprimantes, table traçante, etc.).

Les réseaux informatiques filaires entre sites distants apparaissent dans les années 1970 avec la numérisation du réseau de téléphone d'ATT et ses connexions dédiées à moyen débit.

Aujourd'hui, bien que d'importantes avancées technologiques aient vu le jour (fibre, 5G, nouveaux protocoles IPv6...), les principes de bases des réseaux restent inchangés.

1. Plus d'infos sur le modèle de couches OSI : https://fr.wikipedia.org/wiki/Mod%C3%A8le_OSI

2. Plus d'infos sur les classes IP : https://fr.wikipedia.org/wiki/Classe_d%27adresse_IP

2 Relier les Equipements

2.1 Prenez des équipements ...

2.1.1 Switchs

Un commutateur réseau (Switch), est un équipement qui relie plusieurs câbles dans un réseau informatique. La commutation est un des deux modes de transport de trames au sein des réseaux informatiques et de communication, l'autre étant le routage. Dans les réseaux locaux (LAN), il s'agit le plus souvent d'un boîtier disposant de plusieurs ports RJ45 (entre 4 et plusieurs centaines), il a donc la même apparence qu'un concentrateur (hub). Il existe aussi des commutateurs pour tous les types de réseau en mode point à point comme pour les réseaux ATM, relais de trames, etc.³

2.1.2 Routeurs

Un routeur est un équipement réseau informatique assurant le routage des paquets. Son rôle est de faire transiter des paquets d'une interface réseau vers une autre, au mieux, selon un ensemble de règles. Il y a habituellement confusion entre routeur et relais, le routeur s'occupe d'assurer le cheminement tandis que le relai ne fait pas plus que ce que son nom indique.⁴

2.1.3 Serveurs

Un serveur informatique est un dispositif informatique (matériel et logiciel) qui offre des services à un ou plusieurs clients tels que l'accès aux informations du World Wide Web, le partage de périphériques, le stockage en base de données... En fonctionnement, un serveur répond automatiquement à des requêtes provenant d'autres dispositifs informatiques (les clients), selon le principe dit client-serveur. Le format des requêtes et des résultats est normalisé et se conforme à des protocoles réseaux.⁵

2.1.4 Les équipements clients

Dans un réseau informatique, un client est le logiciel qui envoie des demandes à un serveur. Il peut s'agir d'un logiciel manipulé par une personne, ou d'un bot. Est appelé client aussi bien l'ordinateur ou smartphone depuis lequel les demandes sont envoyées que le logiciel qui contient les instructions relatives à la formulation des demandes et la personne qui opère les demandes.⁶

3. Plus d'infos sur commutateurs réseaux : https://fr.wikipedia.org/wiki/Commutateur_r%C3%A9seau

4. Plus d'infos sur les routeurs : <https://fr.wikipedia.org/wiki/Routeur>

5. Plus d'infos sur les serveurs : https://fr.wikipedia.org/wiki/Serveur_informatique

6. Plus d'infos sur les clients réseaux : [https://fr.wikipedia.org/wiki/Client_\(informatique\)](https://fr.wikipedia.org/wiki/Client_(informatique))

2.2 ... Un peu de topologie ...

Une topologie de réseau informatique correspond à l'architecture (physique, logicielle ou logique) de celui-ci, définissant les liaisons entre les équipements du réseau et une hiérarchie éventuelle entre eux.

Elle peut définir la façon dont les équipements sont interconnectés et la représentation spatiale du réseau (topologie physique). Elle peut aussi définir la façon dont les données transitent dans les lignes de communication (topologies logiques).

2.3 ... et ajoutez de l'adressage

2.3.1 Adresses IPv4

Une adresse IP (Internet Protocol) est un numéro d'identification qui est attribué de façon permanente ou provisoire à chaque périphérique relié à un réseau informatique qui utilise l'Internet Protocol. L'adresse IP est à la base du système d'acheminement (le routage) des paquets de données sur Internet.

Il existe des adresses IP de version 4 sur 32 bits, et de version 6 sur 128 bits. La version 4 est actuellement la plus utilisée : elle est généralement représentée en notation décimale avec quatre nombres compris entre 0 et 255, séparés par des points, ce qui donne par exemple « 192.168.0.1 »⁷

2.3.2 Gestion des ports

La notion de port logiciel permet, sur un ordinateur donné, de distinguer différents interlocuteurs. Ces interlocuteurs sont des programmes informatiques qui, selon les cas, écoutent ou émettent des informations sur ces ports. Un port est distingué par son numéro. Le terme port est aussi parfois utilisé pour désigner les sockets (interfaces de connexion), un concept sensiblement différent. Là où un socket sert d'identifiant unique dans un réseau donné résultant de la concaténation de l'adresse internet et du numéro de port, un port sert localement à identifier un processus.⁸

7. Plus d'infos sur les adresses IP : https://fr.wikipedia.org/wiki/Adresse_IP

8. Plus d'infos sur les ports réseaux : [https://fr.wikipedia.org/wiki/Port_\(logiciel\)](https://fr.wikipedia.org/wiki/Port_(logiciel))

3 Créer une connexion

On passe au concret dès maintenant ! Nous vous proposons d'écrire un client réseau pour communiquer avec un serveur, le tout en Python.

3.1 Step 1

Tout d'abord commençons par récupérer les informations nécessaires à la connexion au serveur.

Une fois cela fait, ouvrez le fichier `my_client.py` dans l'éditeur Visual Studio Code et prenez connaissance du code déjà rédigé. Une fois cela fait, demandez à l'utilisateur de saisir son nom, l'adresse IP et le port à utiliser à l'aide de la fonction suivante :

```
variable = input(prompt)
# brief : Lis une entrée utilisateur dans une variable
# param prompt : Le message à afficher avant la lecture
# return : la valeur lu
```

3.2 Step 2

Vous pouvez créer une nouvelle socket de connexion à l'aide de la fonction fournie suivante :

```
connection = Connection(name, address, port)
# brief : Crée une nouvelle connexion à l'aide des informations renseignées
# param name : Le nom du client
# param address : L'adresse IP du serveur
# param port : Le port à utiliser pour la communication
# return : la connexion fraîchement créée
```

3.3 Pour aller plus loin...

Ce paragraphe fait référence à des concepts de programmation avancés et sa compréhension n'est pas nécessaire pour la suite :

`Connection` n'est pas une fonction classique mais le constructeur de la classe du même nom. Cette classe porte la socket réseau qui sera utilisée et d'autres informations nécessaires au fonctionnement du client et devant transiter entre chaque fonction mais qui nous semblent complexes à appréhender pour une introduction au réseau. C'est également pour cela que les fonctions que vous aurez à utiliser plus loin sont appelées avec la syntaxe `connection.fonction()` ; il s'agit en fait des méthodes de la classe appelées directement sur l'objet `connection`. Si vous souhaitez en savoir plus, vous pouvez jeter un oeil au fichier `my_little_chat.py` à la fin du TP.

4 Communiquer à travers la connexion

Le client doit en permanence être prêt à échanger des messages avec le serveur. Pour cela il est nécessaire de créer une boucle infinie attendant un événement. Dès qu'un événement est déclenché, des actions doivent être exécutées en conséquence. Voici quelques fonctions que nous vous fournissons pour vous aider dans cette mission :

```
event = connection.waitForEvent()  
# brief : En attente d'un événement  
# return : les valeurs d'événements pouvant être retournées sont:  
# EventType.READ : des informations ont été saisies et doivent être lues  
# EventType.RECV : un message est prêt à être reçu par le client  
# EventType.SEND : un message est prêt à être envoyé par le client  
  
connection.readInput()  
# brief : Lis l'entrée utilisateur(le message est stocké en interne)  
  
connection.receive()  
# brief : Récupère le message en attente de réception  
# return : le message reçu  
  
connection.send()  
# brief : envoie le message stocké par ReadInput()
```

5 Avant de partir

Toute connexion qui a été ouverte doit être fermée par vos soins, encore une fois, on vous aide avec la fonction suivante :

```
connection.close()  
# brief : ferme la socket réseau et les éléments utilisés par la connexion
```

Merci à tous !