

Bases des Réseaux

Cédric Vanconingsloo

Table des matières

| | |
|---|----|
| 1. Introduction | 1 |
| 2. Les réseaux de communication | 3 |
| 2.1. Définition et types de réseaux | 3 |
| 2.2. Modes de circulation et de connexion | 3 |
| 2.2.1. Le mode connecté | 4 |
| 2.2.2. Le mode non connecté | 4 |
| 3. Le modèle OSI | 5 |
| 3.1. Critique | 6 |
| 3.1.1. Ce n'était pas le bon moment | 6 |
| 3.1.2. Ce n'était pas la bonne technologie | 6 |
| 3.1.3. Ce n'était pas la bonne implémentation | 6 |
| 3.1.4. Ce n'était pas la bonne politique | 6 |
| 4. Le modèle TCP/IP | 7 |
| 5. Le modèle Hybride | 8 |
| 5.1. La couche Physique | 8 |
| 5.1.1. Le codage PAM-5 | 8 |
| 5.1.1.1. Le câble torsadé | 9 |
| 5.1.1.2. La fibre optique | 10 |
| 5.1.1.3. Les réseaux sans fil | 10 |
| 5.1.2. Les topologies | 10 |
| 5.1.2.1. Le bus | 11 |
| 5.1.2.2. L'étoile | 12 |
| 5.1.2.3. L'anneau | 12 |
| 5.1.2.4. Le maillage | 13 |
| 5.1.2.5. L'arbre | 13 |
| 5.1.3. Le Hub | 14 |
| 5.1.4. Les protocoles CSMA/CD et CSMA/CA | 14 |
| 5.1.4.1. Le CSMA/CD | 14 |
| 5.1.4.2. Le CSMA/CA | 15 |
| 5.2. La couche Liaison | 15 |
| 5.2.1. Le réseau ethernet et l'adresse mac | 15 |
| 5.2.2. La trame ethernet | 16 |
| 5.2.3. Le switch | 16 |
| 5.3. La couche réseau | 17 |
| 5.3.1. Les protocoles | 17 |
| 5.3.1.1. Le protocole IP | 17 |
| 5.3.1.1.1. Classes d'adresses | 17 |
| 5.3.1.1.2. Adressage IPv4 | 17 |
| 5.3.1.1.3. Adressage IPv6 | 18 |
| 5.3.1.1.4. Adresses réservées | 18 |
| 5.3.1.1.5. Masque de sous-réseaux | 19 |
| 5.3.1.2. Le protocole ARP | 19 |
| 5.3.1.2.1. La table ARP | 19 |
| 5.3.1.3. Le protocole ICMP | 20 |
| 5.3.2. Le routeur | 20 |
| 5.3.2.1. Notion de route | 20 |

| | |
|--|----|
| 5.3.2.1.1. Exemple | 20 |
| 5.4. La couche Transport | 21 |
| 5.4.1. Fonctionnement | 21 |
| 5.4.1.1. Exemple | 22 |
| 5.4.2. Protocoles | 22 |
| 5.4.2.1. TCP | 22 |
| 5.4.2.2. UDP | 23 |
| 5.4.2.3. QUIC | 23 |
| 5.4.3. Le Firewall | 23 |
| 5.4.3.1. Principe de fonctionnement | 23 |
| 5.5. La couche Application | 24 |
| 5.5.1. Protocoles | 24 |
| 5.5.1.1. DNS | 24 |
| 5.5.1.2. DHCP | 24 |
| 5.5.1.3. HTTP | 25 |
| 5.5.1.4. SSH | 25 |
| 5.5.1.5. Les sockets | 26 |
| 6. Notion d'adressage réseau en IPv4 | 27 |
| 6.1. Adressage « statefull » | 27 |
| 6.1.1. Autre exemple | 28 |
| 6.2. Adressage VLSM | 29 |

1. Introduction

Internet a fêté ses 50 d'existence en 2019. À l'époque, c'était la guerre froide entre deux grandes puissances, les **USA** et **l'URSS**.

Internet doit sa naissance à deux événements majeurs : la bombe atomique et **Spoutnik**.

Spoutnik est le premier satellite mis en orbite autour de la Terre. Cette sphère de 58 cm de diamètre et de 83 kg, faisant le tour de la planète en 96 min, avait pour seul but l'émission de son fameux bip bip biiiiiiip bien connu.

Ce lancement est un vrai scandale pour les USA, qui craignaient que les Russes, grâce à Spoutnik et à leur avancée technologique, ne soient capables de lancer une bombe atomique sur leur sol.

En quoi la bombe H eût-elle joué un rôle ? La principale force des deux entités à l'époque, c'était la force de dissuasion, et dans une moindre mesure, les télécommunications. Les deux côtés du Rideau de Fer disposent de la bombe H et des réseaux de télécommunications pour la lancer. Ce qui les retenait, c'est que l'autre camp puisse lui aussi appuyer sur le bouton en représailles.

Et c'est précisément là que le bât blesse les États-Unis. Les télécommunications militaires des USA étaient centralisées. Toutes les données de leurs calculateurs transitaient par un point névralgique. Les Russes disposaient de Spoutnik, donc de la possibilité (fausse, bien sûr) de commander une attaque atomique de n'importe où. Et si une bombe venait à détruire de centre des commandes des USA ? Ils seraient dans l'impossibilité de répondre par une frappe nucléaire.

Pour répondre à cette question, l'État-Major américain demande au **DoD** (*Department of Defense*) de former un groupe de travail visant à créer un réseau de télécommunications décentralisé, de telle sorte que si un nœud stratégique fût détruit, les ordres pouvaient continuer de transiter par d'autres nœuds. **L'ARPA** (*Advanced Research Project Agency*) est fondé.

Un modèle de réseau avait déjà été théorisé par le scientifique **Paul Baran** en 1964, soit cinq ans avant l'ARPA. À l'époque, les militaires ont rejeté son idée, jugée trop coûteuse. Baran mit au point un réseau sous forme de grande toile hybride d'architectures maillées et étoilées, dans lequel les données se déplaceraient de nœud en nœud, en passant par le chemin le moins encombré, sans réserver le trajet des données au départ. C'est la théorie de la **commutation de paquets** (*packet switching*).

En août 1969, l'ARPA dévoile ARPANET, le premier réseau décentralisé connectant quatre instituts universitaires avec des câbles allant à 50 kbps et utilisant le protocole **NCP** (*Network Control Protocol*), l'ancêtre de TCP.

Le réseau Arpanet comportait déjà à l'époque certaines caractéristiques fondamentales du réseau actuel :

- Un ou plusieurs nœuds du réseau pouvaient être détruits sans perturber son fonctionnement ;
- La communication entre machines se faisait sans machine centralisée intermédiaire ;
- Les protocoles utilisés étaient basiques.

En octobre **1969** eut lieu la première tentative de connexion à distance. **Leonard Klein-rock**, professeur à l'UCLA (Californie) tente d'envoyer LOG IN d'un ordinateur à un autre. Tentative qui échoua à l'envoi du « G »....

En **1971**, **Ray Tomlinson** crée un petit truc inutile: l'*email*. Il s'envoie à lui-même un petit message: QWERTYUIOP. C'est aussi lui qui détermine le @ comme séparateur entre le nom d'utilisateur et le gestionnaire de réseau.

Plus tard, en **1973**, le protocole *NCP* est remplacé par une version plus performante, **TCP**.

Le **28 février 1990**, les serveurs d'ARPANET s'éteignent. Heureusement, tous les ordinateurs connectés continuent de fonctionner normalement: **Internet** est né.

Un résumé en vidéo ? <http://youtube.com/watch?v=9hIQjrMHTv4> .

2. Les réseaux de communication

Un *réseau* est un moyen de communication qui permet à des individus ou à des groupes de partager des informations et des services. Un réseau est constitué d'équipements appelés *nœuds*. Ces réseaux sont catégorisés en fonction de leur étendue et de leur domaine d'application.

Pour communiquer entre eux, les nœuds utilisent des *protocoles* compréhensibles par tous.

Le terme **réseau** se définit en fonction du contexte :

- **Physiquement**, c'est un ensemble de machines ;
- **Organisationnellement**, c'est l'infrastructure d'un parc informatique avec ses protocoles ;
- **Structurellement**, c'est le mode de connexion des machines (réseau en étoile, wifi...).

2.1. Définition et types de réseaux

Un réseau peut se définir par sa taille :

Le réseau PAN (0-10m) : La plus petite étendue de réseau est le *réseau personnel*, nommé réseau **PAN** (*Personal Area Network*). Centrée sur l'utilisateur, elle désigne une interconnexion d'équipement informatique dans un espace d'une dizaine de mètres autour de celui-ci. Les connexions *Bluetooth* sont un exemple de **PAN**.

Le réseau LAN (10-1000m) : De taille supérieure, s'étendant sur quelques dizaines de mètres à quelques centaines de mètres, le **LAN** (*Local Area Network*) relie entre eux des ordinateurs, des serveurs... couramment utilisés pour le partage des ressources communes au sein d'une entreprise, d'une école ou de la maison.

Le réseau sans fil WLAN (0-5km) : Ces réseaux utilisent des ondes radio pour relier les ordinateurs sans l'aide de câbles. À ne pas confondre avec les réseaux **cellulaires**, qui permettent de se connecter à Internet via la téléphonie mobile.

Le réseau MAN (1-10 km) : Le réseau métropolitain ou **MAN** (*Metropolitan Area Network*) est aussi nommé *réseau fédérateur*. Il assure les communications sur de plus longues distances, interconnectant souvent plusieurs réseaux LAN. Il peut servir à interconnecter, par une liaison privée ou non, différents bâtiments, distants de quelques dizaines de kilomètres.

Le réseau WAN (10km et +) : Les étendues de réseaux les plus conséquentes sont classées en **WAN** (*Wide Area Network*). Constitués de réseaux LAN et MAN, les réseaux étendus sont capables de transmettre les informations dans des villes différentes ou des pays différents.

Le réseau GAN : Les réseaux mondiaux relient des ordinateurs entre eux à travers le monde, comme **Internet**.

2.2. Modes de circulation et de connexion

L'information peut circuler de deux manières différentes :

- Soit elle est envoyée de façon **complète** sur un circuit **établi de bout en bout**. C'est le principe de *communication par circuits*. Ce type de communication est obsolète.

- Soit elle est fragmentée en paquets plus petits. Chaque paquet transite sur le réseau selon son propre circuit, indépendamment des autres. Les paquets sont réassemblés à la destination. C'est le principe de *commutation de paquets*.

Pour la *commutation de paquets*, il existe deux modes de connexion : le mode **connecté** et le mode **non connecté**.

2.2.1. Le mode connecté

Ce mode fonctionne un peu comme le *téléphone*. Une connexion logique est établie entre les deux points avant que les données ne soient transmises. Cela garantit la livraison des paquets et leur ordre d'arrivée.

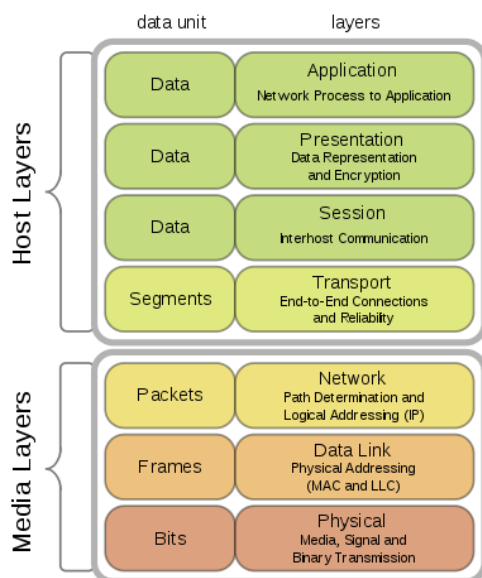
2.2.2. Le mode non connecté

Ici, les paquets sont envoyés sans établir de lien préalable. Chaque paquet est indépendant. La *poste* est l'exemple typique d'un mode non connecté. Ce mode est plus rapide que le mode connecté.

3. Le modèle OSI

Le modèle **OSI** est un modèle théorique de communication à travers un réseau. Ce modèle est une norme qui explique comment les ordinateurs devraient communiquer entre eux.

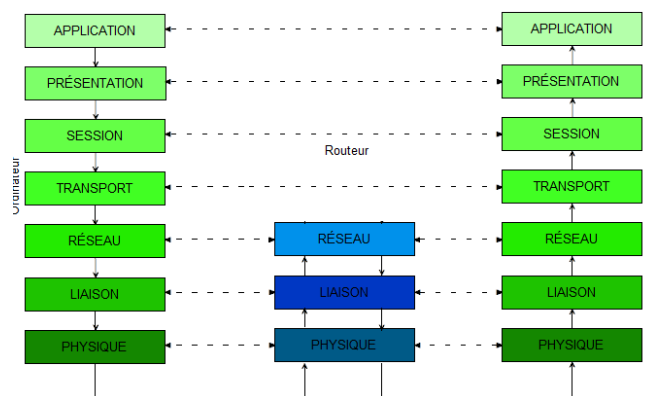
Ce modèle se décompose en sept couches distinctes. Chaque couche dispose de son propre matériel. Une couche de niveau N propose des services à la couche $N+1$ en se servant des services fournis par la couche $N-1$.



Certains types de matériels vont implémenter une à plusieurs couches. Par exemple, un *routeur* implémentera les trois premières couches, un *switch* les deux premières et un *terminal* implémentera les sept couches. En pratique, les couches 5 et 6 sont confondues avec la couche 7.

Le modèle OSI impose que chaque couche soit indépendante et qu'elle ne puisse communiquer qu'avec les couches adjacentes. Le fait de rendre chaque couche indépendante permet de les faire évoluer indépendamment sans devoir remanier tout le modèle.

Prenons comme exemple la communication entre deux ordinateurs via un routeur :



Nous souhaitons aller sur le site <http://www.perdu.com>. Nous entrons l'adresse dans une **application** (couche 7). L'application s'occupe également de la **présentation** des données en créant une requête HTTP et de la création d'une **session** avec le serveur. La requête est ensuite transmise à la couche **transport**, qui va transporter la requête via un protocole (ici, le TCP). Cette couche va créer un **datagramme**, un train de données, et l'envoyer

sur le **réseau** à une destination (une adresse **IP**). Ce train parcourt la **liaison** entre les terminaux via le support **physique**, à savoir ici le câble.

Les données « brutes » arrivent au routeur, qui traite les données dans son stack, analyse l'IP de destination (en couche 3) et renvoie un train dans la bonne direction, vers le serveur.

Le serveur reçoit les données brutes, extrait les informations couche par couche jusqu'à l'application qui traite la requête (le serveur HTTP) et émet sa réponse.

Note

Le moyen mnémotechnique pour retenir les 7 couches du modèle OSI est :

Pour Le Réseau, Tout Se Passe Automatiquement.

3.1. Critique

La chose la plus frappante à propos du modèle OSI est que c'est peut-être la structure réseau la plus étudiée et la plus unanimement reconnue et pourtant ce n'est pas le modèle qui a su s'imposer. Les spécialistes qui ont analysé cet échec en ont déterminé quatre raisons principales :

3.1.1. Ce n'était pas le bon moment

Le modèle TCP/IP était déjà en phase d'investissement prononcé (lorsque le modèle OSI est sorti, les universités américaines utilisaient déjà largement TCP/IP avec un certain succès) et les industriels n'ont pas ressenti le besoin d'investir dessus.

3.1.2. Ce n'était pas la bonne technologie

Le modèle OSI est probablement trop complet et trop complexe. La distance entre l'utilisation concrète (l'implémentation) et le modèle est parfois importante. Au niveau de l'implémentation, TCP/IP est beaucoup plus optimisé et efficace. La plus grosse critique que l'on peut faire au modèle est qu'il n'est pas du tout adapté aux applications de télécommunication sur ordinateur ! Certains choix effectués sont en désaccord avec la façon dont les ordinateurs et les logiciels communiquent. La norme a fait le choix d'un « système d'interruptions » pour signaler les événements, et de langages de programmation de haut niveau, ce qui est peu réaliste et peu réalisable.

3.1.3. Ce n'était pas la bonne implémentation

Cela tient tout simplement du fait que le modèle est relativement complexe, et que les premières implémentations furent relativement lourdes et lentes. À l'inverse, la première implémentation de TCP/IP dans l'Unix de l'université de Berkeley était gratuite et relativement efficace. Les gens ont donc eu une tendance naturelle à utiliser TCP/IP.

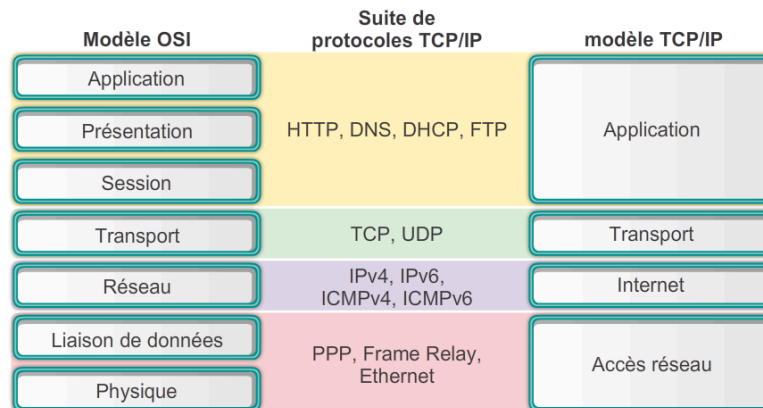
3.1.4. Ce n'était pas la bonne politique

Le modèle OSI a en fait souffert de sa trop forte normalisation. Les efforts d'implémentation du modèle étaient surtout bureaucratiques. À l'inverse, TCP/IP est venu d'Unix et a été tout de suite utilisé par des centres de recherches et les universités, c'est-à-dire les premiers à avoir utilisé les réseaux de manière poussée. Le manque de normalisation de TCP/IP a été contrebalancé par une implémentation rapide et efficace, et une utilisation dans un milieu propice à sa propagation.

4. Le modèle TCP/IP

Le modèle TCP/IP tire son nom de deux protocoles étroitement liés : le protocole de couche 3, **IP** et celui de couche 4, **TCP**.

Ce modèle réseau est constitué de quatre couches, contrairement à OSI. Il s'est peu à peu imposé comme modèle de référence, car il était déjà utilisé avant la création du modèle OSI.



- La couche *accès réseau* est un concept un peu particulier. Elle regroupe le support physique et la liaison de données. En réalité, cela s'explique en partie par l'utilisation du protocole **ETHERNET** qui mixe les deux couches.
- La couche *internet* est la couche la plus importante du modèle. Elle réalise la jonction entre les différents réseaux. Son rôle est d'injecter les trames IP dans un réseau et l'acheminement de ces paquets indépendamment les uns des autres jusqu'à destination, sans connexion préalable. Dans la pratique, cette couche est assurée par le protocole **IP**.
- La couche *transport* ne change pas de celle d'OSI : son rôle est de transporter les données d'un point à un autre. Pratiquement, cette couche est gérée par deux protocoles : **TCP** et **UDP**. Nous les verrons plus loin.

Voyons maintenant un exemple de données transmises via un réseau :

```
> Frame 3113: 113 bytes on wire (904 bits), 113 bytes captured (904 bits) on interface \Device\NPF_{EDF5FE41-8A39-44A4-B7C7-9739F2C92D65}, id 0
> Ethernet II, Src: ASRockIn_f0:47:1b (70:85:c2:f0:47:1b), Dst: Technico_a7:db:e3 (70:5a:9e:a7:db:e3)
> Internet Protocol Version 6, Src: 2a02:2788:4c6:cd8:f5a7:b21d:a9d3:b07, Dst: 2a00:1450:400e:80d::2005
> Transmission Control Protocol, Src Port: 52997, Dst Port: 443, Seq: 62229, Ack: 1966354, Len: 39
> Transport Layer Security
```

Chaque ligne représente une couche du modèle OSI :

- La première ligne nous informe sur le nombre de bits (904b) et d'octets (113o) qui composent le paquet. C'est ce qui est **physiquement** parvenu à l'interface réseau via un câble.
- La seconde ligne (qui commence par **Ethernet**) nous informe sur les adresses MAC sources et destinations, effectuant une **liaison de données** entre deux ordinateurs.
- La troisième ligne indique le protocole utilisé pour l'accès au **réseau**. On remarque ici que l'**IPv6** est utilisée.
- La quatrième ligne correspond à la couche **transport**. C'est le protocole **TCP** qui a été utilisé.
- La cinquième ligne correspond à la couche **session**. C'est le protocole de sécurisation des données **TLS** qui est en œuvre.

5. Le modèle Hybride

Si TCP/IP est parfait, pourquoi s'embêter avec le modèle OSI ? Tout d'abord, le modèle OSI est plus théorique et TCP/IP est un modèle pratique. OSI distingue clairement les notions de services, d'interfaces et de protocoles. Pour TCP/IP, la frontière est floue, simplement parce qu'on a utilisé les protocoles existants (TCP et IP) auxquels on a accolé une théorie justifiant leur utilisation.

En outre, les modèle OSI restreint l'utilisation du choix de connexion. Dans ce modèle, la couche de transport n'autorise qu'un mode connecté, ce que la couche TCP/IP ne permet pas. Et c'est là la force de TCP/IP : c'est l'*application* qui gère la connexion (via **TCP**) ou pas (via **UDP**). Dans le modèle OSI, cette distinction n'est pas possible.

C'est pour ces différentes raisons que l'on étudiera, de façon pédagogique, un modèle *hybride* qui se compose de cinq couches :



On voit que cette implémentation des modèles TCP/IP et OSI inclut une différenciation des couches physiques et de la liaison de données.

L'apprentissage des modèles peut sembler assez théorique, mais c'est un passage obligatoire dans l'apprentissage des réseaux.

5.1. La couche Physique

La couche **physique** a pour but de fournir un support de transport des données au format binaire. On utilise pour cela des câbles ou des ondes. La couche physique s'occupe de la transmission des bits de façon brute sur un canal de communication. Cette couche doit garantir la parfaite transmission des données. Concrètement, cette couche doit normaliser les caractéristiques *électriques* (un bit à **1** doit être représenté par une tension de +5V par exemple), les caractéristiques *mécaniques* (forme des connecteurs, de la topologie...), les caractéristiques *fonctionnelles* des circuits de données et les procédures d'établissement, de maintien et de libération du circuit de données.

5.1.1. Le codage PAM-5

Le codage **PAM-5**, ou *Modulation d'Impulsion en Amplitude à 5 Niveaux*, consiste à coder les bits de données sous forme de niveaux de tension distincts. Contrairement aux systèmes binaires simples qui n'utilisent que deux niveaux, comme le **NRZ** ou **PAM-2**, le PAM-5 peut transmettre plus de données dans le même laps de temps.

Il utilise les quatre paires de fils en mode *full duplex* combinés à différents algorithmes (**Modulation par Treillis Codé**, **décodeur de Viterbi** et **annulation d'écho**) pour envoyer **en un seul cycle d'horloge** 1 octet complet sur le réseau.

Le Gigabit Ethernet utilise toutes les quatre paires d'un câble Ethernet pour transmettre des données à pleine capacité (full-duplex) et atteindre un débit de 1 Gbit/s. Pour cela, il emploie la modulation PAM-5 (Pulse Amplitude Modulation with 5 levels), qui permet l'envoi d'information sur chaque paire à une vitesse de 125 millions de symboles par seconde.

Théoriquement, coder deux bits nécessite quatre niveaux de tension. Cependant, le PAM-5 utilise cinq niveaux pour inclure un cinquième niveau crucial pour la correction d'erreurs via la modulation par treillis (Trellis Coded Modulation). Ce cinquième niveau permet au récepteur de détecter et corriger les erreurs sans avoir besoin de retransmettre les données, améliorant ainsi la robustesse du signal même sur des câbles de qualité inférieure.

Le débit total est calculé comme suit : 125 millions de symboles par seconde multipliés par 2 bits par symbole et utilisés sur 4 paires de fils, ce qui donne un débit binaire de 1 Gb/s. Cette méthode assure une communication fiable et efficace en utilisant le cinquième niveau pour la correction d'erreurs sans impacter significativement la bande passante utile.

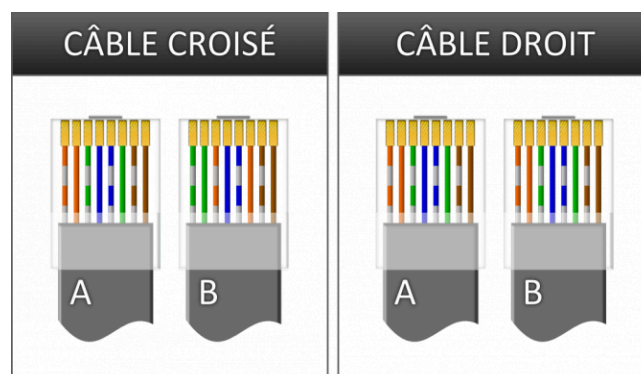
5.1.1.1. Le câble torsadé



Le câble torsadé est celui qui est le plus utilisé pour les réseaux LAN filaires. Il s'agit de huit fils torsadés deux par deux, chaque fil ayant une couleur et une fonction spécifique. Dans certains cas, ces câbles sont blindés par une armature métallique visant à les isoler des perturbations extérieures. Bien que communément appelé *câble RJ45*, ce n'est que la prise qui porte ce nom. Nous devrions appeler ce genre de câble des **8P8C** (8 positions et 8 contacts).

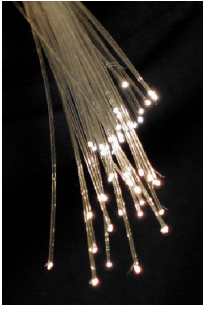
Il existe deux types de branchements : soit on parle de câble *droit*, soit on parle de câble *croisé*. La différence est au niveau de la prise 8P8C.

- Le câble *droit* est utilisé pour connecter un ordinateur à un switch ou à un routeur. Les deux extrémités du câble sont identiques.
- Le câble *croisé* permet de connecter deux ordinateurs directement, sans passer par un périphérique tiers. Le fait de croiser deux paires s'explique par le fait que l'émission des données se fait sur une paire et la réception sur l'autre.



Heureusement, la technologie aidant, on peut désormais utiliser un câble droit ou un câble croisé sans se poser de questions. Les équipements vont s'adapter pour faire en sorte de recevoir et renvoyer les données sur les bonnes paires. Toutefois, les câbles croisés sont de plus en plus rares.

5.1.1.2. La fibre optique



Une fibre optique est un fil dont l'âme est en plastique ou en verre très fin (plus fin qu'un cheveu) et qui a la propriété de conduire la lumière. Le principe repose sur la réfraction pour garder le rayon lumineux à l'intérieur de la fibre.

On distingue deux catégories de fibres :

- la fibre **monomode** : adaptée à une seule longueur d'onde, elle ne peut véhiculer qu'un seul rayon lumineux (généralement rouge).
 - la fibre **multimode** : adaptée à la lumière blanche, elle peut véhiculer des
- ondes de différentes couleurs.

Dans la pratique, on utilisera de la fibre **monomode** pour les connexions **WAN**, car l'atténuation du signal est plus faible qu'en *multimode*. Un faisceau en monomode peut transporter l'information sur plus de 100 km, alors qu'un faisceau en multimode est limité à 2 km. Néanmoins, on privilégiera la fibre **multimode** pour la connexion des réseaux **LAN**.

Avec la fibre optique, on peut atteindre des débits de l'ordre de 60 Gpbs.

5.1.1.3. Les réseaux sans fil

Le Wifi est la méthode d'accès la plus courante à un réseau sans fil. Le câble est remplacé par une ou plusieurs ondes radio-électriques. Les réseaux sans fil permettent de relier très facilement des équipements distants d'une dizaine de mètres à quelques kilomètres.

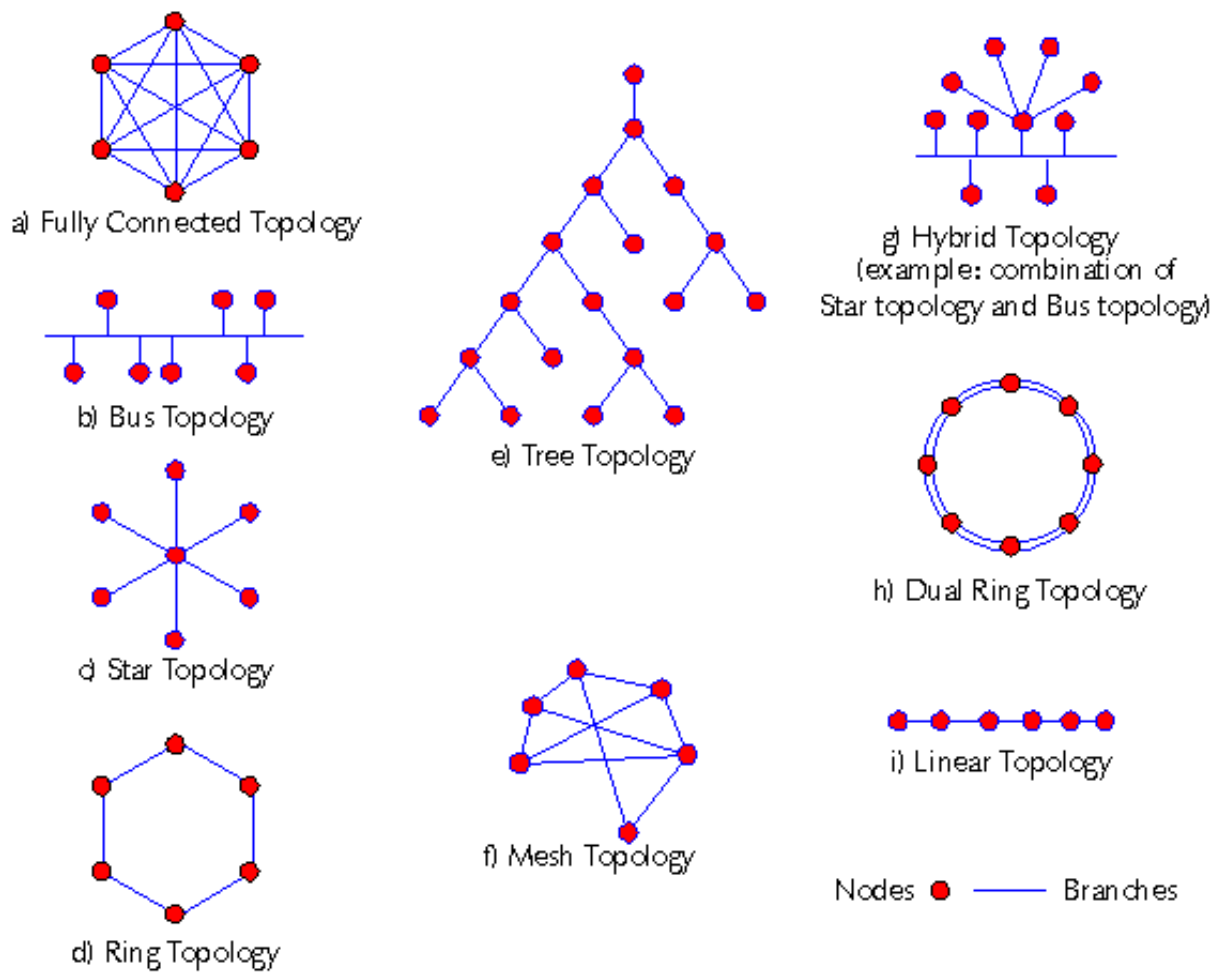
En contrepartie, les ondes radio sont sensibles aux interférences et les ondes hertziennes sont plus aisément **piratables**.

Les réseaux sans fil sont répartis en plusieurs catégories :

- le **WPAN**, réseau sans fil personnel, utilisé principalement par la norme *Bluetooth* ;
- le **WLAN**, plus communément appelé le Wifi ;
- Le **WWAN**, regroupant les systèmes GSM, la 4G, la 5G...

5.1.2. Les topologies

La couche physique assure le transport de la communication, mais aussi organise cette communication entre les machines sur le réseau. Elle peut définir la façon dont les équipements sont interconnectés et la représentation spatiale du réseau. Elle peut aussi définir la manière dont les données transitent dans les lignes de communication. On parlera alors de *topologie logique*.



Il existe deux modes de propagation des données, classant les différentes topologies :

Mode de diffusion : Ce mode de fonctionnement consiste à n'utiliser qu'un seul support de transmission. Le principe est que le message est envoyé sur le réseau et que toute unité sur ce réseau est capable de voir le message et d'analyser le *header* afin de savoir si le message lui est adressé ou non. Ce mode est surtout utilisé dans les topologies en bus ou en anneau.

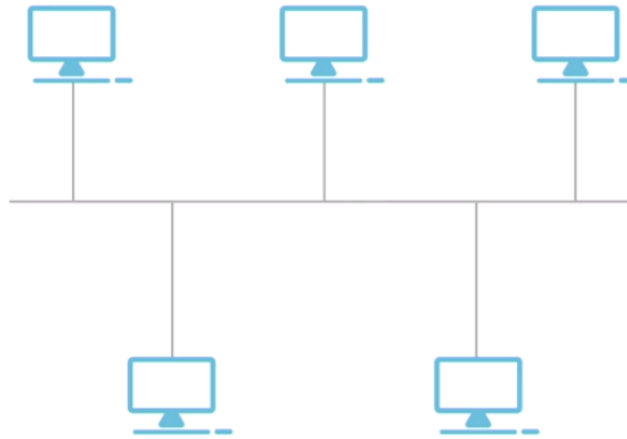
Mode point à point : Dans ce mode, le support physique ne relie qu'une paire d'unités seulement. Pour que les deux unités réseaux communiquent entre elles, elles passent obligatoirement par un *nœud*. Ce mode est notamment utilisé dans les topologies maillées ou étoilées.

Pour les réseaux PAN et LAN, on utilisera essentiellement les topologies en anneau ou en bus. On privilégiera les topologies maillées, étoilées ou hybrides pour les réseaux MAN et WAN.

5.1.2.1. Le bus

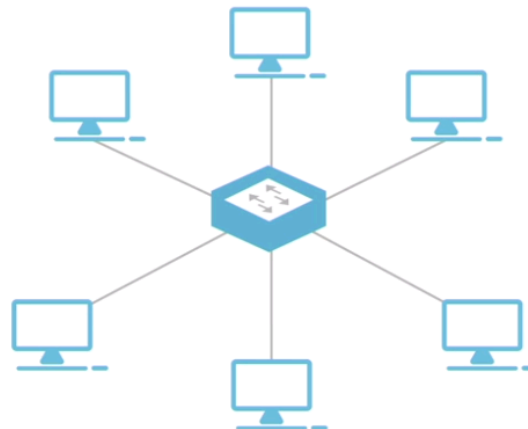
La topologie en bus repose sur un câblage sur lequel viennent se connecter des nœuds. Ce câble est l'unique élément matériel constituant le réseau et seuls les nœuds génèrent des signaux. La quantité de câbles utilisés est minimale et ne nécessite pas de point central.

L'inconvénient majeur est qu'une seule coupure dans le câble empêche tout nœud d'échanger des informations sur le réseau et que seul un nœud ne peut émettre un signal à la fois. Cette topologie est la plus ancienne et n'est pratiquement plus employée à ce jour.



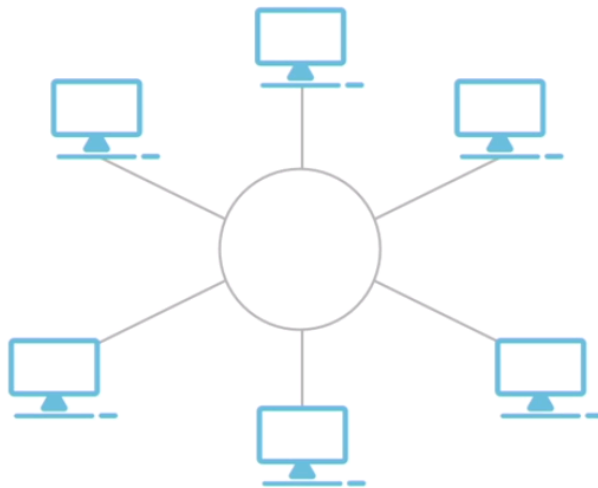
5.1.2.2. L'étoile

La topologie en étoile repose sur des matériels actifs. Dans une topologie en étoile, les ordinateurs du réseau sont reliés à un système matériel central nommé **concentrateur**. Cette topologie est plus coûteuse, car elle nécessite plus de câbles et de commutateurs, mais elle est aussi plus résistante aux pannes. Un nœud ou un câble défectueux ne fera pas tomber tout le réseau. C'est la topologie la plus employée actuellement.



5.1.2.3. L'anneau

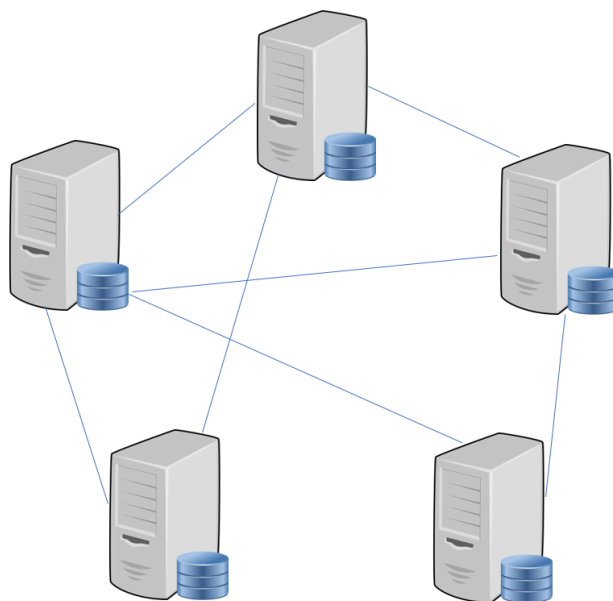
Autre topologie ancienne, mais encore utilisée, la topologie en anneau est similaire à la topologie en bus dans le sens où elle nécessite un câble **dorsal** (principal). La différence est que cette dorsale est connectée à elle-même pour former un anneau. Gros inconvénient cependant, seul un nœud ne peut communiquer à la fois (comme pour le bus). Un *jeton* va parcourir l'anneau et seul le nœud possédant le jeton peut communiquer sur l'anneau. De même, un seul nœud ou câble endommagé provoque la panne de l'ensemble du réseau.



En pratique, les nœuds sont reliés entre eux à un répartiteur appelé *MAU* qui va gérer la communication entre les nœuds en leur imposant un temps de parole. Physiquement, la topologie sera étoilée, mais logiquement, la topologie sera en anneau.

5.1.2.4. Le maillage

La topologie maillée est une évolution de la topologie en étoile. Elle correspond à plusieurs liaisons point à point. Chaque nœud est connecté à chaque autre nœud en liaison point à point. Cette topologie est onéreuse, car chaque nœud doit avoir autant de carte réseau et autant de câble que le nombre de nœuds dans le réseau.

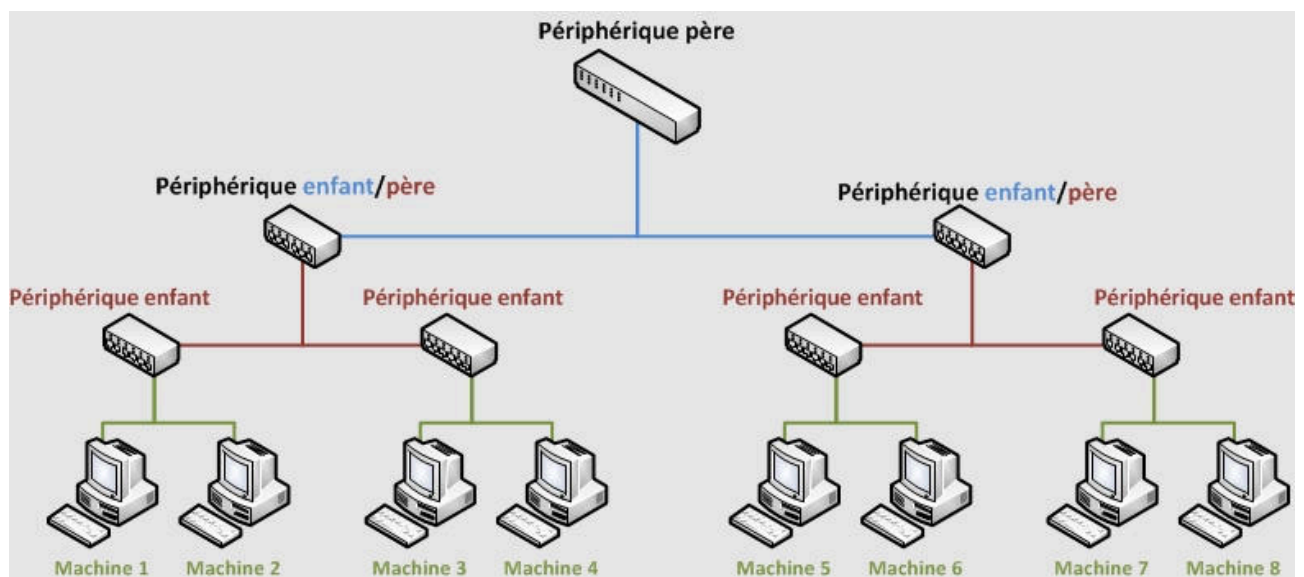


Cette topologie n'est quasi jamais employée dans les réseaux LAN. Par contre, elle est utilisée dans les réseaux WAN et le réseau Internet.

5.1.2.5. L'arbre

La topologie en arbre est une dérivée de l'étoile. Elle consiste à prendre la topologie étoile centrale et à connecter d'autres unités satellites aux nœuds principaux (branches). Cette topologie permet de combiner les avantages des topologies maillées, bus et étoilée. En effet, si un câble principal est défectueux, le réseau reste fonctionnel tant qu'une branche ou une sous-arête ne l'est pas. De plus, la quantité totale de câbles utilisés est

importante, mais inférieure à celle du maillage complet ; et enfin, cette topologie autorise des défaillances ponctuelles sans que cela impacte le reste du réseau.



5.1.3. Le Hub



Le *hub* ou le répéteur en français est une pièce d'équipement de la topologie en étoile visant à connecter plusieurs terminaux entre eux. Le hub est un équipement de connexion de couche 1, c.-à-d. qu'il gère uniquement les données brutes. Il ne fait qu'amplifier le signal reçu sur une de ses portes et transfère ce signal à toutes les autres portes sans distinction.

L'inconvénient d'un hub est justement sa capacité à répéter les informations sur toutes ses portes. Plus il y a d'utilisateurs connectés sur les portes, plus ils émettent – et reçoivent – des données. Ce volume de données finit par créer des *collisions de paquets*, créant de ce fait un engorgement du hub et son ralentissement.

Les hubs sont incapables d'émettre et de recevoir des données en même temps (connexion *half-duplex*). Ce type d'équipement n'est plus utilisé de nos jours. Il est remplacé par un équipement de couche 2, le **switch**.

5.1.4. Les protocoles CSMA/CD et CSMA/CA

Nous venons de le voir, plus il y a d'ordinateurs sur une topologie en bus ou en étoile, plus le risque de créer des collisions de paquets augmente. Pour éviter ces collisions, il existe deux types d'accès (basé sur la concentration des données) qui sont couramment utilisés dans les réseaux modernes :

le CSMA/CD : Méthode de détection de collision par détection de porteuse. Cette méthode est utilisée par les systèmes Ethernet câblés.

le CSMA/CA : Méthode de prévention des collisions, surtout employée dans les réseaux wifi.

5.1.4.1. Le CSMA/CD

Le CSMA/CD signifie **Carrier Sense Multiple Acces / Collision Detection**. Son but est de gérer les collisions de paquets. Une collision de paquets survient quand deux ordinateurs parlent en même temps. Le Hub va répéter les informations sur toutes ses portes, et

fatalement, les deux transmissions vont entrer en collision. CSMA/CD va tenter de limiter le nombre de collisions en appliquant une série de règles :

1. Chaque terminal écoute en permanence le câble pour savoir si quelqu'un parle ou non ;
2. Le terminal ne peut parler que si personne d'autre ne parle ;
3. En cas de collision, les deux machines ayant parlé en même temps doivent se taire et attendre un temps aléatoire (quelques ms) ;
4. Une fois le temps attendu, les machines peuvent reparler en suivant les règles.

L'astuce réside dans le temps aléatoire. Les deux terminaux vont attendre un certain temps et il y a très peu de chances que le temps soit exactement le même ! Fatalement, une des deux machines va parler avant l'autre, sauf si, bien sûr, une troisième a pris la parole entre-temps.

5.1.4.2. Le CSMA/CA

Le CSMA/CA signifie /Carrier Sense Multiple Access/Collision Avoidance/. Il fonctionne de manière similaire à son homologue. La principale différence est que, au lieu de parler directement, un terminal va d'abord envoyer un signal d'avertissement aux autres nœuds. Les autres nœuds entendent ce paquet et vont éviter de parler. Une fois l'avertissement passé, l'émetteur va (enfin) délivrer son message. Chaque nœud verra le message passer et pourra dès lors émettre son propre paquet d'avertissement. En cas de collision, le CSMA/CD sera employé.

5.2. La couche Liaison

La couche 2 est la couche *liaison de données*. Elle a pour rôle d'assurer la connexion des machines sur un réseau local. Elle fournit différents moyens pour transférer des données entre différents terminaux et peut aussi détecter et corriger certaines erreurs induites par la couche physique.

La couche 2 s'occupe du traitement des *trames* émises par les terminaux présents sur le réseau. Les trames de couche 2 ne franchiront pas les limites du réseau LAN.

5.2.1. Le réseau ethernet et l'adresse mac

Il existe un identifiant de chaque matériel ou machine : l'adresse **MAC** (Medium Access Control). C'est un identifiant unique stocké dans une interface réseau. Elle est utilisée pour attribuer une adresse à la couche de liaison. Théoriquement, cette adresse ne peut pas être modifiée.

Une adresse MAC est composée de **48 bits**, soit 6 octets écrits en hexadécimal, les octets étant séparés par le caractère « : ». Lorsque le constructeur a épuisé toutes les adresses qui lui étaient destinées, un nouveau triplet constructeur lui est associé.

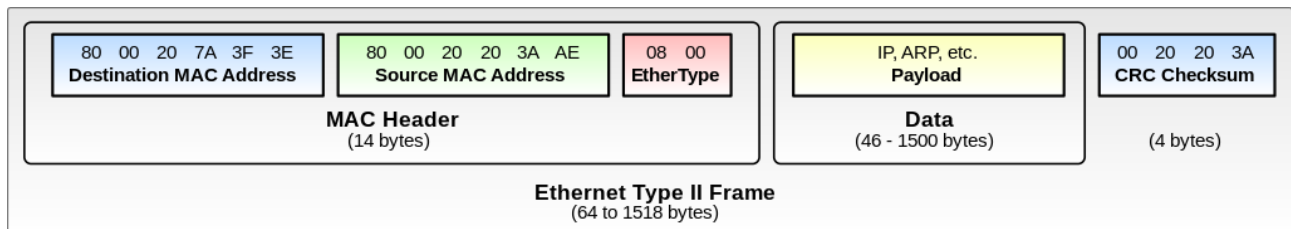
En détails, les 48 bits sont découpés comme suit :

- 1 bit I/G** : Ce bit indique si l'adresse est individuelle (0) ou fait partie d'un groupe d'adresse (1) (typiquement pour les switches) ;
- 1 bit U/L** : Ce bit indique si l'adresse est universelle (0) ou administrée localement (1) ;
- 22 bits réservés** : Chaque constructeur d'équipement réseau dispose de 22 bits d'identification uniques ;
- 24 bits d'adresse unique** : Ces bits identifient la carte réseau. C'est l'équivalent du numéro de série.

Un cas particulier est l'adresse **FF:FF:FF:FF:FF:FF** qui est l'adresse de *broadcast*, permettant d'adresser toutes les machines d'un même réseau local.

5.2.2. La trame ethernet

Le protocole le plus employé sur la couche 2 est le protocole **ethernet** (norme 802.3). Une trame ethernet doit avoir une taille d'au moins 64 octets pour que la détection de collisions fonctionne et peut avoir une taille maximale de 1518 octets. Le paquet commence toujours par un *préambule*, qui contrôle la synchronisation entre l'émetteur et le récepteur, et un *Start Frame Delimiter*, qui définit la trame.



La trame contient des informations sur les adresses sources et de destination de la trame, des informations concernant l'*ethertype* (toujours **II**) et la taille de la trame, ainsi que le paquet de données. Enfin, le FCS (Frame Check Sequence) ferme la trame comme un CRC. Le paquet est terminé par un **Inter Frame Gap** qui définit une pause de transmission de 9,6µs.

5.2.3. Le switch

Le *switch*, ou le *commutateur* en français, est un matériel qui s'occupe exclusivement de la couche 2. Un commutateur est boîtier sur lequel sont présentes plusieurs prises 8P8C femelles permettant de brancher dessus des machines à l'aide de câbles à paires torsadées. Un switch decode l'entête de trame pour ne l'envoyer que vers un port Ethernet associé, ce qui réduit le trafic sur l'ensemble du câblage réseau par rapport à un **hub** qui renvoie les données sur tous les ports.

Chaque switch utilise une table de correspondance MAC-n° de connexion, qui fait l'association entre un port du switch et une adresse MAC. Cette table est appelée la **table CAM**, construite dynamiquement. Le switch va apprendre, au fur et à mesure qu'il voit passer les trames, quelle machine est branchée à quel port.

Admettons que deux ordinateurs dialoguent entre eux. **PC1** envoie un message à **PC2**. Deux cas se présentent dès lors :

- Si le switch connaît l'adresse MAC de **PC2**, il redirige vers le port associé. Dans le même temps, il met à jour sa table CAM avec l'adresse MAC de **PC1** et son port de connexion.
- Si le switch ne connaît pas l'adresse MAC de **PC2**, il envoie la trame à **tout le monde**. Les ordinateurs qui ne sont pas concernés ne traiteront pas la trame. Quand **PC2** répondra, le switch se mettra à jour.

Le switch introduit aussi une évolution de la couche **PHY**. En effet, puisqu'il redirige intelligemment les données, contrairement au hub, les collisions sont moindres ! Cela s'explique par le fait que le switch envoie les trames d'une porte à l'autre sur les paires torsadées, et selon la topologie en étoile, chaque terminal est en point à point sur une ligne. De ce fait, le protocole CSMA/CD est moins sollicité, ce qui augmente les débits de transfert. On dit alors que le switch est en mode **full duplex**.

5.3. La couche réseau

Le rôle de la couche 3 (**NET**) est majeur : c'est cette couche qui interconnectera les réseaux entre eux. Son rôle est d'utiliser la couche 2 pour transmettre un paquet d'un réseau à un autre. Pour ce faire, on utilisera des *routeurs* et des *protocoles de routage*.

5.3.1. Les protocoles

5.3.1.1. Le protocole IP

Une adresse IP est un numéro d'identification *logique* qui est attribué de façon permanente ou provisoire à chaque périphérique relié à Internet. Ce numéro est totalement indépendant de l'adresse MAC.

Deux ordinateurs communiquent entre eux au moyen de leur carte réseau. Ils envoient des *trames* contenant l'adresse de destination, celle de l'émetteur et le message à transférer. Ces trames sont codées de manière précise.

Lorsqu'on doit gérer des réseaux étendus et complexes comme le réseau Internet, il est indispensable de pouvoir grouper les ordinateurs (appelés *hôtes*) d'un réseau en sous-réseaux organisés hiérarchiquement.

Le système des adresses IP est adapté pour structurer facilement un réseau en multiples sous-réseaux et pour contrôler l'unicité des noms de chaque hôte. En effet, il ne peut pas y avoir deux fois la même adresse IP !

Ce système permet aussi de faciliter les protocoles de routage.

5.3.1.1.1. Classes d'adresses

La communauté Internet a défini cinq **classes d'adresses** appropriées à des réseaux de différentes tailles. Il y a, a priori, peu de réseaux de grande taille (classe *A*). Il y a plus de réseaux de taille moyenne (classe *B*) et beaucoup de réseaux de petite taille (classe *C*). La taille du réseau est exprimée en nombre d'hôtes potentiellement connectés.

Toutefois, en 1992, la notation en classes n'est plus adaptée à la taille croissante d'Internet et la **RFC 1338** abolit cette notion de classe au profit d'une notion plus adaptée, le **CIDR** (*Classless Inter-Domain Routing*), afin de diminuer la taille de la table de routage contenue dans les routeurs.

Un fournisseur d'accès peut dès lors se voir attribuer un *bloc d'adresses* afin d'y créer des sous-réseaux de tailles variables en fonction des besoins. Seul le bloc sera visible de l'extérieur (donc sur le Web), ce qui réalise une *agrégation* d'adresses. De plus, avec le CIDR, le masque de sous-réseau s'écrit en notation abrégée (par exemple, /24).

5.3.1.1.2. Adressage IPv4

Lorsqu'on doit gérer des réseaux étendus et complexes, il est indispensable de pouvoir grouper les ordinateurs (appelés *hôtes*) d'un réseau en sous-réseaux organisés hiérarchiquement.

Le système des adresses IP est adapté pour structurer facilement un réseau en multiples sous-réseaux et pour contrôler l'unicité des adresses de chaque hôte.

Une adresse IP est constituée de **32 bits**, souvent regroupés en 4 octets écrit en décimal (de 0 à 255), comprenant :

- l'adresse IP de l'hôte (son identifiant sur le réseau) ;

- le masque de sous-réseau (qui qualifie le réseau);

Cette approche permet donc de :

- segmenter un réseau en un ou plusieurs sous-réseaux;
- d'attribuer une adresse de sous-réseau unique à chaque sous-réseau;
- d'attribuer un numéro d'hôte unique à chaque ordinateur branché sur ces sous-réseaux.

Seulement, si on part du principe original que chaque interface réseau possède sa propre adresse unique, on remarque qu'il n'y a que 2^{32} adresses disponibles pour le monde entier. Or, nous possédons plusieurs périphériques connectés plus ou moins en permanence à Internet. Il y a trop de périphériques par rapport au nombre d'adresses disponibles.

Pour résoudre ce problème, nous utilisons des adresses dites *privées*. Mais malgré cela, le nombre d'adresses publiques est arrivé à saturation. C'est pourquoi, depuis quelques années, nous utilisons de plus en plus un protocole IP plus évolué, l'IPv6.

5.3.1.1.3. Adressage IPv6

Une adresse IPv6 est une adresse de 128 bits, soit 16 octets. Le nombre d'adresses disponibles (2^{128}) est tel qu'il faudrait placer **667 milliards** d'appareils connectés par **mm²**! Pour avoir un tel ordre de grandeur, l'Internet mondial actuel tient à peine dans $\frac{1}{5}$ de l'IPv6.

La notation décimale est abandonnée au profit de l'hexadécimal et les groupes d'octets sont désormais séparés par des « : ».

5.3.1.1.4. Adresses réservées

Certaines adresses IP sont réservées à un usage spécifique et ne sont donc pas routables sur Internet.

0.0.0.0 (::0 en IPv6) : Cette adresse réservée est l'adresse de **route par défaut**. Elle est de type **0.x.x.x**. Tous les paquets envoyés à un réseau inconnu seront redirigés vers cette adresse. Cette adresse est aussi utilisée par les nœuds pour connaître leur adresse IP lors d'un appel au serveur *DHCP*.

127.0.0.1 (ou ::1 en IPv6) : Cette adresse est l'adresse de **bouclage** (*loopback* ou *localhost* en anglais). Cette adresse sert principalement à tester le fonctionnement des cartes réseaux et sert aussi au fonctionnement de clients/serveurs sur l'ordinateur.

l'adresse de réseau : Cette adresse désigne **tous les postes disponibles** sur un réseau. Elle est utilisée dans les tables de routages. Elle s'obtient en positionnant *tous les bits réservés aux hôtes* à 0 dans le masque.

l'adresse de diffusion : Cette adresse désigne aussi **tous** les postes du réseau, à ceci près qu'elle permet de diffuser un message à tous les nœuds. Elle se nomme généralement adresse de **broadcast**. Elle s'obtient en positionnant *tous les bits réservés aux hôtes* à 1 dans le masque.

les adresses privées Ces adresses sont utilisées pour construire un réseau privé. Elles ne sont pas routables sur Internet. Les différentes plages sont :

| Plage d'adresses | de | à | nb de réseaux max | nb d'hôtes max |
|------------------|-------------|----------------|-------------------|----------------|
| 10.0.0.0/8 | 10.0.0.1 | 10.255.255.254 | 4 | 16 777 214 |
| 172.16.0.0/16 | 172.16.0.1 | 172.16.255.254 | 16 | 65 534 |
| 192.168.0.0/24 | 192.168.0.1 | 192.168.0.254 | 256 | 254 |

5.3.1.1.5. Masque de sous-réseaux

Un masque de sous-réseau permet de distinguer les bits d'une adresse IP utilisés pour identifier le sous-réseau de ceux utilisés pour identifier l'hôte. L'adresse de sous-réseau est obtenue en appliquant l'opérateur booléen (**ET**) entre l'adresse IP et le masque.

Le masque de sous-réseaux est une série de chiffres similaire à une adresse IP, à ceci près qu'il s'agit en binaire d'une suite continue de bits à 1, représentant la partie *réseau*, suivi d'une suite continue de bits à 0, représentant la partie *hôtes*. En notation CIDR, on indique le nombre de bits à 1.

Un masque ne peut pas être « troué », ce qui signifie qu'il ne peut pas y avoir de 0 dans la suite de bits à 1.

En réalité, le masque ne sert qu'aux routeurs, afin de séparer les réseaux. Si deux ordinateurs sont situés dans le même réseau local, on sait qu'ils peuvent dialoguer via la couche 2. Par contre, si ce sont des réseaux différents, le système envoie les données en couche 3, pour recréer une trame capable de transmettre les données. C'est le rôle du **routeur**.

5.3.1.2. Le protocole ARP

Le protocole ARP (Address Resolution Protocol) est un protocole un peu spécial : il est à la fois un protocole de couche 2 **et** de couche 3. Son rôle est de faire correspondre une adresse IP avec une adresse MAC.

5.3.1.2.1. La table ARP

La table ARP est une table de données qui vise à limiter l'envoi des requêtes ARP. En effet, sans cette table, chaque datagramme IP serait précédé d'un datagramme ARP. La table ARP conserve les associations MAC/IP.

Or, **comment** l'ordinateur peut-il connaître l'adresse MAC du routeur ? L'ordinateur peut la demander au routeur, mais pour lui faire la demande, il doit pouvoir lui parler, et pour lui parler, il faut connaître son adresse MAC !

Heureusement, le protocole **ARP** va résoudre le problème. L'ordinateur va émettre une requête ARP en broadcast, c.-à-d. à destination de toutes les adresses du réseau, en demandant à *qui appartient l'adresse IP 192.168.0.1*.

Note

L'adresse MAC d'un broadcast est toujours 00:00:00:00:00:00.

L'équipement qui se reconnaît (ici, le **routeur**) va émettre alors une réponse ARP vers le demandeur en lui donnant son adresse MAC. Les deux équipements vont écrire les données reçues dans la table ARP.

Note

Faites le test! Dans l'invite de commande Windows, tapez `arp -a`.

Note

Les informations contenues dans la table ARP sont limitées dans le temps.

5.3.1.3. Le protocole ICMP

Le protocole **ICMP** (Internet Control Message Protocol) est considéré comme un protocole de couche 3, même si pratiquement il s'encapsule dans le datagramme IP. Il permet aux routeurs de gérer les informations relatives aux erreurs aux machines connectées sur le réseau.

5.3.2. Le routeur



Le but du routeur est de relier les réseaux entre eux et de choisir automatiquement le meilleur chemin pour transférer les paquets. Pour pouvoir être gérés, les routeurs sont dotés de connecteurs physiques. Ces connecteurs sont appelés **ports de gestion**.

Contrairement aux interfaces Ethernet et série, les ports de gestion ne sont pas utilisés pour le transfert des paquets. Le port de gestion le plus courant est le /port de console/. Le port de console est utilisé pour connecter un terminal ou, plus fréquemment, un PC exécutant un logiciel émulateur de terminal, afin de configurer le routeur sans qu'il soit nécessaire d'accéder au réseau. Le port de console doit être utilisé pendant la configuration initiale du routeur.

Le routeur dispose de plusieurs **interfaces réseaux** indépendantes (contrairement aux switches). Chaque interface réseau sera connectée à un réseau en particulier. De ce fait, un routeur dispose de plusieurs adresses MAC.

Le routeur possède aussi en mémoire une table de routage. Cette table va contenir une liste d'autres routeurs auxquels on pourra transférer la trame. Cette trame sautera de routeur en routeur jusqu'à destination.

5.3.2.1. Notion de route

Une *route* en langage réseau est l'équivalent à une route en langage routier: c'est une voie d'accès entre deux points. Sauf qu'à la place des villes, ce sont des routeurs.

La **route par défaut** est une notion plus importante. C'est en effet le trajet que doit suivre un paquet si le routeur ne connaît pas l'adresse de destination. On parlera alors de **passerelle**. Typiquement, votre routeur à la maison fait office de passerelle vers Internet. Il possède deux interfaces réseau indépendantes: une connectée à votre réseau local (en réalité, le routeur domestique intègre un switch sur cette interface réseau) et l'autre connectée au modem.

5.3.2.1.1. Exemple

Nous souhaitons accéder au site <http://www.perdu.com>. Nous verrons plus loin, avec le protocole /DNS/, que ce nom cache l'IP 208.97.177.124 (cette adresse peut varier). Nous

ne connaissons pas l'adresse MAC du serveur, et même si nous la connaissions, il serait impossible de l'atteindre, car nous ne sommes pas sur le même réseau physique. Pour l'exemple, notre adresse est 192.168.0.15/24, d'adresse MAC 00:11:22:33:44:55.

1. On veut accéder à 208.97.177.124. Or, ce n'est pas sur notre réseau, donc impossible de dialoguer avec l'adresse MAC. Donc, direction la *route par défaut* du routeur. Pour cela, on va passer par la *passerelle*, qui est l'adresse IP du routeur. Ça tombe bien, le routeur étant dans le réseau local, on peut passer un datagramme de couche 2. La trame Ethernet ressemblera à ceci :

| Adr MAC routeur | Adr MAC source | Type | En-tête IP | IP source | IP dest | Données | CRC |
|-------------------|-------------------|--------|------------|--------------|----------------|---------|-----|
| AA:BB:CC:DD:EE:FF | 00:11:22:33:44:55 | 0x8000 | XXX | 192.168.0.15 | 208.97.177.124 | XXX | FCS |

Notez que l'adresse MAC est celle du routeur !

1. Le routeur va lire le datagramme. C'est bien son adresse MAC, l'ethertype est 0x8000, donc les données sont de type IP. il envoie le datagramme IP en couche 3 pour traitement. De là, il regarde l'adresse IP de destination. En fonction de sa table de routage, il va décider par quel routeur passer. De plus, en voyant passer les datagrammes, il va ajouter dans sa table de routage les données manquantes. Une fois que son choix est fait, il recrée une trame de couche 2 :

| Adr MAC dest | Adr MAC source | Type | En-tête IP | IP source | IP dest | Données | CRC |
|-------------------|-------------------|--------|------------|--------------|----------------|---------|-----|
| 02:CF:D1:6C:05:03 | AA:BB:CC:DD:EE:FF | 0x8000 | XXX | 192.168.0.15 | 208.97.177.124 | XXX | FCS |

1. Le prochain routeur reçoit le datagramme et le traite, faisant progresser le paquet jusqu'à destination.
2. Une fois arrivé au serveur, il traitera les données et répondra à l'ordinateur en faisant le trajet inverse. Il a pour ça toutes les informations nécessaires, puisque l'IP source n'a jamais été modifiée.

5.4. La couche Transport

5.4.1. Fonctionnement

Jusqu'ici, nous avons vu que chaque couche disposait de sa propre adresse de communication. Pour la couche 4, cette adresse spécifique se nomme le **port**. Le port est donc l'adresse utilisée par une application pour « écouter » les demandes des clients.

Les ports les plus connus sont:

80 : le port web (HTTP);

25 : le port mail (SMTP);

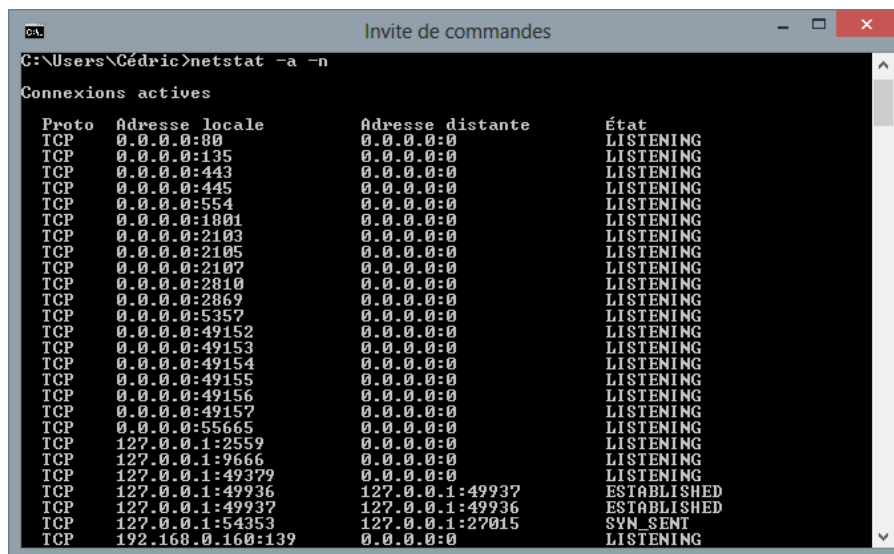
143 : le port IMAP;

443 : le port HTTPS;

20 et 21 : les ports FTP;

53 : le port DNS.

Un ordinateur peut à la fois être client et serveur. Un simple `netstat -an` dans le terminal affichera la liste des services en cours:



```
C:\Users\Cédric>netstat -a -n

Connexions actives

Proto  Adresse locale      Adresse distante    État
TCP    0.0.0.0:80           0.0.0.0:0           LISTENING
TCP    0.0.0.0:135          0.0.0.0:0           LISTENING
TCP    0.0.0.0:443          0.0.0.0:0           LISTENING
TCP    0.0.0.0:445          0.0.0.0:0           LISTENING
TCP    0.0.0.0:554          0.0.0.0:0           LISTENING
TCP    0.0.0.0:1801         0.0.0.0:0           LISTENING
TCP    0.0.0.0:2103         0.0.0.0:0           LISTENING
TCP    0.0.0.0:2105         0.0.0.0:0           LISTENING
TCP    0.0.0.0:2107         0.0.0.0:0           LISTENING
TCP    0.0.0.0:2810         0.0.0.0:0           LISTENING
TCP    0.0.0.0:2869         0.0.0.0:0           LISTENING
TCP    0.0.0.0:5357         0.0.0.0:0           LISTENING
TCP    0.0.0.0:49152        0.0.0.0:0           LISTENING
TCP    0.0.0.0:49153        0.0.0.0:0           LISTENING
TCP    0.0.0.0:49154        0.0.0.0:0           LISTENING
TCP    0.0.0.0:49155        0.0.0.0:0           LISTENING
TCP    0.0.0.0:49156        0.0.0.0:0           LISTENING
TCP    0.0.0.0:49157        0.0.0.0:0           LISTENING
TCP    0.0.0.0:55665        0.0.0.0:0           LISTENING
TCP    127.0.0.1:2559        0.0.0.0:0           LISTENING
TCP    127.0.0.1:9666        0.0.0.0:0           LISTENING
TCP    127.0.0.1:49379      0.0.0.0:0           LISTENING
TCP    127.0.0.1:49936      127.0.0.1:49937     ESTABLISHED
TCP    127.0.0.1:49937      127.0.0.1:49936     ESTABLISHED
TCP    127.0.0.1:54353      127.0.0.1:27015     SYN_SENT
TCP    192.168.0.160:139    0.0.0.0:0           LISTENING
```

Les ports sont codés en décimal sur deux octets, ils peuvent donc prendre 2^{16} valeurs, soit 65536 valeurs. Les 1024 premiers ports sont réservés aux applications « standards », même si de plus en plus d'applications bloquent les ports au-delà de 1024. Pour que la communication s'établisse, les clients doivent aussi avoir un n° de port. Ce port ne sera pas réservé et sera attribué automatiquement par le système d'exploitation.

5.4.1.1. Exemple

Prenons un navigateur se connectant à un site (Google par exemple). Le navigateur est un client qui interroge par défaut le port 80 d'un serveur situé sur une IP.

Avant de se connecter au serveur de Google, l'OS va spécifier aléatoirement un port de connexion (par ex. le port **12345**). Ce port sera utilisé par Google pour envoyer les réponses au navigateur. Ce dernier va faire une demande au serveur en envoyant son port de connexion.

Le serveur enverra en réponse l'autorisation de dialoguer, ainsi qu'un n° de port (au-dessus de 1024) dédié à cette communication.

Par la suite, le client et le serveur vont dialoguer sur leurs ports dédiés, libérant le port 80 du serveur qui attendra une autre demande.

5.4.2. Protocoles

Cette couche utilise deux protocoles distincts pour fonctionner, *TCP* et *UDP*. La différence entre ces deux protocoles se justifie par des besoins différents.

Certaines applications nécessitent un transport fiable des données, au détriment de la vitesse; d'autres vont privilégier le transport immédiat de la donnée, quitte à perdre des informations en chemin. Il est plus rapide de renvoyer un paquet perdu que de demander une confirmation à chaque paquet émis!

5.4.2.1. TCP

TCP, pour *Transport Control Protocol*, est le protocole le plus utilisé. Il permet le transport fiable des paquets. La plupart des applications sur le Net utilisent TCP pour fonctionner.

TCP est un protocole dit **connecté**. Il reste en contact avec son correspondant jusqu'à la fin de la transmission. Le principe de TCP est de contrôler que chaque octet émis soit correctement reçu. Avant de commencer à dialoguer, TCP va établir une connexion au moyen d'un système nommé *three way handshake*. Cet échange permet de synchroniser les n° de séquences afin d'assurer la transmission fiable des données. Les n° de séquences permettent de remettre les paquets dans l'ordre et de s'assurer qu'aucun d'entre eux ne s'est perdu en chemin.

5.4.2.2. UDP

UDP, pour User Data Protocol, est un protocole simplifié, plus rapide que TCP, mais aussi moins fiable. Il est surtout employé dans les émissions de données en streaming.

À contrario, UDP est dit non-connecté. Il se fiche de savoir si le destinataire reçoit toutes les informations. Il émet, c'est tout!

5.4.2.3. QUIC

Le protocole *QUIC* (Quick UDP Internet Connections) est une alternative intéressante à TCP/IP. Il permet d'obtenir un meilleur débit et une réduction des temps de latence. De plus, il propose la possibilité d'établir plusieurs connexions simultanées sur le même port physique. Conçu par Google, ce protocole basé sur *UDP* permet de réduire les latences grâce à la transmission multiplexée, qui permet d'envoyer plusieurs flux simultanément sans établir de connexion TCP.

Le *three way handshake* est remplacé par une connexion multiplexée dès la phase de chiffrement (via TLS 1.3). Ce nouveau protocole est utilisé dans le futur standard **HTTP/3**.

5.4.3. Le Firewall

Un firewall est un équipement spécialement conçu pour faire respecter la sécurité des communications transitant sur un réseau. Le firewall va « masquer » les ports d'un ordinateur, le rendant invisible aux yeux extérieurs. La plupart des routeurs disposent, eux aussi, d'un firewall matériel, destiné à masquer les adresses IP d'un réseau interne.

5.4.3.1. Principe de fonctionnement

Le firewall a pour but de contrôler le trafic de données entre les zones de confiance et le monde extérieur. En pratique, il va bloquer les connexions non demandées du monde extérieur, tout en autorisant les ordinateurs d'un réseau local à se connecter à la toile.

La principale différence entre un firewall matériel et un firewall logiciel se situe au niveau du traitement des données. Le firewall matériel va filtrer uniquement les ports, sans distinction, tandis que le firewall logiciel permet de contrôler les accès des différentes applications au web.

Il est possible de convertir un vieil ordinateur en un firewall puissant avec des OS dédiés. Un firewall fonctionne à l'aide de listes ordonnées de la forme « règle, action ».

Chaque fois qu'un paquet de données arrive, le firewall compare ce paquet à chaque règle (dans l'ordre) jusqu'à en trouver une qui corresponde au paquet. Il exécute alors l'action correspondante à la règle.

Les règles peuvent être: adresse de destination du paquet, adresse source, port de destination, port source, date, heure, etc.

Les actions peuvent être : refuser le paquet, ignorer le paquet, accepter le paquet, transmettre le paquet sur un autre réseau, modifier les entêtes du paquet...

5.5. La couche Application

Cette couche n'est en réalité pas gérée par le stack IP, mais directement par le système d'exploitation. Cette couche va gérer les divers services que peut proposer un serveur. Nous verrons en détails certains de ces services, comme le *DNS*, le *DHCP* et les sockets.

Nous verrons aussi comment fonctionne le serveur *SSH* avec son protocole associé, **TLS**.

5.5.1. Protocoles

5.5.1.1. DNS

Un serveur **DNS** (Domain Name System) est un serveur essentiel dans le monde Internet. C'est un annuaire qui va permettre de traduire un nom de domaine (par exemple `http://www.perdu.com`) en une adresse IP (dans ce cas, 208.97.177.124).

En tapant une adresse dans le navigateur, il va tout d'abord envoyer une requête DNS *query* à un serveur connu par l'ordinateur (généralement fourni par votre fournisseur d'accès) en lui demandant l'adresse IP du serveur. Si le serveur DNS connaît la réponse, il envoie une requête DNS *response* à l'ordinateur avec l'adresse demandée. Sinon, il demande à un autre DNS. Dans le cas où personne ne connaît la réponse, cela signifie que le nom de domaine n'existe pas et le serveur DNS envoie une réponse nulle.

5.5.1.2. DHCP

Un serveur **DHCP** (Dynamic Host Configuration Protocol) est un service qui va fournir des adresses IP aux ordinateurs qui se connectent sur le réseau.

Puisqu'une adresse IP doit être unique sur le réseau sur lequel elle est située, le serveur DHCP va gérer les adresses et n'attribuer que les adresses non utilisées à tout nouvel ordinateur qui en fait la demande. Par défaut, votre box internet est aussi un serveur DHCP. Le serveur DHCP va délivrer un *bail*, c.-à-d. une adresse IP valable pendant un certain temps, à l'ordinateur qui en fait la demande. Dans ce bail figure plusieurs infos, dont notamment la durée de vie du bail. L'adresse IP fournie au client et les paramètres du réseau, comme les serveurs DNS.

L'avantage des baux limités dans le temps permet au serveur DHCP de récupérer des adresses IP non utilisées. La durée d'un bail est variable et peut s'étendre entre quelques minutes à quelques semaines. Tout dépend de la configuration du serveur.

À l'expiration du bail, le serveur DHCP va vérifier si les ordinateurs sont toujours connectés. Si c'est le cas, l'ordinateur va recevoir un nouveau bail avec la même adresse IP. Sinon, l'adresse est remise en disponibilité pour un autre périphérique.

Les adresses IP fournies par le service DHCP sont dites *dynamiques*, car les périphériques reçoivent une adresse temporaire. À contrario, les adresses *statiques* sont des adresses « bloquées » qui sont enregistrées directement sur le périphérique.

L'avantage d'une adresse statique est que son adresse ne change justement pas, ce qui est un élément crucial pour les serveurs ! En effet, si un serveur web change d'adresse IP régulièrement, il n'est pas facilement joignable. C'est un peu comme si vous changiez de n° de téléphone tous les jours...

Toutefois, les serveurs DHCP vont tenter de vous donner la même adresse IP de connexion en connexion. On pourrait donc croire que notre adresse IP est statique, mais ce n'est pas le cas. Aussi, les serveurs DHCP peuvent **réserver** une adresse IP pour un périphérique. Par exemple, si on possède une imprimante réseau, on peut demander au serveur DHCP de lui attribuer toujours la même adresse IP.

5.5.1.3. HTTP

Le protocole **HTTP** (HyperText Transfer Protocol) est le protocole le plus utilisé sur Internet depuis 1990. Le but du protocole HTTP est de permettre un transfert de fichiers HTML localisés grâce à une URL entre un navigateur (le client) et un serveur web (le serveur).

La communication entre le navigateur et le serveur se fait en deux temps :

- Le client effectue une */HTTP request/*, un ensemble de lignes comprenant :
 - Une **ligne de requête**, une ligne précisant le type de document demandé, la méthode de récupération et la version du protocole utilisé, le tout séparé par un espace.
 - Les **champs d'en-tête de la requête**, un ensemble de lignes facultatives permettant de donner les informations supplémentaires sur la requête et le client.
 - Le **corps de la requête**, un ensemble de lignes optionnelles séparées par une ligne vide et permettant un envoi de données par une commande *POST* lors de l'envoi des données au serveur par un formulaire.
- Le client reçoit une *HTTP response*, un ensemble de lignes envoyées au navigateur par le serveur. Elle comprend :
 - Une **ligne de statut**, précisant la version du protocole utilisé et l'état du traitement de la requête à l'aide d'un code et d'un texte d'explication.
 - Les **champs d'entête de la réponse**, ensemble de lignes facultatives permettant de donner des informations supplémentaires sur la réponse et/ou le serveur. Chacune de ces lignes est composée d'un nom qualifiant le type d'en-tête, suivi de deux points et de la valeur de l'en-tête.
 - Le **corps de la réponse**, contenant le document demandé.

5.5.1.4. SSH

Internet permet de réaliser un grand nombre d'opérations à distance, comme l'administration de serveurs ou le transfert de fichiers. Le plus vieux protocole de gestion à distance, *telnet* possède l'inconvénient majeur de faire circuler en clair sur le réseau les informations échangées, notamment l'identifiant et le mot de passe pour l'accès de la machine distante. Ainsi, un pirate situé sur un réseau entre l'utilisateur et la machine distante a la possibilité de capturer le trafic et donc de récupérer ces infos.

Le protocole *SSH* (Secure SHell) répond à cette problématique en permettant à des utilisateurs d'accéder à une machine à travers une communication chiffrée, nommée **tunnel**. Il s'agit d'un protocole permettant à un client (un utilisateur) d'ouvrir une session interactive sur un serveur afin d'envoyer des commandes ou des fichiers de manière sécurisée. Les données circulant entre le client et le serveur sont chiffrées, ce qui garantit leur confidentialité. Le client et le serveur s'authentifient mutuellement pour assurer que les deux machines qui communiquent sont bien celles que chacune des parties croit être. Il n'est donc plus possible pour un pirate d'usurper l'identité du client ou du serveur.

5.5.1.5. Les sockets

La notion de **sockets** a été introduite dans les distributions *BSD*. Il s'agit d'un modèle permettant la communication inter-processus afin de permettre à divers processus de communiquer aussi bien sur une même machine en local qu'à travers un réseau TCP/IP.

La communication par sockets permet d'utiliser le protocole TCP (en mode connecté) ou en UDP (en mode non connecté). En mode connecté, 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. En mode non connecté, chaque envoi de données nécessite l'adresse de destination et aucun accusé de réception n'est donné.

Pour comprendre le fonctionnement des sockets, nous allons construire une application de chat en Python, en créant un serveur et un client.

6. Notion d'adressage réseau en IPv4

La notion d'adresse IP est fondamentale pour le routage des données dans les réseaux informatiques. Une adresse IP unique identifie chaque appareil sur un réseau, permettant au système de distinguer entre différentes destinations et sources lors du transfert des paquets. Le processus d'adressage implique l'attribution de ces adresses aux périphériques connectés via les routeurs ou switches qui les relaient jusqu'à leur destination finale.

6.1. Adressage « statefull »

Prenons l'exemple suivant: on souhaite diviser notre réseau $212.168.218.0 /24$ en **7** sous-réseaux de **30** machines.

Il existe deux techniques pour créer un plan d'adressage IP adapté à ce besoin. La première est de convertir **tout** en binaire. Nous n'allons pas la détailler ici, car elle est longue et peut être susceptible de générer des erreurs.

La seconde utilise une technique « maison » qui est simple et fonctionnelle: **le nombre magique**.

1. La première étape est de tracer notre tableau d'adresses (un tableur peut être utile). Dans notre exemple, notre adresse IP « globale » est $212.168.218.x$, x étant variable (de 0 à 255). Avec le masque « global » ($/24$), nous savons que nous pouvons avoir au maximum **254** machines dans notre réseau. $32 - 24 = 8$ et $2^8 - 2 = 254$. Nous devons retirer **2 adresses** (l'adresse **NET** et l'adresse **BROAD**).

Pour chaque *sous-réseau*, nous aurons également 2 adresses **bloquées**, d'où le « -2 » qui apparaît dans les formules.

2. Ensuite, nous allons calculer le nombre **magique**. Le nombre magique est un nombre **puissance de 2** et correspond à la formule suivante: $(MAG = 2^x) - 2 \geq \text{nb mach}$

Dans ce cas, $2^x - 2 \geq 30$, donc $MAG = 32(2^5)$

3. Les **adresses réseaux (NET)** se calculent avec le nombre magique, à partir de la *première adresse IP*. Le truc est simple: on part de **0** et on multiplie le n° du réseau avec le nombre MAG.
4. L'adresse **FROM** est la première adresse utilisable de chaque sous-réseau.
5. L'adresse **TO** est $MAG - 2$ plus loin que l'adresse **NET**.
6. Avec cette construction, le broadcast (**BROAD**) est l'adresse juste *après* la dernière adresse disponible et *avant* le subnet suivant.
7. Enfin, le masque peut se calculer aussi avec le masque. Sachant qu'un masque IPv4 fait au maximum **32 bits**, on va soustraire *l'exposant magique*.

Dans notre exemple, on sait que l'exposant magique $MAG = 2^x = 32$ donc $x = 5$.

| N° | MAG | NET | FROM | TO | BROAD | MASK |
|--------------------------------------|-----|-----------------------|-----------------------|------------------------|-----------------------|------|
| Global IP Address 212.168.218.0 / 24 | | | | | | |
| 0 | 32 | 212.168.218. 0 | 212.168.218. 1 | 212.168.218. 30 | 212.68.218. 31 | /27 |
| 1 | 32 | .32 | .33 | .62 | .63 | /27 |
| 2 | 32 | .64 | .65 | .94 | .95 | /27 |
| 3 | 32 | .96 | .97 | .126 | .127 | /27 |
| 4 | 32 | .128 | .129 | .158 | .159 | /27 |
| 5 | 32 | .160 | .161 | .190 | .191 | /27 |
| 6 | 32 | .192 | .193 | .222 | .223 | /27 |
| 7 | 32 | .224 | .225 | .254 | .255 | /27 |

Note

Dans cet exemple, puisque nous avons découpé notre réseau en sous-réseaux équivalents, il existe un 8^e sous-réseau qui n'est pas utilisé. Ces adresses sont *perdues* dans ce type d'adressage. C'est pourquoi nous allons utiliser l'adressage **VLSM** qui permet d'utiliser un *masque variable*.

6.1.1. Autre exemple

Prenons un autre exemple: nous souhaitons diviser le subnet 172.16.0.0 /16 en **7** réseaux de **840** machines.

Reprenons notre tableau et notre marche à suivre.

1. Le masque « global » étant /16, nous savons que nous pouvons utiliser au maximum $2^{16} - 2 = 65.534$ machines. On doit placer env. 6000 machines, donc on a assez de place.
2. $(MAG = 2^x) - 2 \geq 840$, donc $MAG = 1024$. Ici, nous avons un problème, car cela dépasse 256 (un octet). Nous allons devoir travailler sur 2 octets pour réaliser notre adressage.
3. Pour les adresses **NET**, ce nombre magique va poser problème. Pour le sub 1, on doit faire $1 * 1024 = 1024$. Mais, comme chaque bloc IP ne peut dépasser 255, nous allons diviser le nombre obtenu par 256. Le résultat obtenu sera reporté sur l'octet **précédent**.
4. Remplissons maintenant les **FROM**.
5. Pour les **TO**, de nouveau une division va nous sauver. Il faut se décaler de 1022 par rapport au subnet. Une petite division euclidienne: $\frac{1022}{256} = 3$ et le reste $R = 1022 \% 256 = 254$.
6. Le calcul du masque n'a pas changé: $32 - 10 = 22$.

| N° | MAG | NET | FROM | TO | BROAD | MASK |
|-----------------------------------|-----|--------------------|--------------------|----------------------|----------------------|------|
| Global IP Address 172.16.0.0 / 16 | | | | | | |
| 0 | | 172.16. 0.0 | 172.16. 0.1 | 172.16. 3.254 | 172.16. 3.255 | /22 |
| 1 | | .4.0 | .4.1 | .7.254 | .7.255 | /22 |
| 2 | | .8.0 | .8.1 | .11.254 | .11.255 | /22 |
| 3 | | .12.0 | .12.1 | .15.254 | .15.255 | /22 |
| 4 | | .16.0 | .16.1 | .19.254 | .19.255 | /22 |
| 5 | | .20.0 | .20.1 | .23.254 | .23.255 | /22 |
| 6 | | .24.0 | .24.1 | .27.254 | .27.255 | /22 |

6.2. Adressage VLSM

Dans le premier exemple, nous avons utilisé l'adressage **classique**, avec un masque fixe de /27 pour tous les sous-réseaux. Mais dans le deuxième exemple, on a vu que le masque était toujours /22 (car $32 - 10 = 22$) mais que les adresses changeaient.

C'est pourquoi il existe une autre technique d'adressage : l'**VLSM**. L'idée est simple : si on a besoin de gros sous-réseaux, on va utiliser un masque **court**, et pour des petits, on utilisera un masque **long**. Le subnetting VLSM permet donc d'avoir une allocation plus efficace.

Dans le premier exemple, nous n'avons qu'un total de $7 \times 30 + 2 = 214$ adresses IP utilisables. Mais avec les sous-réseaux /27, on a alloué $32 - 2 = 30$ machines par sous-réseau, donc un gaspillage énorme (car 32 est la puissance de deux supérieure à 30).

Avec l'adressage VLSM, on peut avoir des tailles de réseaux différentes. On va utiliser le masque /24 pour les petits et /27 seulement si nécessaire.

Heureusement, la technique de l'adressage par nombre magique est toujours valable !

Prenons un exemple : Une entreprise possède le réseau 192.168.10.0/24 et souhaite le subdiviser en sous-réseaux en utilisant la technique VLSM afin de répondre aux besoins suivants :

- Service A : 60 hôtes
- Service B : 30 hôtes
- Service C : 12 hôtes
- Service D : 6 hôtes
- R1 et R2: 2 hôtes (connexion point à point)

La technique est la même, mais avec une subtilité. Au lieu d'avoir un masque commun, on va calculer un masque de sous-réseau **par subnet**, puis les classer des masques les plus petits (plus grands réseaux) au plus grands (plus petits réseaux).

- MAG A : $2^x - 2 \geq 60$ donc $MAG_A = 64$, masque : $32 - 6 = 26$.
- MAG B : $2^x - 2 \geq 30$ donc $MAG_B = 32$, masque : $32 - 5 = 27$.
- MAG C : $2^x - 2 \geq 12$ donc $MAG_C = 16$, masque : $32 - 4 = 28$.
- MAG D : $2^x - 2 \geq 6$ donc $MAG_D = 8$, masque : $32 - 3 = 29$.
- MAG E : $2^x - 2 \geq 2$ donc $MAG_E = 4$, masque : $32 - 2 = 30$.

1. Pour le calcul des subnets, il faut désormais ajouter le **nombre magique précédent** au subnet.

2. Le reste se fait comme précédemment.

| N° | MAG | NET | FROM | TO | BROAD | MASK |
|-------------------------------------|-----|----------------|---------|---------|---------|------|
| Global IP Address 192.168.10.0 / 24 | | | | | | |
| 0 | | 192.168.10.0 | .10.1 | .10.62 | .10.63 | /26 |
| 1 | | 192.168.10.64 | .10.65 | .10.94 | .10.95 | /27 |
| 2 | | 192.168.10.96 | .10.97 | .10.110 | .10.111 | /28 |
| 3 | | 192.168.10.112 | .10.113 | .10.118 | .10.119 | /29 |
| 4 | | 192.168.10.120 | .10.121 | .10.122 | .10.123 | /30 |

Notes

Avec cette méthode, il nous reste encore 1/2 range disponible!

Dans la pratique, on essaie de mettre les subnets les plus petits en fin de range.