

Principes et architecture des réseaux

NIKIEMA Ousmane
Ousmane.nikiema@outlook.com

Avril 2020

Table des matières

Première partie	4
I.1. Les réseaux:présentation générale	6
I.1.1. C'est quoi un réseau? Ça sert à quoi?	6
I.1.2. Les 2 réseaux mondiaux incontournables	7
I.1.2.1. Le réseau Internet	8
I.1.2.2. Le réseau Télécom	8
I.2. Les topologies.....	9
I.2.1. Avant tout.....	9
I.2.1.1. LAN: le réseau local	9
I.2.1.2. WAN: le réseau étendu	10
I.2.2. C'est quoi une topologie?	10
I.2.2.1. Topologie physique	11
I.2.2.2. Topologie logique.....	11
I.2.3. Réseau en bus	11
I.2.4. Topologie de type étoile.....	12
I.2.5. Réseau en anneau: l'ring, mais pas de boxe.....	13
I.2.6. Topologie maillée	14
I.2.7. Topologie hybride.....	15
Deuxième partie.....	17
II.1. Introduction aux protocoles	19
II.1.1. Vous avez dit protocole?	19
II.1.1.1. Le protocole : un genre de langue.....	19
II.1.2. L'utilité d'un protocole par l'exemple	20
II.1.3. Les exigences d'un protocole	22
II.2. Ils en tiennent une couche: OSI et TCP/IP	25
II.2.1. Le modèle OSI en douceur.....	25
II.2.1.1. Qu'est-ce que le modèle OSI?	25
II.2.2. Le modèle OSI par l'exemple : le facteur	27
II.2.3. Survol des couches du modèle OSI.....	28
II.2.3.1. Comment ça fonctionne?	28
II.2.3.2. Résumé	31
II.2.3.3. Processus de transmission/réception	31

II.2.4. TCP/IP vs OSI : le verdict?	32
II.2.4.1. Il y a une génération.....	32
II.2.4.2. Comparaison dans la structure	33
II.2.4.3. Point vocabulaire : les unités de données.....	36
II.2.4.4. Faites attention à l'abstraction des noms de couches.....	37
II.2.4.5. Critiques du modèle OSI	37
II.2.4.6. Critiques du modèle TCP/IP	38
II.2.4.7. Et maintenant : le verdict des juges.....	38
II.2.5. Principe d'encapsulation	39
Troisième partie	44
III.1. Des adresses en folie!	46
III.1.1. IP vs MAC	46
III.1.1.1. Adresse IP : l'adresse relative au réseau.....	46
III.1.1.2. Adresses MAC : l'adresse relative à la carte réseau.....	47
III.1.1.3. En résumé.....	48
III.1.2. Masque de sous-réseau et passerelle	48
III.1.2.1. Les sous-réseaux et leurs masques.....	49
III.1.2.2. La passerelle	50
III.1.3. Le client et le serveur	51
III.2. Les masques de sous-réseaux: à la découverte du subnetting	52
III.2.1. En bref	52
III.2.2. L'importance des masques	52
III.2.2.1. Relation entre network ID et masques.....	52
III.2.2.2. Des règles fondamentales à connaître absolument.....	53
III.2.3. Introduction au subnetting	53
III.2.3.1. Délégation de l'administration	53
III.2.3.2. La réduction du trafic.....	54
III.2.3.3. La facilité du diagnostic.....	54
III.2.3.4. L'économie d'adresses.....	54
III.2.4. Analyse des contraintes et plan d'adressage	55
III.2.4.1. Analyse des contraintes	55
III.2.4.2. Le prix	55
III.2.4.3. L'évolution du réseau	55
III.2.4.4. Le nombre d'adresses IP	56
III.2.4.5. L'organisation.....	56
III.3. Le subnetting en pratique	59
III.3.1. Comment?	59

III.3.1.1. Le comment du pourquoi.....	59
III.3.2. À partir du nombre de sous-réseaux désirés.....	60
III.3.2.1. Exemple de subnetting	61
III.3.3. À partir du nombre d'adresses d'hôtes désirées	63
III.3.3.1. Explications sur l'adresse de <i>broadcast</i> et l'identité du réseau.....	63
III.3.3.2. Un autre exemple de subnetting	64
III.3.3.3. Exemple de subnetting avec moins de 254 hôtes par sous-réseau	65
III.3.4. La notation du masque	65
III.3.4.1. La notation «classique»	66
III.3.4.2. La notation avec un slash (/)	66
III.4. L'adressage par classes (obsolète)	66
III.4.1. C'est quoi une classe?	67
III.4.2. Classe A.....	69
III.4.2.1. Présentation.....	69
III.4.2.2. Structure d'une adresse IP de la classe A	71
III.4.3. Classes B et C	72
III.4.3.1. Classe B 72	
III.4.3.2. Classe C 73	
III.4.4. Classes D et E	74
III.4.4.1. Quelques informations.....	75
III.4.5. Notion de classe privée	75
III.5. L'adressage CIDR	76
III.5.1. Révision de l'adressage par classes	76
III.5.2. CIDR et le supernetting.....	77
III.5.2.1. CIDR : le comment	78
III.5.2.2. Comment résumer une route	79
III.5.2.3. Quelques exercices pour la route.....	80

Première partie

Le concept et les bases

I. Le concept et les bases

Dans cette partie nous allons apprendre beaucoup de théorie. Nous allons dans un premier temps nous attarder sur cette question : qu'est-ce qu'un réseau ? Nous étudierons et comprendrons ce que c'est qu'un réseau. Nous allons voir que les réseaux n'existent pas qu'en informatique. Nous verrons de quoi est composé un réseau, le matériel nécessaire, et la forme qu'un réseau peut prendre. On commence donc doucement, mais sûrement!

I.1. Les réseaux : présentation générale

Dans ce chapitre, nous allons aborder la notion de réseau, en commençant par une question toute simple : c'est quoi un réseau ?

I.1.1. C'est quoi un réseau? Ça sert à quoi?

Avant toute chose, il est indispensable de répondre à la question suivante : qu'est-ce qu'un réseau? On pourrait définir le mot « réseau » en une phrase : un réseau est un groupe d'entités en communication.

?

C'est quoi une entité?

Une entité peut désigner une « chose » parmi d'autres. Par exemple, une personne dans un groupe de personnes est une entité de ce groupe. Pour rester dans cet exemple, on parle de réseau quand deux ou plusieurs personnes parlent ensemble.

?

C'est tout, un réseau c'est juste quand on parle ensemble?

Oui, mais n'oubliez pas que « parlent ensemble » c'est aussi « s'échangent des informations » !;) Donc, en gros, un réseau consiste en l'échange d'informations, et il existe (dans la vie courante) plusieurs moyens d'échanges d'informations, sans faire intervenir la technologie (Internet, téléphone, etc.). Si on veut vous donner un livre, on prend le livre, et on vous tend la main, puis vous prenez le livre. 🍎 Vous l'aurez compris, il existe plusieurs manières de partager des données entre les humains, sans les technologies.

Ce qui est intéressant, c'est que je peux envoyer (transmettre) un livre à André, en passant par Pierre.

Eh Pierre, si tu vois André, passe-lui le livre, et qu'il te le remette quand il aura fini de le lire.
Ce qui se passe dans ce cas est :

- Je donne le livre à Pierre
- Pierre trouve André et le lui donne
- André a fini, il rend le livre à Pierre
- Pierre vient me rendre le livre

Nous allons supposer dans ce cas présent qu'André et moi ne nous voyons pas, donc, Pierre est dans ce cas un intermédiaire. Justement, le principe d'intermédiaire est un des fondements des réseaux informatiques. Vous allez rapidement vous en rendre compte.



Pour communiquer, les 2 entités doivent parler la même langue. Ou alors, l'intermédiaire doit parler la langue de chacun de ses interlocuteurs. En réseau informatique, c'est pareil, sauf qu'on ne parle pas de langue mais de protocole.

Si vous avez compris ce qu'est un réseau avec des humains, vous avez tout compris. Un réseau informatique est exactement pareil, sauf qu'il faut remplacer les humains par des machines. Hé oui.



Mais... mais... et les câbles, les adresses je-ne-sais-pas-quoi... ? On en fait quoi?



On ne va pas se compliquer l'existence tout de suite hein. :-° Pour l'instant, on reste dans l'approche globale du réseau; les liaisons et la configuration, on verra plus tard. Vous ne voulez quand même pas que l'on monte un réseau d'entreprise dès le premier chapitre, si? o_O



Concrètement, un réseau informatique, ça sert à quoi?

Eh bien, sans réseau informatique, vous ne seriez pas en train de lire ce tuto, déjà. 🍊

De manière globale, un réseau informatique permet l'échange d'informations à distance. On peut trouver plein d'applications à un réseau : discuter avec une personne, s'échanger des documents, jouer en ligne...



Retenez bien le terme d'application de réseau ! Une application est l'utilisation (voire l'exploitation) d'une ressource pour en faire quelque chose de concret. Ici, on exploite un réseau informatique pour discuter par exemple. En mécanique, on peut exploiter du matériel pour faire une voiture : c'est une application de la mécanique (le rédacteur ayant écrit ça n'y connaît absolument rien en mécanique, si quelqu'un veut refaire l'exemple qu'il n'hésite pas).



I.1.2. Les 2 réseaux mondiaux incontournables

Dans un tutoriel sur les réseaux informatiques, on ne pouvait pas manquer de parler d'Internet, bien évidemment. Mais à l'origine, pour accéder à Internet, on passe par une ligne téléphonique. C'est pourquoi, dans la catégorie réseau informatique, on distingue deux types de réseaux :

- Le réseau Internet;
- Le réseau Télécom.

I.1.2.1. Le réseau Internet

Internet désigne le réseau public et mondial permettant l'échange de données. Quand un ordinateur est connecté à ce réseau, on dit qu'il a accès à Internet. On confond parfois ce réseau avec le World Wide Web, alors qu'il y a une grande différence entre ces deux notions.

I.1.2.1.1. Le World Wide Web

Le World Wide Web, ou Web pour faire plus court, est l'ensemble des sites Web (appelés par abus de langage «sites Internet») présents sur le réseau Internet. La toile, comme on dit parfois en français, c'est donc l'ensemble de tous les sites Web que nous pouvons visiter grâce à notre navigateur Web (Firefox, Opera, ...).

I.1.2.1.2. Internet

Internet, par contre, c'est l'ensemble des nœuds (connexions, câbles, etc.) entre les machines qui nous donnent accès au web. Internet est donc l'ensemble des réseaux qui nous permettent de partager des données, notamment sur la toile. Donc, quand une personne vous demande si vous avez Internet, elle veut savoir si votre ordinateur a accès à Internet. Par ailleurs c'est encore un abus de langage que de dire que l'on a Internet : ce réseau gigantesque n'appartenant à personne, on ne peut qu'avoir accès à Internet.

I.1.2.2. Le réseau Télécom

Étymologiquement, le mot télécommunication (abrégé télécom) signifie communication à distance. Le réseau Télécom a donc pour but d'assurer la communication à distance, par la transmission électrique de la voix. Ce réseau est similaire au réseau Internet en plusieurs points, comme l'identité unique, les «sous-réseaux» formés par les délimitations territoriales... Nous ne pouvons pas expliquer ça dès le début, mais soyez patient·e : vous comprendrez l'analogie dans peu de temps. 🍊

On espère que ce chapitre ne vous a pas ennuyé, car il est primordial si l'on veut avancer dans le cours. Nous avons abordé le fonctionnement de la transmission des données, en nous inspirant de la vie courante. L'exemple n'était certes pas original, mais il est tout de même très pratique pour comprendre les adresses, les protocoles, etc. Vous n'allez pas tarder à vous en rendre compte!

I.2. Les topologies

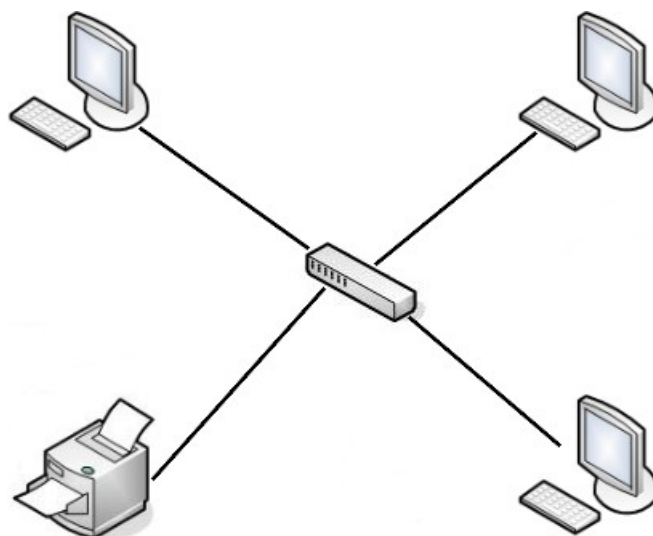
Dans ce chapitre nous allons étudier les topologies. Il s'agit des différentes formes que peuvent prendre des réseaux.

I.2.1. Avant tout...

Avant tout, il faut que vous connaissiez quelques types de réseaux, cela aidera à comprendre pourquoi certaines topologies existent.

I.2.1.1. LAN : le réseau local

Un LAN, *Local Area Network* (en français réseau local) est un réseau limité à un espace géographique comme un bâtiment. Par exemple, l'ensemble des ordinateurs dans une école forme un LAN.



Les réseaux de zéro - zestedesavoir.com

FIGURE I.3.1. – Représentation schématique d'un LAN simple



Un WLAN, *Wireless Local Area Network*, ou *Wireless LAN*, est un LAN mais qui utilise la transmission sans fil (Wi-Fi, ...). Le mot *wireless* signifie «sans fil» (*wire* = câble, *less*

= sans). Par exemple, un *hotspot* Wi-Fi, c'est-à-dire un point d'accès Wi-Fi public comme on en trouve dans des lieux publics tels qu'un hôtel, est un réseau local sans fil (WLAN).

I.2.1.2. WAN : le réseau étendu

WAN signifie *Wide Area Network*, en français, on peut le traduire par « réseau étendu ». Il s'agit d'une interconnexion de réseaux sur de longues distances. Quand une grande entreprise interconnecte ses différents sites (usines de production, plateformes logistiques, ...), on parle de WAN. Les réseaux étendus se caractérisent par l'utilisation de technologies différentes des LAN. Par exemple, alors qu'on trouve souvent des câbles Ethernet ou du Wi-Fi sur des réseaux locaux, les connexions WAN se font plutôt en fibre optique voire en 4G.

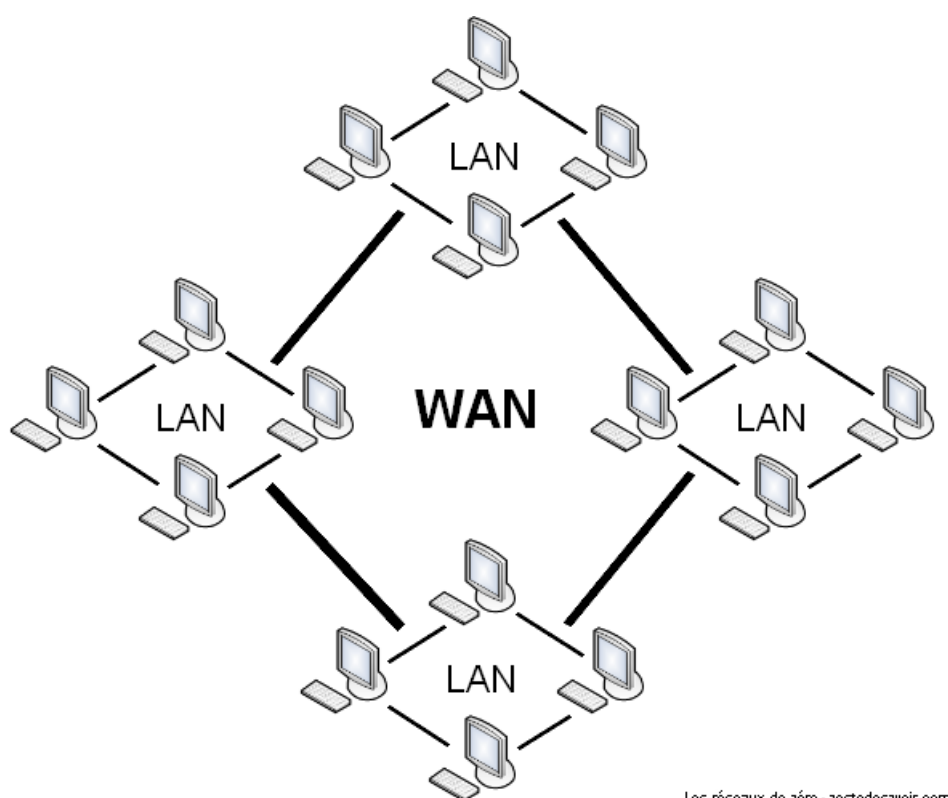


FIGURE I.3.2. – Représentation schématique d'un WAN

I.2.2. C'est quoi une topologie?

Bonne question, qu'est-ce qu'une topologie? Tout d'abord, il faut savoir qu'il existe deux types de topologies : physique et logique.

I.2.2.1. Topologie physique

Une topologie physique est en fait la structure physique de votre réseau. C'est donc la forme, l'apparence du réseau. 🍊

Il existe plusieurs topologies physiques : le bus, l'étoile (la plus utilisée), le mesh (topologie maillée), l'anneau, hybride, etc. Cependant nous n'allons parler que des plus utilisées.

I.2.2.2. Topologie logique

Une topologie logique est la structure logique d'une topologie physique, c'est-à-dire que la topologie logique définit *comment* se passe la communication dans la topologie physique. 🍊

!

Attention avec ces deux notions !

L'une (topologie physique) définit la **structure physique** (l'apparence physique, la forme) de votre réseau, l'autre (topologie logique) définit **comment la communication se passe** dans cette forme physique. ;) Retenez bien ces 2 notions, et ne les confondez pas, tant qu'à faire.



I.2.3. Réseau en bus

Comme son nom l'indique, la topologie bus a les caractéristiques d'un bus (pensez, une ligne droite). Dans cette topologie, tous les ordinateurs sont connectés entre eux par le biais d'un seul câble réseau débuté et terminé par des **terminateurs**.

Les terminateurs ont pour but de maintenir les **frames** (signaux électriques de données) dans le câble et d'empêcher les «rebonds» des données le long du fil. 🍊

Franchement, ce n'est pas pratique du tout, et ce pour 2 raisons majeures. La première est que, parce que toutes les machines utilisent le même câble, s'il vient à ne plus fonctionner, alors le réseau n'existe plus. Il n'y a plus de communication possible étant donné que tous les hôtes partagent un câble commun. La seconde est que, puisque que le câble est commun, la vitesse de transmission est très faible. 🍊 Il y a d'autres raisons qui font que cette topologie est très peu utilisée.

Dans cette topologie, étant donné que le câble de transmission est commun, il ne faut pas que 2 machines communiquent simultanément, sinon... Bam, ça crée des collisions ! 🍊 Pour éviter ce problème, on utilise une méthode d'accès appelée CSMA/CD. Avec cette méthode, une machine qui veut communiquer écoute le réseau pour déterminer si une autre machine est en train d'émettre. Si c'est le cas, elle attend que l'émission soit terminée pour commencer sa communication. Sinon, elle peut communiquer tout de suite.

C'est un peu complexe, heureusement que d'autres topologies plus simples et plus pratiques existent !

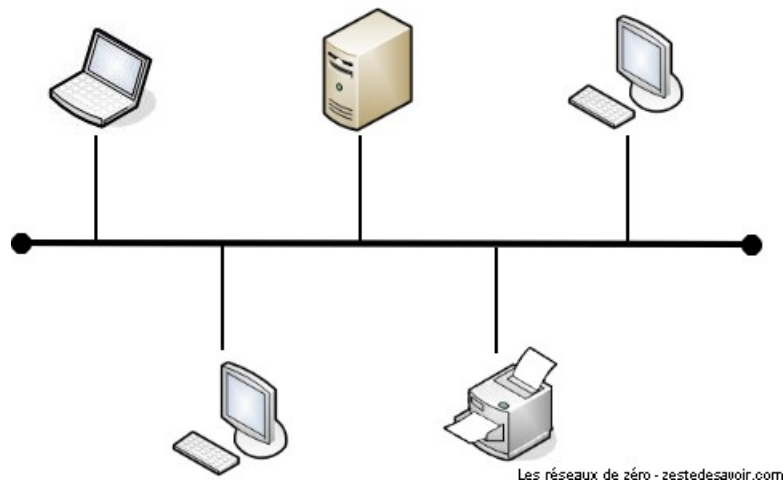


FIGURE I.3.3. – Représentation schématique d'un réseau en bus

I.2.4. Topologie de type étoile

Dans un réseau en étoile, la forme physique du réseau ressemble à une étoile. Une image est plus parlante :

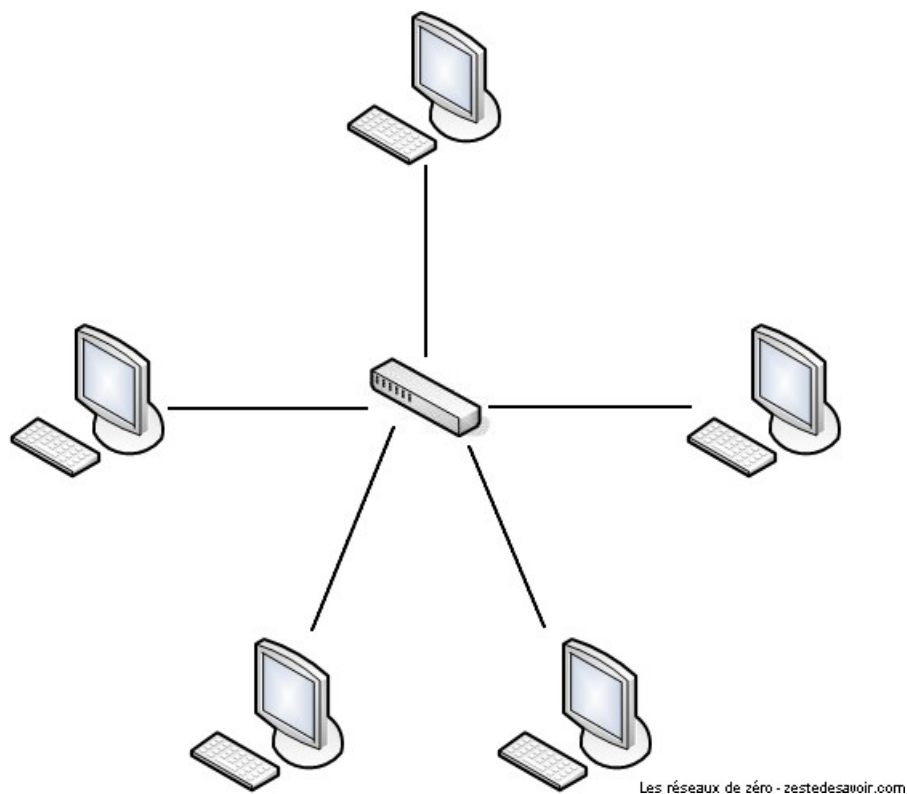


FIGURE I.3.4. – La forme physique du réseau ressemble à une étoile

N'importe quel appareil (routeur, commutateur, concentrateur, ...) peut être au centre d'un réseau en étoile. L'important, c'est que pour parler à une autre entité on passe par le matériel

central (qui peut être le *hub*, le *switch*, etc.). En pratique, dans un réseau d'entreprise en étoile, au centre on trouve un *switch*. 🍊

Le principal défaut de cette topologie, c'est que si l'élément central ne fonctionne plus, plus rien ne fonctionne : toute communication est impossible. Cependant, il n'y a pas de risque de collision de données.



Si vous reliez des ordinateurs à un *hub*, la topologie **physique** sera l'étoile. Mais la topologie **logique** sera... le bus ! En effet, sur un *hub*, seule une machine peut émettre à la fois. Les autres doivent écouter le réseau pour savoir si elles peuvent émettre !

I.2.5. Réseau en anneau : le ring, mais pas de boxe

Oui bon, le jeu de mot est pourri... Enfin, vous devez commencer à avoir l'habitude ! 🍊

On attaque un morceau assez compliqué, du moins plus complexe que ce qu'on a vu jusqu'à présent. Nous allons donc essayer de faire simple (très contradictoire 🍊).

Comme vous pouvez vous en douter, un réseau en anneau a la forme d'un... anneau, oui, il n'y a pas de piège ! Cependant, la topologie physique d'un réseau en anneau est... le bus.



Mais alors un réseau en anneau c'est comme un réseau en bus avec les machines disposées en cercle ?

Si on veut, mais il a une particularité : la topologie logique est le *token ring*.



Anneau à jeton ? On met un jeton dans la machine pour avoir un anneau ? 🍊

Pas du tout ! 🍊 Rappelez-vous, la topologie de type bus possédait un problème de collision de données : 2 machines ne doivent pas échanger des données en même temps, sinon elles s'entrechoquent. Ce principe est repris dans le réseau en anneau. Sauf que là, le système de *token ring* utilise la CSMA-CA, une méthode anti-collision différente.

Le principe est assez simple : une machine connectée au réseau possède un jeton virtuel. Ce jeton, c'est une **autorisation de communiquer**. Une fois que la machine a transmis ce qu'elle voulait, elle passe le jeton à la machine suivante, et ainsi de suite. Si le détenteur du jeton n'a rien à dire, il le passe au suivant.



Vous allez nous dire qu'on radote, mais on le répète quand même : la topologie physique, ici le bus, définit la forme physique du réseau (bon ici le bus est un peu courbé... 🍊). La topologie logique, ici le *token ring*, définit la manière de communiquer dans un réseau. Si vous confondez, vous allez vous retrouver à vouloir brancher un jeton de casino sur une



machine pour qu'elle puisse communiquer... 🍊

Voici une animation décrivant de manière simplifiée le fonctionnement logique d'un réseau en anneau. Le jeton rouge se transmet de machine en machine.

Voir l'image token ring.gif

FIGURE I.3.5. – Réseau en anneau. Des ordinateurs attendent le jeton (token) pour transmettre des données.

I.2.6. Topologie maillée

La topologie maillée est LA topologie que l'on vous souhaite de ne jamais utiliser! 🍊

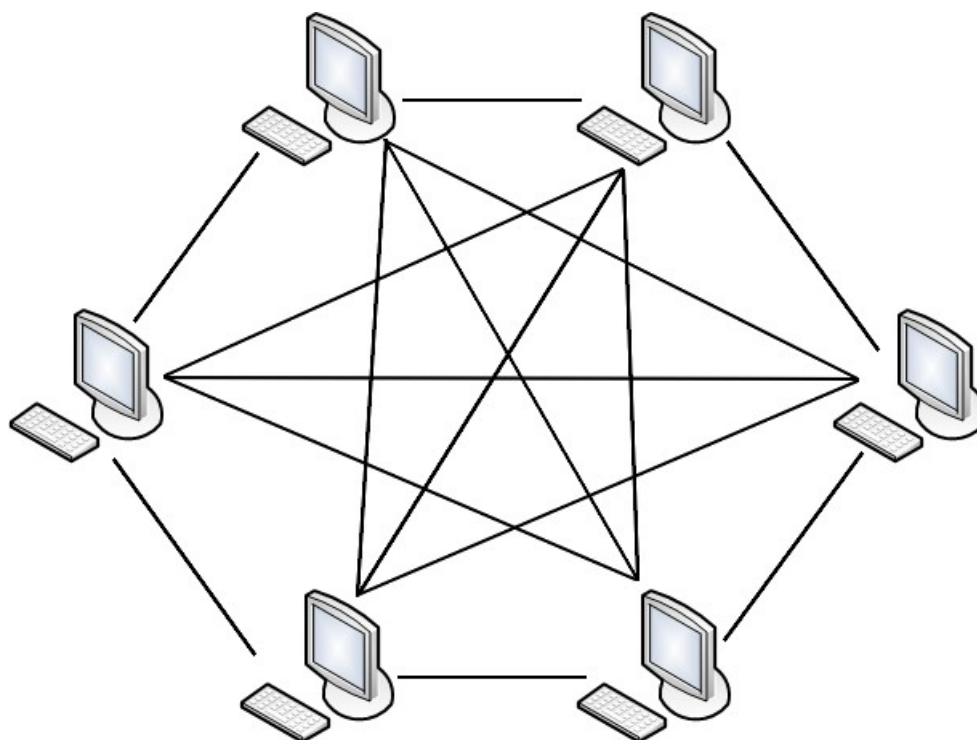


Pourquoi?

Eh bien, c'est qu'il y a vraiment, vraiment, vraiment, vraiment... trop de câbles. 🍊 Le principe de la topologie maillée est de relier tous les ordinateurs entre eux (full mesh, maillage complet) ou du moins, un grand nombre. Comme ça, aucun risque de panne générale si une machine tombe en rade, mais si vous vous prenez les pieds dans des câbles, étant donné qu'il y en a partout, c'est la cata, vous faites tout tomber! 🍊

Si on veut relier toutes les machines entre elles, la formule pour connaître le nombre de câbles est $n(n-1)/2$, avec n le nombre d'ordinateurs. Donc rien qu'avec 8 ordinateurs par exemple, ça nous donnera $8(8-1)/2$, soit jusqu'à **28 câbles**! 🍊 En pratique, on n'a quasiment jamais recours à des réseaux *full mesh*, mais seulement partiellement maillés.

Cette topologie reste peu utilisée vu la difficulté à mettre en place une telle infrastructure. Histoire de vous faire halluciner, imaginez une école, où il y a 500 ordinateurs, si on voulait les relier tous entre eux. Ça ferait... $500*(500-1)/2 = \dots$ Faites le calcul vous-même si vous ne nous croyez pas, mais ça fait bien **124.750 câbles**! Il ne vaut mieux même pas penser au prix que peut coûter une centaine de milliers de câbles. En plus, chaque câble doit être relié à 2 cartes réseau, ça ferait 499 cartes réseau par machine, soit 249.500 cartes réseau en tout... Donc oui, ce n'est pas facile à mettre en place, et c'est utilisé sur de petits réseaux dans des cas bien précis. 🍊



Les réseaux de zéro - zestedesavoir.com

FIGURE I.3.6. – Représentation schématisée d'un réseau complètement maillé

I.2.7. Topologie hybride

Une topologie hybride, c'est très simple (enfin, dans le principe) : c'est juste le regroupement de plusieurs topologies différentes. Par exemple, Internet est une parfaite illustration d'un réseau hybride car il joint des réseaux en anneau avec des réseaux en bus, avec des réseaux en étoile, ... Rien de spécial au final. 🍌

i

Internet peut aussi être vu comme un réseau maillé, dans son sens logique. Rappelez-vous, dans un réseau maillé, la multiplication du nombre de câbles permet plusieurs chemins de communication (dans le réseau Internet, toutes les machines ne sont pas toutes reliées entre elles, c'est un mélange de regroupements de nœuds et autres joyeusetés). Comme il y a tout plein de câbles, il y a donc plusieurs chemins possibles pour parler à son destinataire. 🍌 On peut donc décrire Internet comme un réseau maillé (dans le sens logique), car on peut transmettre des données par plusieurs chemins.

Il faut avouer que ce chapitre n'était pas vraiment difficile. Ce qu'il faut comprendre et maîtriser, c'est la différence entre une topologie physique et une topologie logique. Dans le monde professionnel, on utilise généralement des topologies (physiques et logiques) de type étoile.

Maintenant qu'on a fait un rapide tour du matériel, il faudrait peut-être établir des communications !
Pour cela, direction la partie 2, où on va se pencher sur les **protocoles** et sur le **modèle OSI**! 🍊

Maintenant que vous connaissez la théorie nécessaire, nous allons pouvoir passer à la seconde partie.

Deuxième partie

Un modèle qui en tient une couche

II. Un modèle qui en tient une couche

Nous allons nous intéresser aux fondements de la communication. On va voir ce que sont les protocoles, quels sont ceux qu'on utilise couramment, puis on va commencer à rentrer dans des choses un peu moins évidentes mais absolument passionnantes : le modèle en couches. 🍊

II.1. Introduction aux protocoles

Pour que des machines puissent communiquer entre elles, elles doivent respecter certains **protocoles**. Mais qu'est-ce qu'un protocole ?

II.1.1. Vous avez dit protocole?

Avant d'étudier comment communiquent les hôtes dans un vaste réseau tel qu'Internet, il nous faut comprendre ce qu'est un protocole pour commencer.

■ A protocol is a set of rules that define how communication occurs in a network.

C'est la définition la plus basique d'un protocole que vous retrouverez certainement dans plusieurs cours anglais de réseaux. En français, on dit qu'un protocole est **un ensemble de règles qui définissent comment se produit une communication dans un réseau**. Pour mieux appréhender cela, nous allons considérer une analogie.

II.1.1.1. Le protocole : un genre de langue

Communiquer est l'une des activités les plus courantes. Les personnes qui communiquent ne peuvent se comprendre que dans deux cas :

- Si elles parlent la même langue.
- Si elles ont un intermédiaire qui parle leurs deux langues respectives pour faire office d'interprète.

Mais une langue que les humains parlent, qu'est-ce que c'est, au final?

D'après Wikipédia, une langue est un système constitué de signes linguistiques, vocaux, graphiques, gestuels, tenu en cohésion par des règles précises qui, lorsque respectées, permettent la communication.

En réseau, c'est la même chose. La langue que les humains parlent, c'est un protocole pour les hôtes dans un réseau. Pas n'importe quel protocole, car il en existe plusieurs. Mais celui qui nous concerne est appelé « **protocole de communication** ».

Quant à l'interprète de notre exemple, dans un réseau, ce sera la passerelle (applicative) qui permettra de faire communiquer deux réseaux basés sur des protocoles différents en assurant plusieurs fonctions telles que la traduction des protocoles et des signaux, l'isolation d'erreurs, l'adaptation d'impédances, etc.

Si vous ne comprenez pas ces termes techniques, ce n'est pas important pour l'instant.

II.1.2. L'utilité d'un protocole par l'exemple

Bien ! Vous avez compris le concept de protocole ? Maintenant essayons de voir à quoi ça sert dans un réseau. Pour comprendre cela, très souvent, on utilise une analogie que nous qualifions de « classique » en réseau, car plusieurs professeurs utilisent presque toujours cette dernière pour faire assimiler les fonctions assurées par un protocole. Il s'agit de la communication téléphonique entre deux humains.

Pierre veut transmettre un message à Jean. Il compose donc son numéro de téléphone et il peut entendre la tonalité (tuuuut... tuuuut...). Il attend que Jean décroche, car la communication ne peut avoir lieu qu'à ce moment-là. Jean, de son côté, entend son téléphone sonner. Il décroche, et c'est là qu'intervient le classique « Allô? ». 🍊

À ce niveau, la « session de communication » est établie, c'est-à-dire que Pierre peut maintenant dire à Jean ce qu'il a en tête. Il va donc gentiment se présenter : « **Salut, c'est Pierre...** » et évoquer le contexte ou la raison de son appel : « **C'était juste pour te dire que demain, il y aura une fête chez Anne-Sophie, qui habite au numéro 10 de la rue Lézard!** ».

Jean peut éventuellement demander à Pierre de répéter, pour être sûr d'avoir bien saisi son message : « **Chez qui ? Anne qui ?** ». Alors Pierre répétera cette partie pour que Jean comprenne. Finalement, la conversation terminée, il faut se séparer en douceur (🍊). Un classique « salut » ou « au revoir » des deux côtés avant qu'ils ne raccrochent leurs combinés.

Les protocoles nous permettent de faire tout ça. Essayons un peu de réexaminer ce scénario avec un langage un peu plus informatique. 🍊

Pierre veut transmettre un message à Jean.

Il compose donc son numéro de téléphone et il peut entendre la tonalité (tuuuut... tuuuut...). Il attend que Jean décroche, car la communication ne peut avoir lieu qu'à ce moment-là.

L'hôte Pierre, à l'adresse IP 124.23.42.13, souhaite communiquer avec l'hôte Jean à l'adresse IP 124.23.12.13. Il lui enverra un paquet de demande d'initialisation de session (il compose son numéro et attend que Jean décroche et dise « Allô »). À ce stade, il peut se passer quatre choses dans le contexte naturel :

1. Le numéro est incorrect.
2. Le numéro est correct mais indisponible.
3. Le numéro est correct et Jean décroche en disant « Allô ».
4. Le numéro est correct, disponible, mais Jean ne décroche pas (c'est donc un peu comme le cas 2 🍊).

Étudions ces cas :

- Cas 1 : Pierre aura un message vocal disant « **Le numéro que vous avez composé n'existe pas** ». En réseau ce sera un paquet **ICMP** (*Internet Control Message Protocol*) qui enverra une **erreur de type 3 (destination unreachable, destination inaccessible)** et de **code 7 (destination host unknown, destinataire inconnu)**.

i

ICMP est un protocole dans la suite protocolaire **TCP-IP** utilisé pour envoyer des messages d'erreurs dans un réseau. Il travaille en partenariat avec le protocole **IP**. Nous allons le voir en détail, voir les différents types d'erreurs, leurs codes, leurs significations et les scénarios dans lesquels elles se manifestent.



- Cas 2 : Ici, un message vocal dira à Pierre « L'abonné que vous souhaitez appeler est injoignable pour l'instant, veuillez rappeler dans quelques instants ». En réseau, il s'agira également d'une erreur de type 3.
- Cas 3 : Si le numéro est correct et que Jean décroche en disant « Allô », c'est le début de la conversation. En réseau on dira donc qu'une session a été initialisée. 🍊
- Cas 4 : Ici, classiquement, ce sera le répondeur de Jean qui dira « **Je ne suis pas disponible pour l'instant, laissez-moi un message, je vous rappellerai dès que possible** ». En réseau, c'est un peu différent. L'hôte Pierre va recevoir **une erreur ICMP de type 3 (destination inaccessible) et de code 1 (destinataire inaccessible)**. En gros, c'est pour dire qu'on n'arrive pas à atteindre le destinataire. En fait, si un numéro de téléphone est disponible, sonne, mais que personne ne répond, ça veut dire qu'on n'a pas atteint le destinataire final en fait. Donc c'est un peu pareil que le cas 2.

Continuons l'analyse de notre analogie. 🍊

« C'était juste pour te dire que demain, il y aura une fête chez Anne-Sophie, qui habite au numéro 10 de la rue Léopard ».

Jean peut éventuellement demander à Pierre de répéter, pour être sûr d'avoir bien saisi son message « Chez qui ? Anne qui ? ». Alors Pierre répétera cette partie pour que Jean comprenne.

Si Jean demande à Pierre de répéter quelque chose, de façon radicale on peut conclure qu'il n'a pas **reçu** ce que Pierre a dit (si l'on considère que recevoir un message = comprendre le message). En réseau, l'hôte Jean va envoyer un paquet à Pierre disant « **je n'ai pas reçu le dernier paquet, renvoie-le stp** ». Pierre va alors renvoyer le dernier paquet. En fait, c'est un peu plus précis que ça. Suivant le protocole que vous utilisez (**UDP** ou **TCP**, nous allons les comparer dans les prochains chapitres), Pierre peut demander à la fin de chaque phrase si Jean a compris. En réseau, l'hôte Pierre pourrait donc demander un message d'accusé de réception à chaque envoi de paquet, et l'hôte Jean devra répondre « **oui j'ai reçu, envoie le prochain** » tout le long de la communication si l'on utilise le protocole **TCP** qui est dit **connection-oriented (orienté connexion)** par opposition au protocole **UDP** qui est dit **connectionless-oriented**. Tenez-vous tranquille, avec **TCP** on peut faire encore plus fort que ça.

Qu'est-ce qui se passe, si Pierre se met à raconter sa vie à raconter une histoire à Jean et que ce dernier dépose le combiné et s'en va faire un tour aux toilettes sans prévenir ? Pierre aurait perdu son temps en parlant pour rien ! Pour prévenir ce genre de chose, Pierre peut vérifier la présence de Jean en demandant toutes les x minutes « **Tu me suis ? Tu es là ?** ». En réseau, avec **TCP** il s'agit d'une **vérification périodique de l'état de la session de communication**. Ceci dit, l'hôte Pierre enverra un paquet de « **vérification de session** » pour savoir si l'hôte Jean est toujours connecté. Si Jean ne répond pas après un certain laps de temps, la communication est terminée (la session se termine là).



Ici, nous sommes dans l'explication de ce que fait le protocole **TCP**. Vous n'étiez pas censé le savoir, c'était juste pour vous illustrer le fonctionnement des protocoles sans vous dire duquel il s'agissait. Mais nous avons préféré vous le dire, car nous faisons allusion à des paquets ici, mais en fait il s'agit des valeurs précises qui se trouvent dans l'en-tête des paquets **TCP**.



Finalement, la conversation terminée, il faut se séparer en douceur. 🍊 Un classique « salut » ou « au revoir » des deux côtés avant qu'ils ne raccrochent leurs combinés.

À ce stade, la session de communication est terminée.

II.1.3. Les exigences d'un protocole

Un protocole de communication digne de ce nom doit remplir quelques exigences rigoureuses. Un protocole est un ensemble de règles dictant comment doit s'effectuer la communication entre deux entités. Ceci dit, il faudrait que ledit protocole soit en mesure d'assurer des fonctions vitales au bon déroulement d'une communication. Il existe plusieurs « fonctions vitales » (comprendre exigences) qu'un protocole de communication doit être capable de remplir. Dans la sous-partie précédente, nous avons vu quelques-unes de ces fonctions le long de l'exemple sans vous les pointer directement. Parmi ces fonctions figurent en bonne et auguste posture :

- **La gestion du format des données** : un protocole, comme nous l'avons répété, définit comment s'effectue la communication. Or, qui dit communication dit échanges de données. Le protocole doit donc avoir des « fonctions » permettant de gérer le format de ces données. Nous verrons plus tard dans quelle couche du modèle OSI on trouve ces services de formatage. En général, les données seront constituées de deux choses : d'un entête et du contenu. L'entête sera un peu « réservé » au protocole. C'est à ce niveau que l'on trouve des informations « techniques » tandis que le contenu... bah, c'est le contenu! 🍊
- **La gestion du format d'adresses** : durant la procédure de transmission des données, il faudrait bien gérer les adresses : qui est l'émetteur, qui est le destinataire? Dans une communication dans le monde naturel, quand on écrit une lettre, dans l'entête, on met l'adresse de l'émetteur et celle du destinataire, et même sur l'enveloppe d'ailleurs. Si on ne le fait pas, on ne sait pas à qui envoyer la lettre, et celui qui la reçoit ne sait même pas si elle lui est destinée et de qui elle provient. Par comparaison, dans l'entête des données « encapsulées », il faudrait qu'un protocole soit en mesure de spécifier l'adresse de l'émetteur et du destinataire.
- **Correspondance d'adresses** : quand vous inscrivez l'adresse du destinataire sur une enveloppe, cette dernière est « logique ». Logique dans ce sens que le destinataire n'habite pas sur cette enveloppe (🍊), mais cette adresse indique l'adresse physique du destinataire, là où vous pouvez le trouver si vous vous y rendez physiquement. 🍊 Le facteur doit donc faire une correspondance entre cette adresse logique sur l'enveloppe et l'adresse physique. Par analogie, un protocole doit assurer des fonctions de correspondance entre les adresses logiques (**IP**) et les adresses physiques (**MAC**). Cette correspondance s'appelle « **address mapping** » en anglais. 🍊

- **Routage** : nous allons passer un long moment sur ce sujet dans la suite de ce tuto. Dit simplement, le routage consiste à « diriger » les données entre deux réseaux d'un plan d'adressage différent.
- **Détection d'erreurs de transmission** : il se peut qu'une erreur se produise dans la procédure de transmission des informations. Un protocole devrait donc être en mesure de détecter ces erreurs. Comme nous allons le voir, il s'agit d'un **CRC (Cyclic Redundancy Check, Contrôle de Redondance Cyclique)** qui est ajouté à la fin des paquets.
- **Accusé de réception** : quand vous recevez un mail, très souvent vous y répondez. Cette réponse informe explicitement à l'émetteur que vous avez reçu son mail. C'est en quelque sorte un accusé de réception. Certains protocoles permettent donc à un hôte récepteur d'informer un hôte émetteur qu'il a reçu le paquet envoyé pour empêcher ce dernier de renvoyer les mêmes choses. D'autres par contre n'implémentent pas cette fonction.
- **La gestion de perte d'informations** : de même que des erreurs peuvent se produire lors de la transmission, il peut y avoir des pertes d'informations. Pertes ? Dans un réseau ? Oui ! Généralement quand un paquet met trop du temps à arriver à son destinataire, « **il se perd** ». :D Voilà pourquoi c'est important qu'un protocole gère la reconnaissance des paquets. Si l'hôte-récepteur B répond dans un intervalle de x secondes à l'hôte-émetteur A, ce dernier saura alors que B a bien reçu les données, et n'essaiera plus de les renvoyer. Si B par contre ne répond pas à A, ce dernier peut donc conclure que les données « **se sont perdues** » et va les renvoyer dans un espace de temps déterminé par le protocole.
- **La direction du flux d'informations** : A et B peuvent-ils communiquer (s'échanger des données) simultanément ? Si oui, il s'agit d'un système de communication **full-duplex**. Sinon, il s'agit d'un système de communication **half-duplex**. Nous allons en parler un peu plus tard dans cette partie du cours. 🍌 Un protocole doit donc dicter la direction de flux dans la communication pour empêcher à deux hôtes de communiquer simultanément dans un système *half-duplex* par exemple.
- **Contrôle de séquences** : toute information envoyée sur un réseau est segmentée en plusieurs « séquences » (nous y reviendrons). Elles sont ensuite envoyées au destinataire. Selon la congestion (le degré d'occupation) des routes qu'elles vont emprunter, elles peuvent arriver « en désordre », ou même en double (s'il y a eu des retransmissions). Grâce au contrôle de séquences d'un protocole, on peut « numéroté » chaque « morceau » afin que le destinataire sache les « remettre en ordre » ou supprimer les doublons. Nous allons voir comment fonctionne cette « segmentation » en étudiant le protocole **BitTorrent**.
- **Gestion de flux** : quand deux personnes parlent, il est nécessaire de donner à celui qui « écoute » le temps de comprendre ce qui est dit, puisqu'il se peut que l'émetteur parle plus vite que le récepteur. Il faut donc gérer cette volubilité, ce flux. Dans les réseaux, il y a des cas où un hôte-émetteur peut transmettre plus vite que ne peut recevoir un hôte-récepteur. C'est là qu'intervient l'utilité de la gestion des flux.



Un seul protocole peut faire tout ça?




Mais non ! :D Les fonctions citées ne peuvent pas être réalisées par un seul protocole. Il s'agit d'une suite protocolaire, une suite de protocoles. Il y a des protocoles qui s'occupent de la transmission, d'autres du routage, etc. Une suite de protocoles est un ensemble de protocoles fonctionnant en harmonie et cohésion pour le bon déroulement de la communication. Vous avez

déjà entendu l'expression « **protocole TCP/IP** » ? En fait, ce n'est pas un protocole. **TCP** en est un, **IP** en est un autre. Mais **TCP/IP**, ça fait deux. :D C'est une suite (une **pile** pour être précis) de protocoles en fait, **protocol stack** en anglais. 🍊

Voilà, les bases sont posées ! Vous savez ce qu'est un protocole, maintenant. Tout au long du cours, nous allons parler des protocoles les plus courants et importants. Mais avant cela, nous allons survoler un peu le modèle OSI, qui est à la base de la plupart des communications informatiques.

II.2. Ils en tiennent une couche : OSI et TCP/IP

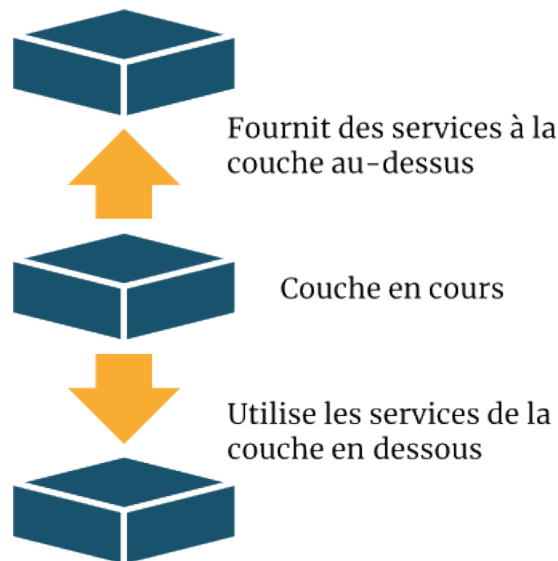
Ne soyez pas déçus ! Nous n'en sommes qu'au début du cours, alors ce chapitre sera plus une introduction aux modèles que *Le Modèle OSI de A à Z en vingt-cinq volumes...* Mais ne vous inquiétez pas, vous aurez déjà fort à faire! 

II.2.1. Le modèle OSI en douceur

Dans cette sous-partie, nous allons définir le plus simplement possible ce qu'est le modèle OSI. En effet, vous allez le comprendre, il n'y a aucun rapport avec la mode ni la 3D (si si, nous vous le jurons).

II.2.1.1. Qu'est-ce que le modèle OSI?

Le modèle OSI (*Open Systems Interconnection* : « interconnexion de systèmes ouverts ») est une façon standardisée de segmenter en plusieurs blocs le processus de communication entre deux entités. Chaque bloc résultant de cette segmentation est appelé couche. Une couche est un ensemble de services accomplissant un but précis. La beauté de cette segmentation, c'est que chaque couche du modèle OSI communique avec la couche au-dessus et au-dessous d'elle (on parle également de couches adjacentes). La couche au-dessous fournit des services que la couche en cours utilise, et la couche en cours fournit des services dont la couche au-dessus d'elle aura besoin pour assurer son rôle. Voici un schéma pour illustrer ce principe de communication entre couches :



zestedesavoir.com | Les réseaux de zéro

FIGURE II.2.1. – Représentation schématique d'un modèle en couches (CC BY)

Ainsi le modèle OSI permet de comprendre de façon détaillée comment s'effectue la communication entre un ordinateur A et un ordinateur B. En effet, il se passe beaucoup de choses dans les coulisses entre l'instant t , où vous avez envoyé un mail (par exemple), et l'instant $t+1$, où le destinataire le reçoit.

Le modèle OSI a segmenté la communication en sept couches :

- Application (ou couche applicative).
- Présentation.
- Session.
- Transport.
- Réseau.
- Liaison de données.
- Physique.

Une façon efficace de connaître ces couches par cœur, de haut en bas et en anglais, serait de mémoriser la phrase suivante : **All People Seem To Need Data Processing**, ce qui signifie :

« Tout le monde a besoin du traitement de données. » Chaque majuscule représente la première lettre d'une couche : A pour application, P pour présentation, S pour session, T pour transport, N pour réseau (*network* en anglais), D pour *data* (liaison de données) et finalement le dernier P (*processing*) pour physique.

De bas en haut, le moyen mnémotechnique anglais utilisé est **Please Do Not Throw Sausage Pizza Away**. Ce qui donne en français : « S'il vous plaît, ne jetez pas les saucisses de pizza.

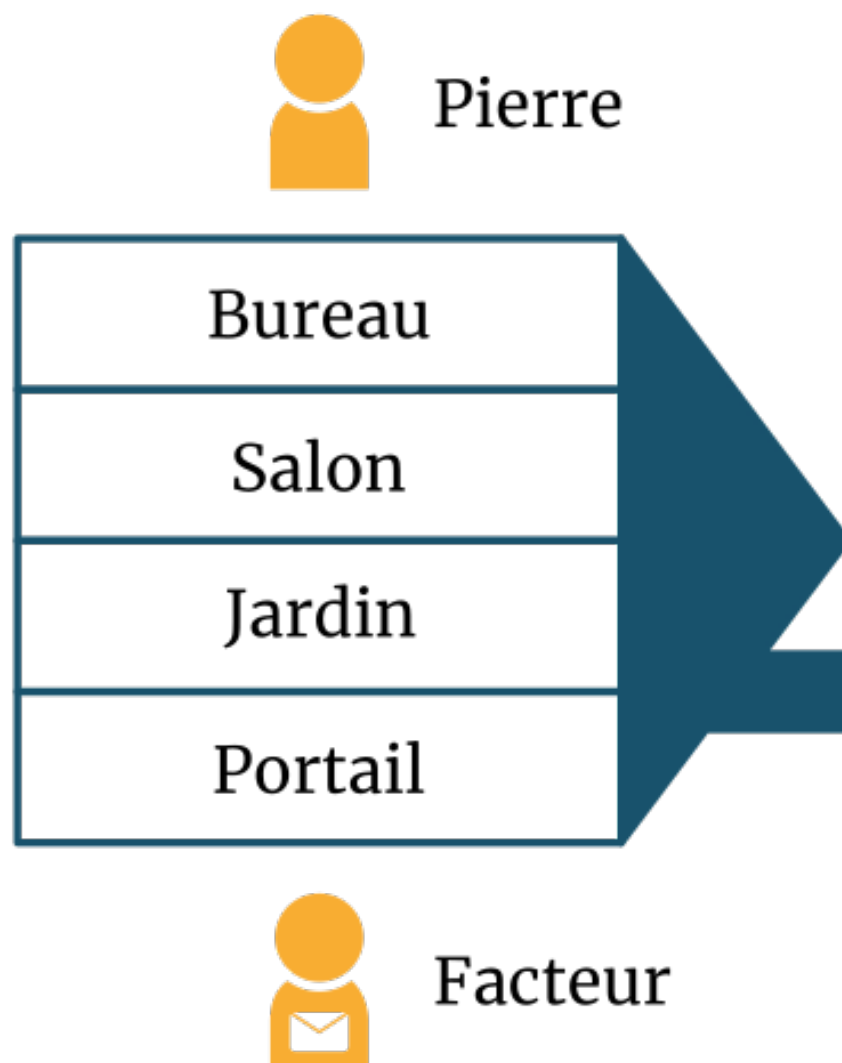
» Ces sacrés anglophones ont des inspirations hilarantes ! 🍌 On trouve d'autres moyens mnémotechniques en français pour retenir ces couches de bas en haut : **Partout Le Roi Trouve Sa Place Assise**, ou encore : **Pour Le Réseau, Tout Se Passe Automatiquement**.

II.2.2. Le modèle OSI par l'exemple : le facteur

Oui, nous le savons, vous êtes impatients ; néanmoins, allons-y lentement mais sûrement. 🍌 Nous n'allons rien vous enseigner de trop complexe, rassurez-vous. Nous avons pris l'habitude de toujours illustrer nos propos par un exemple concret, une analogie parlante.

Pour comprendre le modèle OSI, nous allons inventer un scénario. Vous vous souvenez de Pierre et de Jacques? Oui, nos camarades d'antan! Pierre garde une lettre dans son bureau. Il veut la donner au facteur, qui attend devant le portail de sa belle villa. La lettre est destinée à Jacques, mais Pierre n'a pas le droit d'entrer dans le bureau de Jacques. Jacques non plus n'a pas le droit de sortir de son bureau. Seul le facteur peut entrer dans le bureau de Jacques pour délivrer la lettre, mais il lui est interdit d'aller dans celui de Pierre pour la chercher.

La maison de Pierre est mal construite : il n'y a pas de couloir, juste un alignement vertical de pièces séparées par une porte. Pour aller du bureau au portail, Pierre doit traverser le salon et le jardin. Schématiquement, cela donne ceci :



zestedesavoir.com | Les réseaux de zéro

FIGURE II.2.2. – Une maison presque réaliste (CC BY)

Dans le schéma ci-dessus, chaque pièce de la maison peut être considérée comme une couche. Pierre doit quitter la couche la plus élevée pour se diriger vers la plus basse (le portail). Une fois la lettre remise au facteur, ce dernier devra faire l'inverse chez Jacques, c'est-à-dire quitter la couche la plus basse pour aller vers la couche la plus élevée (le bureau de Jacques).

Chaque pièce de la maison possède une fonction précise. Le bureau est généralement réservé au travail; le salon, à la distraction (discussions, télévision, etc.). Le jardin, lui, nous offre sa beauté et son air pur. Quant au portail, il permet d'accéder aussi bien au jardin qu'à la maison.

Faisons intervenir un autre personnage, Éric, dans notre histoire. Éric ne connaît absolument rien au processus de transfert de lettres. Alors quand Pierre lui dit : « J'ai écrit une lettre à Jacques », Éric imagine le scénario suivant :

- Pierre a écrit la lettre.
- Il l'a envoyée.
- Jacques a reçu la lettre.

Éric, c'est un peu vous avant la lecture de ce tutoriel. 🍊 Vous pensiez sans doute qu'après avoir envoyé un mail, par exemple, M. le destinataire le recevait directement. Mais vous venez de comprendre grâce à l'exemple de la lettre que votre mail est passé par plusieurs couches avant d'arriver au destinataire. Cet exemple vous semble peut-être aberrant, mais nous pensons qu'il a aidé plusieurs personnes à mieux concevoir le principe du modèle OSI.



Pour illustrer ce processus et faciliter votre compréhension, nous n'avons abordé que quelques couches du modèle OSI en faisant appel à un facteur. N'en déduisez pas quoi que ce soit !

II.2.3. Survol des couches du modèle OSI

Nous y sommes presque ! Nous allons regarder le modèle OSI d'un œil plus technique, cela devrait vous plaire ! 🍊 Le modèle OSI est donc constitué de sept couches distinctes. Dans chacune de ces couches opèrent un certain nombre de protocoles.

II.2.3.1. Comment ça fonctionne ?

Lorsque vous voulez envoyer un mail à l'équipe des rédacteurs de ce tutoriel (comment ça, ça ne vous tente pas ? 🍊), plusieurs choses se passent en coulisses.

II.2.3.1.1. Couche applicative

Vous avez besoin d'accéder aux services réseaux. La couche applicative fait office d'interface pour vous donner accès à ces services, qui vous permettent notamment de transférer des fichiers, de rédiger un mail, d'établir une session à distance, de visualiser une page web... Plusieurs protocoles assurent ces services, dont FTP (pour le transfert des fichiers), Telnet (pour l'établissement des sessions à distance), **SMTP** (pour l'envoi d'un mail), etc.

II.2.3.1.2. Couche présentation

Il vous faut formater votre mail pour une bonne présentation. C'est dans la couche... présentation que cela se passe. Elle s'occupe de la sémantique, de la syntaxe, du chiffrement/déchiffrement, bref, de tout aspect « visuel » de l'information. Un des services de cette couche, entre autres : la conversion d'un fichier codé en EBCDIC (*Extended Binary Coded Decimal Interchange Code*) vers un fichier codé en **ASCII** (*American Standard Code for Information Interchange*).

i

Le chiffrement peut être pris en charge par une autre couche que la couche de présentation. En effet, il peut s'effectuer dans la couche application, transport, session, et même réseau. Chaque niveau de chiffrement a ses avantages.

i

Certains protocoles, tels que le HTTP, rendent la distinction entre la couche applicative et la couche de présentation ambiguë. Le HTTP, bien qu'étant un protocole de la couche applicative, comprend des fonctionnalités de présentation comme la détection du type de codage de caractères utilisé.

II.2.3.1.3. Couche session

Une fois que vous êtes prêt(e) à envoyer le mail, il faut établir une session entre les applications qui doivent communiquer. La couche session du modèle OSI vous permet principalement d'ouvrir une session, de la gérer et de la clore. La demande d'ouverture d'une session peut échouer. Si la session est terminée, la « reconnexion » s'effectuera dans cette couche.

II.2.3.1.4. Couche transport

Une fois la session établie, le mail doit être envoyé. La couche de transport se charge de **préparer** le mail à l'envoi. Le nom de cette couche peut prêter à confusion : elle n'est pas responsable du transport des données proprement dit, mais elle y contribue. En fait, ce sont les quatre dernières couches (transport, réseau, liaison de données et physique) qui toutes ensemble réalisent le transport des données. Cependant, chaque couche se spécialise. La couche de transport divise les données en plusieurs segments (ou séquences) et les réunit dans la couche transport de l'hôte récepteur (nous y reviendrons). Cette couche permet de choisir, en fonction des contraintes de communication, la meilleure façon d'envoyer une information. « Devrai-je m'assurer que la transmission a réussi, ou devrai-je juste l'envoyer et espérer que tout se passe bien ? Quel port devrai-je utiliser ? » La couche de transport modifie également l'en-tête des données en y ajoutant plusieurs informations, parmi lesquelles les numéros de ports de la source et de la destination. Le protocole **TCP** (*Transmission Control Protocol*) est le plus utilisé dans la couche de transport.

II.2.3.1.5. Couche réseau

Maintenant que nous savons quel numéro de port utiliser, il faut aussi préciser l'adresse **IP** du récepteur. La couche réseau se charge du routage (ou relai) des données du point A au point B et de l'adressage. Ici aussi, l'en-tête subit une modification. Il comprend désormais l'en-tête ajouté par la couche de transport, l'adresse **IP** source et l'adresse **IP** du destinataire. Se fait également dans cette couche le choix du mode de transport (mode connecté ou non connecté, nous y reviendrons là encore). Le protocole le plus utilisé à ce niveau est bien sûr le protocole **IP**.

II.2.3.1.6. La couche liaison

Présentation effectuée ? O.K. !

Session établie ? O.K. !

Transport en cours ? O.K. !

Adresses **IP** précisées ? O.K. !

Il reste maintenant à établir une liaison « physique » entre les deux hôtes. Là où la couche réseau effectue une liaison logique, la couche de liaison effectue une liaison de données physique. En fait, elle transforme la couche physique en une liaison, en assurant dans certains cas la correction d'erreurs qui peuvent survenir dans la couche physique. Elle fragmente les données en plusieurs trames, qui sont envoyées une par une dans un réseau local. Par conséquent, elle doit gérer l'acquittement des trames (nous... enfin bref, ce chapitre n'est qu'une introduction, vous l'avez compris 🍊). Quelques exemples de protocoles de cette couche : Ethernet, PPP (*Point to Point Protocol*), HDLC (*High-Level Data Link Control*), etc.

i

La couche 2 assure la livraison des trames dans un réseau local. Cela dit, elle utilise des adresses physiques, la transmission des données au-delà du réseau local ne peut donc pas être gérée à ce niveau. Logique, quand on y pense : c'est le rôle de la couche 3. Tous les protocoles de cette couche n'ont pas forcément la possibilité de gérer l'acquittement des trames, qui se fait alors dans une couche supérieure.

II.2.3.1.7. Finalement : la couche physique

Notre mail est en cours de transport, mettons-le sur le média. La couche physique reçoit les trames de la couche de liaison de données et les « convertit » en une succession de *bits* qui sont ensuite mis sur le média pour l'envoi. Cette couche se charge donc de la transmission des signaux électriques ou optiques entre les hôtes en communication. On y trouve des services tels que la détection de collisions, le *multiplexing*, la modulation, le *circuit switching*, etc.

II.2.3.2. Résumé

Nous avons abordé, en donnant quelques détails, chacune des couches du modèle OSI ; voici un tableau récapitulatif.

Position dans le modèle OSI	Nom de la couche	Rôle de la couche
7	Application	Point de contact avec les services réseaux.
6	Présentation	Elle s'occupe de tout aspect lié à la présentation des données : format, chiffrement, encodage, etc.
5	Session	Responsable de l'initialisation de la session, de sa gestion et de sa fermeture.
4	Transport	Choix du protocole de transmission et préparation de l'envoi des données. Elle spécifie le numéro de port utilisé par l'application émettrice ainsi que le numéro de port de l'application réceptrice. Elle fragmente les données en plusieurs séquences (ou segments).
3	Réseau	Connexion logique entre les hôtes. Elle traite de tout ce qui concerne l'identification et le routage dans le réseau.
2	Liaison de données	Établissement d'une liaison physique entre les hôtes. Fragmente les données en plusieurs trames.
1	Physique	Conversion des trames en <i>bits</i> et transmission physique des données sur le média.

II.2.3.3. Processus de transmission/réception

Quand un hôte A envoie un message à un hôte B, le processus d'envoi va de la couche 7 (application) à la couche 1 (physique). En revanche, quand il s'agit de recevoir, le message emprunte le chemin inverse : il part de la couche 1 (physique) pour arriver à la couche 7 (application). Souvenez-vous de l'exemple de Pierre, Jacques et le facteur : Pierre quittait le salon pour le portail afin d'envoyer sa lettre, alors que le facteur quittait le portail et se dirigeait vers le bureau de Jacques pour la délivrer.

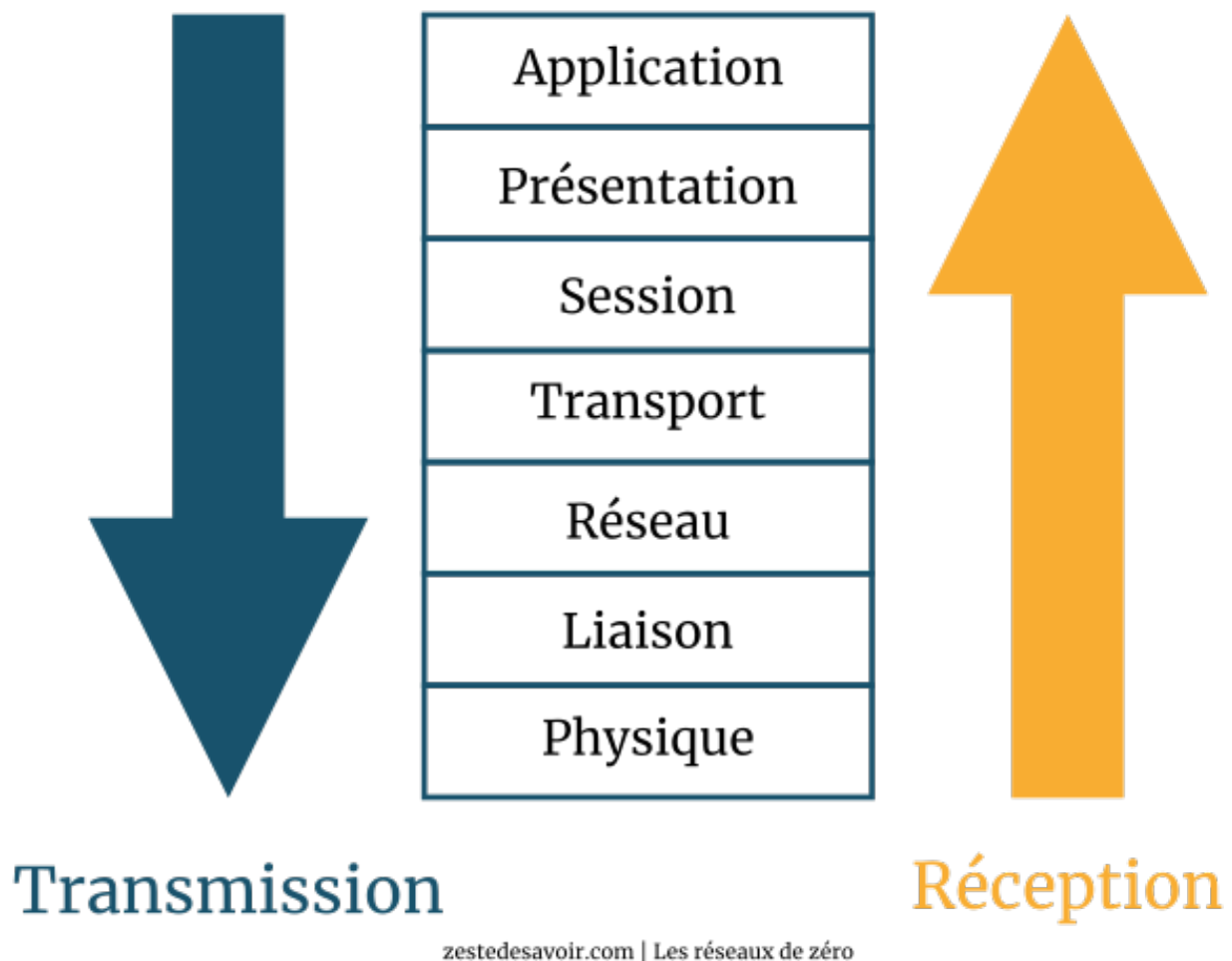


FIGURE II.2.3. – Traversée des couches pour la transmission et la réception (CC BY)

II.2.4. TCP/IP vs OSI : le verdict?

Vous vous êtes peut-être posé la question de savoir pourquoi le titre de ce chapitre était *Ils en tiennent une couche : OSI et TCP/IP*, alors que ce dernier semble être passé aux oubliettes. Nous allons bien étudier deux modèles différents : TCP/IP et OSI. Nous allons commencer par revoir leurs origines et le but de leur création, ensuite nous comparerons leurs architectures respectives.

II.2.4.1. Il y a une génération...

Le modèle TCP/IP fut créé dans les années 1970 par le département de la Défense des États- Unis d'Amérique, plus précisément par l'agence DARPA (*Defense Advanced Research Projects Agency*). C'est pour cette raison que vous le trouverez aussi sous l'appellation *DoD Model* pour *Department of Defense Model* (« modèle du département de la Défense »). Quant au modèle OSI, il a été créé en 1978 par l'Organisation internationale pour la standardisation (ou ISO, *International Organization for Standardization*). C'est un certain Charles Bachman qui proposa la segmentation de la communication dans un réseau en sept couches distinctes.

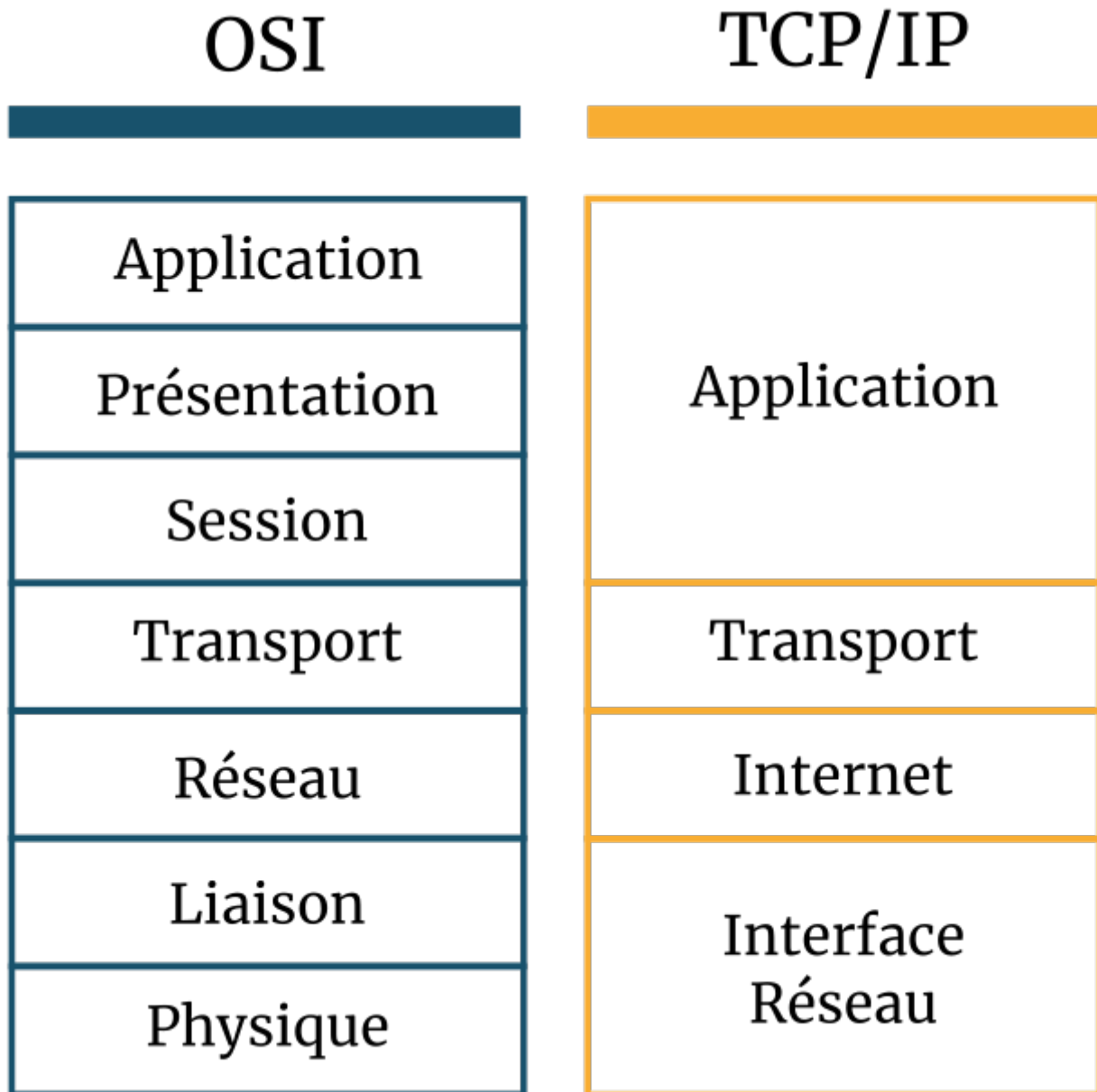
Les buts de ces deux modèles ne sont pas les mêmes. En effet, le modèle OSI a été développé à vocation normative, c'est-à-dire pour servir de référence dans le déroulement de la communication entre deux hôtes. D'ailleurs, il est également connu sous les noms *OSI Reference model* (« modèle de référence OSI ») ou OSI-RM. Alors que le modèle TCP/IP a une vocation descriptive, c'est-à-dire qu'il décrit la façon dont se passe la communication entre deux hôtes. En d'autres termes, si vous voulez comprendre comment se déroule la communication « sur le terrain », prenez le modèle TCP/IP. Par contre, si vous voulez comprendre la suite logique, la procédure selon la norme, penchez-vous sur le modèle OSI. Ceci dit, c'est le modèle OSI qui vous servira de « plan » si vous voulez créer un protocole ou un matériel en réseau.

i

Il se peut qu'*Internet Reference Model* fasse parfois référence au modèle TCP/IP. Cette appellation n'est pas fautive, mais inexacte : la suite protocolaire TCP/IP sert de description plutôt que de référence.

II.2.4.2. Comparaison dans la structure

Voici un schéma comparatif des deux modèles.



zestedesavoir.com | Les réseaux de zéro

FIGURE II.2.4. – Comparaison OSI vs TCP/IP (CC BY)

Comme vous le voyez, le modèle **TCP/IP** n'est constitué que de quatre couches. Ce sont des couches d'abstraction, autrement dit des couches qui cachent les détails d'implémentation de la communication et leurs noms ne reflètent pas mot pour mot les fonctions qu'elles assurent. Le modèle OSI, quant à lui, est fièrement constitué de sept couches. Les trois premières couches du modèle OSI correspondent à la couche applicative du modèle **TCP/IP**.



Cette correspondance ne veut pas dire que la couche applicative du modèle **TCP/IP** soit une synthèse des trois premières couches du modèle OSI. Non ! Elle ne remplit que les rôles des couches application et présentation du modèle OSI, comme le spécifie la [RFC 1122](#) .



Les deux modèles possèdent une couche de transport. La couche réseau du modèle OSI correspond à la couche Internet(work) du modèle TCP/IP. Les couches liaison de données et physique du modèle OSI forment une seule couche pour le modèle TCP/IP : interface réseau. Les couches application, présentation, session et transport sont dites « couches hôtes » (*host layers* en anglais). En effet, ces couches « concernent » directement les hôtes. Les couches réseau, liaison et physique, elles, sont des couches de médias (*media layers*) : elles sont plus liées au média qu'à l'hôte. Voici un schéma illustrant cette correspondance :

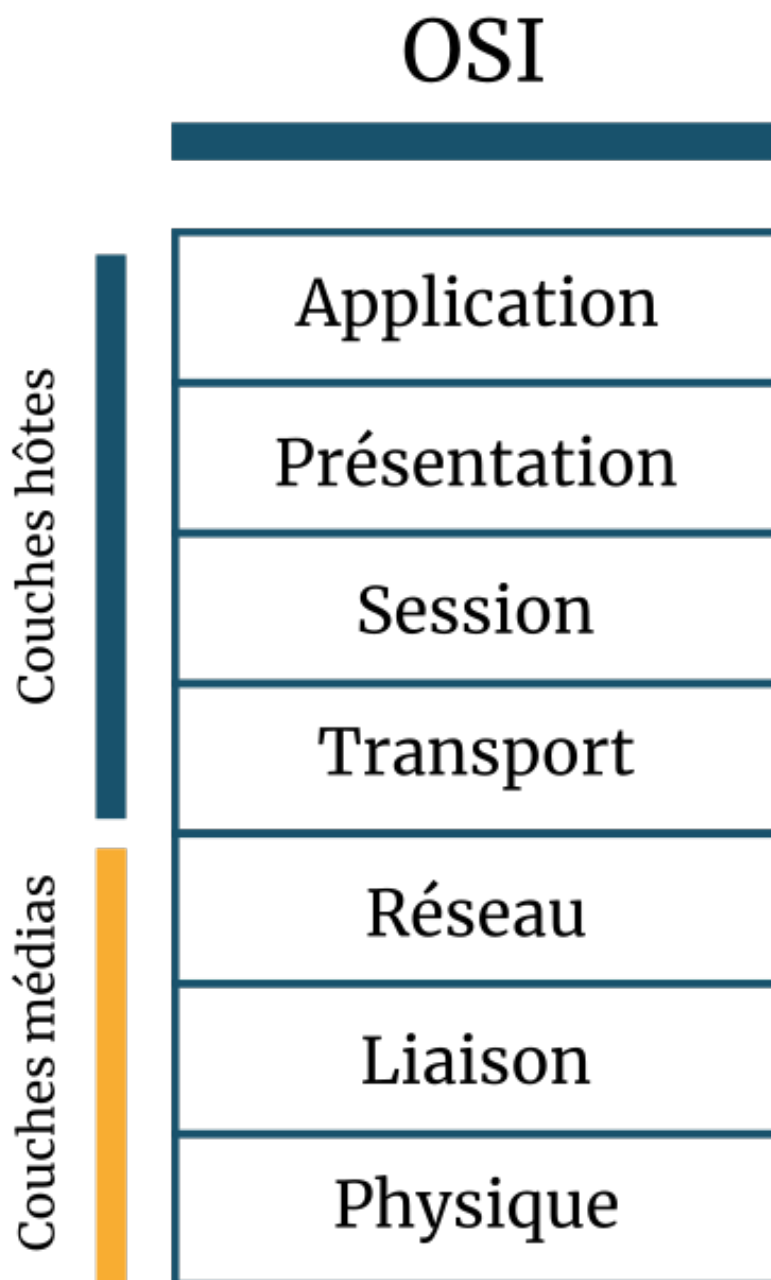


FIGURE II.2.5. – Différenciation entre couches médias et couches hôtes sur le modèle OSI (CC BY)

II.2.4.3. Point vocabulaire : les unités de données

Au début de la communication entre deux hôtes, chaque information qui sera transmise est une donnée. Cependant, cette donnée a plusieurs unités selon la couche dans laquelle elle se trouve : il s'agit de la même donnée, mais sous plusieurs appellations. Prenons un exemple : votre père, vous l'appellez papa à la maison. Au travail, on l'appelle M. X ; chez son frère, ses neveux l'appellent tonton, etc. C'est bien la même personne, connue sous plusieurs appellations selon le milieu.

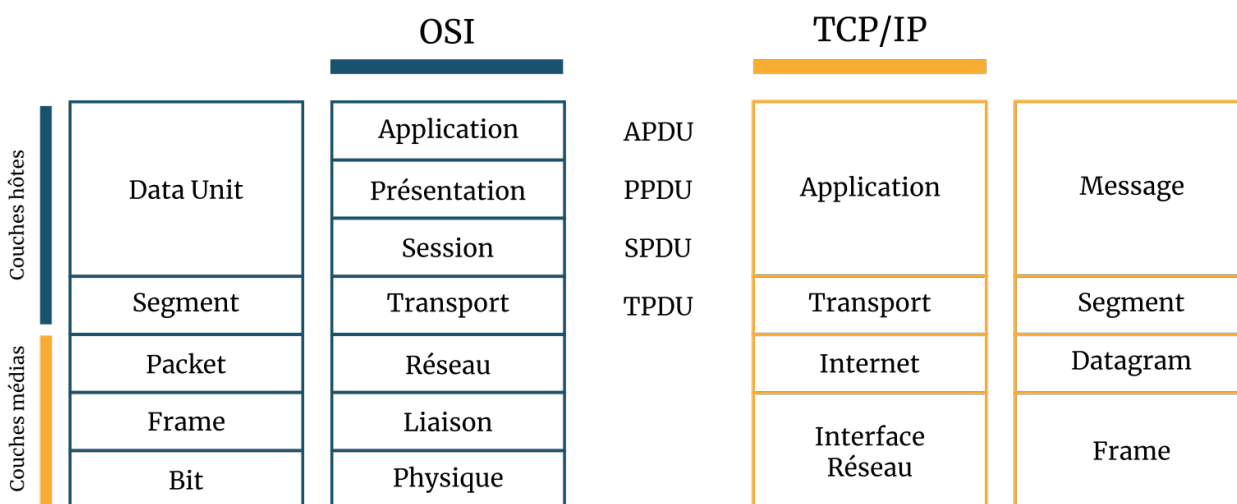
Ainsi, les données que vous transmettez sont tout simplement appelées unité de données (*data unit* en anglais). On les nomme parfois PDU (*Protocol Data Unit* : « unité de données de protocole »); dans ce cas, leur nom sera précédé de l'initiale de la couche dont ces données sont issues. Par exemple dans la couche applicative, elles prennent le nom d'APDU (*Application Protocol Data Unit* : « unité de données de protocole d'application »). Dans la couche de session, elles s'appelleront donc... SPDU (*Session Protocol Data Unit* : « unité de données de protocole de session »). Même principe pour la couche de présentation. Une fois dans la couche de transport, où elles sont segmentées, ces données deviennent logiquement des segments. (Nous les avons appelés séquences dans le chapitre précédent.)



L'appellation TPDU (*Transport Protocol Data Unit*) est également correcte en ce qui concerne la couche de transport.

Dans la couche réseau du modèle OSI, ces données prennent le nom de paquets; dans les couches liaison et physique, respectivement ceux de *frame* (trame) et *bit*.

Voici une image résumant cela pour votre plus grand plaisir. 🍊 Les acronymes dans l'image ci-dessous sont en anglais parce qu'ils sont plus courts. 🍌 Vous ne devriez pas avoir de difficulté à les comprendre puisque leurs équivalents français sont juste plus haut.



zestedesavoir.com | Les réseaux de zéro

FIGURE II.2.6. – Unités de données selon les modèles OSI et TCP/IP (CC BY)

i

Vous pouvez remarquer la présence de *datagram* dans le schéma. *Datagram* (datagramme) est le nom donné à un PDU transmis par un service non fiable (UDP par exemple). On pourra, dans certains cas, retrouver ce terme sur la couche transport. Mais le moment n'est pas encore venu.



i

Tout au long du tutoriel, nous utiliserons souvent les mots «données» et «paquets» pour faire référence à toute information qui se transmet. Vous vous rendrez vite compte qu'il est difficile de faire autrement. L'utilisation du mot approprié interviendra lorsqu'elle sera de rigueur.

II.2.4.4. Faites attention à l'abstraction des noms de couches

Les noms des couches des modèles TCP/IP ou OSI sont abstraits, voilà pourquoi nous vous avons parlé de couches d'abstraction. Leurs noms ne sont pas toujours synonymes de leurs fonctions et peuvent par moments être vagues. Par exemple, la couche application du modèle OSI ne veut pas dire grand-chose. Quand vous lisez *application*, est-ce que cela vous donne une idée de la fonction de cette couche? Ce nom n'est pas si explicite. La couche transport des deux modèles est certainement la plus abstraite dans sa dénomination. Quand on lit *transport*, on a tendance à croire que cette couche transporte vraiment les données jusqu'à son destinataire — alors que la transmission s'effectue à la couche 1 (physique) du modèle OSI et à la couche interface réseau du modèle TCP/IP. Par contre, la couche réseau est la moins abstraite, l'on comprend tout de suite qu'il s'agit de l'exercice des fonctions intimement liées au réseau.

II.2.4.5. Critiques du modèle OSI

En dehors de l'abstraction des noms de couches, dont le modèle TCP/IP est également coupable, les reproches faits à ce modèle relèvent de quatre domaines : la technologie, l'implémentation, la durée de recherche et l'investissement.

II.2.4.5.1. La technologie

Par technologie, nous voulons parler de degré de complexité. Le modèle OSI est plus complexe que le modèle TCP/IP. En effet, sept couches contre quatre : y a pas photo! 🍊 Cette complexité peut faire douter de l'utilité de certaines couches. Par exemple, les couches présentation et session sont assez rarement utilisées. Lorsque l'ISO a voulu « neutraliser » la normalisation/standardisation du modèle OSI, les Britanniques n'ont pas hésité à demander la suppression de ces couches-là. Comme nous l'avons vu en survolant les couches de ce modèle, certaines fonctions se partagent entre plusieurs niveaux. Par conséquent, la complexité même du modèle OSI réduit l'efficacité de la communication.

II.2.4.5.2. L'implémentation

À cause de la complexité de ce modèle, ses premières implémentations ont été très difficiles, lourdes et surtout lentes.

II.2.4.5.3. La durée et l'investissement

En technologie, il faut sortir le bon produit au bon moment, n'est-ce pas ? OSI n'a pas respecté cette règle. Les recherches de l'ISO pour mettre au point un modèle normalisé ont pris du temps : OSI est sorti alors que le modèle **TCP/IP** était déjà utilisé. De ce fait, l'ISO a rencontré des difficultés pour trouver un investissement, le monde n'étant pas tellement intéressé par une deuxième suite de protocoles.

II.2.4.6. Critiques du modèle TCP/IP

N'allez pas croire que le modèle **TCP/IP** est parfait ! Nous pouvons lui reprocher plusieurs choses :

- Contrairement au modèle OSI, **TCP/IP** ne distingue pas clairement le concept de services réseaux, des interfaces et des protocoles. Par conséquent, il ne respecte même pas la bonne procédure de l'ingénierie logicielle.
- Le modèle **TCP/IP** est un peu « carré ». Nous voulons dire par là qu'il est tellement spécifique que l'on ne peut pas se servir de ce modèle pour en décrire un autre, alors que le modèle OSI peut être utilisé pour décrire le principe du modèle **TCP/IP**.
- Interface réseau : c'est ainsi que l'académie Cisco appelle cette couche du modèle **TCP/IP**. La [RFC 1122](#) la nomme tout simplement lien ; on la trouve aussi sous l'appellation hôte-à-réseau (*host-to-network*). Cette couche a été fortement critiquée parce qu'il ne s'agit pas d'une couche à proprement parler, mais d'une interface entre le réseau et la liaison de données.
- Le modèle **TCP/IP** ne fait pas la distinction entre la couche physique et la couche liaison de données. En principe, la couche physique devrait être une couche à part, car elle « conclut » la transmission grâce à la mise sur média.

II.2.4.7. Et maintenant : le verdict des juges

Après avoir comparé les deux modèles, l'heure est à ~~la sanction~~ au verdict !

En conclusion à cette analyse/critique des deux modèles, il est clair que **TCP/IP** a plus de succès qu'OSI. Mais ce succès est simplement dû au fait que les protocoles de ce modèle sont les plus utilisés. Sans ses protocoles, le modèle **TCP/IP** serait pratiquement inexistant. Par contre, le modèle OSI, avec ou sans protocoles, est la parfaite norme dictant la procédure de communication. Plusieurs personnes ont sanctionné le modèle OSI au profit de **TCP/IP** et, d'après elles, **TCP/IP** gagnerait ce duel.

Le paragraphe suivant a été écrit par Jean Pouabou, auteur historique du cours

Cependant, je ne partage pas cet avis, et après quelques recherches fructueuses, je me déclare pro-OSI. Je voterais même pour le remplacement du modèle **TCP/IP**. La seule chose que je peux reprocher au modèle OSI, qui est encore d'actualité, est la présence des couches présentation et session — qui sont presque inutiles. Sans elles, le modèle OSI serait, pour moi, le modèle idéal. Cette conviction est également fondée sur le rapport analytique publié en 2004 par Internet Mark 2 Project, intitulé *Internet Analysis Report 2004 - Protocols and Governance* (« Rapport de l'analyse d'Internet - Protocoles et gouvernance »). Vous pouvez télécharger un résumé de ce rapport gratuitement [ici](#) et le rapport complet (en anglais) se trouve [là](#) .



L'analyse en soi est très critiquable. À votre niveau, vous ne serez peut-être pas capable d'en proposer une autre. Ce n'est pas grave. Cependant, notez qu'il y a matière à réflexion dans certaines remarques.



Si le modèle OSI est meilleur que le **TCP/IP**, pourquoi ce dernier a-t-il plus de succès ?

TCP/IP est sorti, et fut donc largement utilisé, avant le modèle OSI. De cette utilisation massive découle une complexité de migration vers un autre modèle, d'où le maintien du succès de **TCP/IP**.



Je ne comprends pas l'anglais, mais je veux lire le rapport de l'analyse. Une solution ?

Oui : apprendre l'anglais! 🍌

II.2.5. Principe d'encapsulation

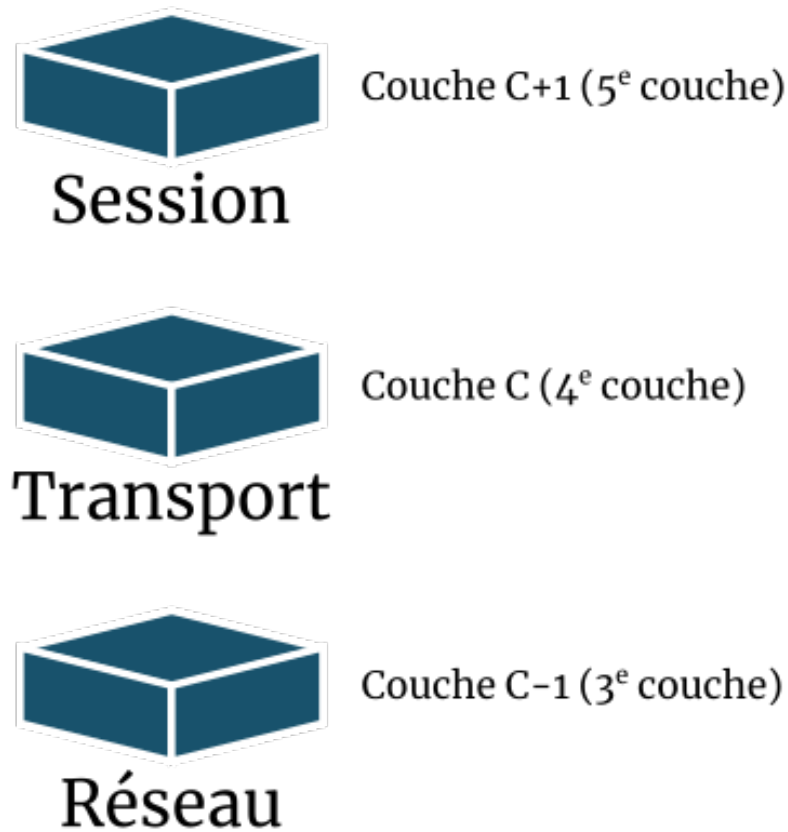
Chaque couche du modèle OSI a une fonction déterminée. Nous avons vu que la couche en cours utilise les services de la couche au-dessous d'elle qui, à son tour, en offre pour la couche du dessous. Cette corrélation indique bien que certaines informations peuvent se retrouver d'une couche à une autre. Cela n'est possible que grâce au principe d'encapsulation.

L'encapsulation consiste à encapsuler. 🍌 En d'autres termes, elle consiste à envelopper les données à chaque couche du modèle OSI.

Quand vous écrivez une lettre (pas un mail), vous devez la glisser dans une enveloppe. C'est à peu près le même principe dans le modèle OSI : les données sont enveloppées à chaque couche et le nom de l'unité de données n'est rien d'autre que le nom de l'enveloppe. Nous avons vu dans la sous-partie précédente que, dans la couche applicative, l'unité de données était l'APDU (ou plus simplement le PDU). Ensuite, nous avons vu que dans la couche réseau, l'unité de données était le paquet. Ces PDU forment une sorte d'enveloppe qui contient deux choses : la donnée en elle-même et l'en-tête spécifique à cette couche. La partie « donnée » de ce paquet est composée de la donnée initiale, mais aussi des en-têtes des couches qui la précèdent. Il existe une toute

petite formule mathématique définissant la relation entre les couches. Ce n'est pas difficile, pas la peine de fuir !

Considérons l'image ci-dessous :



zestedesavoir.com | Les réseaux de zéro

FIGURE II.2.7. – Relation entre couches (CC BY)

Soit C une couche du modèle OSI. La couche C + 1 utilise les services de la couche C. Facile, n'est-ce pas ? La couche session utilise les services de la couche transport, par exemple. La donnée que la couche C + 1 transmet à la couche C est appelée **SDU** tant qu'elle n'a pas encore été encapsulée par cette dernière. Si, par contre, la couche C encapsule ce **SDU**, on l'appelle désormais... PDU.

?

Quelle est donc la relation entre le PDU et le **SDU** ?

Dans une couche C, le PDU est le **SDU** de la couche C + 1 plus son en-tête (couche C). Ce **SDU** ne devient un PDU qu'après l'encapsulation. La couche C ajoute des informations dans l'en-tête (*header*) ou le pied (*trailer*), voire les deux, du **SDU** afin de le transformer en un PDU. Ce PDU sera alors le **SDU** de la couche C - 1. Donc le PDU est un **SDU** encapsulé avec un en-tête.

Voici la constitution d'un PDU :

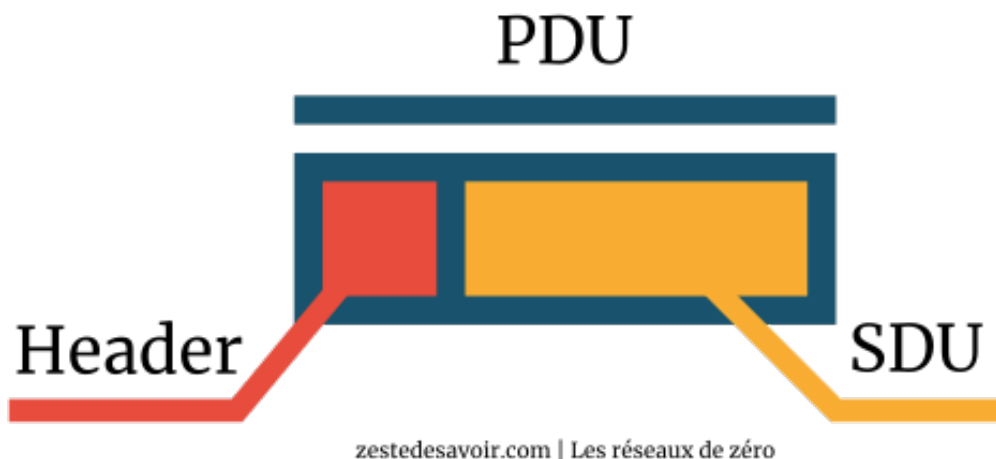


FIGURE II.2.8. – Constitution d'un PDU (CC BY)

Comprendre la relation entre un **SDU** et un PDU peut être complexe. Pour vous simplifier la tâche, nous allons considérer un exemple inspiré du monde réel et vous aurez ensuite droit à un schéma.



Nous classons l'exemple ci-dessous entre les catégories « un peu difficile » et « difficile ». Il est important de ne pas admirer les mouches qui voltigent dans votre bureau en ce moment. Soyez concentrés. 🍌

Quand vous écrivez une (vraie) lettre, c'est un **SDU**. Vous la mettez dans une enveloppe sur laquelle est écrite une adresse. Cette lettre qui n'était qu'un **SDU** devient un PDU du fait qu'elle a été enveloppée (encapsulée). Votre lettre arrive à la poste. Un agent du service postal regarde le code postal du destinataire et place la lettre dans un sac. Mais on ne la voit plus, puisqu'elle est dans un sac. Pour l'instant, la lettre, l'enveloppe et le sac forment un **SDU**. L'agent du service postal va alors inscrire le code postal du destinataire sur le sac en question, qui devient donc un PDU. S'il contient d'autres lettres partant pour la même ville, elles seront alors toutes mises dans une caisse : c'est un **SDU**. Tout comme on a ajouté des informations sur l'enveloppe et sur le sac, il faut également mettre un code postal sur la caisse. Cet ajout fait de cette caisse un PDU.

Voilà pour la procédure de transmission. Mais pour la réception, les sacs à l'intérieur de la caisse (des **SDU**) sont enlevés lorsqu'elle atteint sa destination. **Attention, c'est ici que vous devez être très attentif·ve.** Si un individu prend un sac et en lit le code postal pour l'acheminer à son destinataire, le sac **n'est plus** considéré comme un **SDU** mais comme un PDU. C'était un **SDU** au moment de sa sortie de la caisse. Étant donné qu'il y a des informations de plus sur le sac, c'est un PDU pour celui qui les lit.

Lorsque le destinataire recevra la lettre, les informations ajoutées sur le sac ou sur la caisse ne seront plus visibles : il ne restera plus qu'une enveloppe contenant la lettre originale (un **SDU**).

Tenez, un schéma illustrant l'encapsulation des **SDU** dans le modèle OSI :

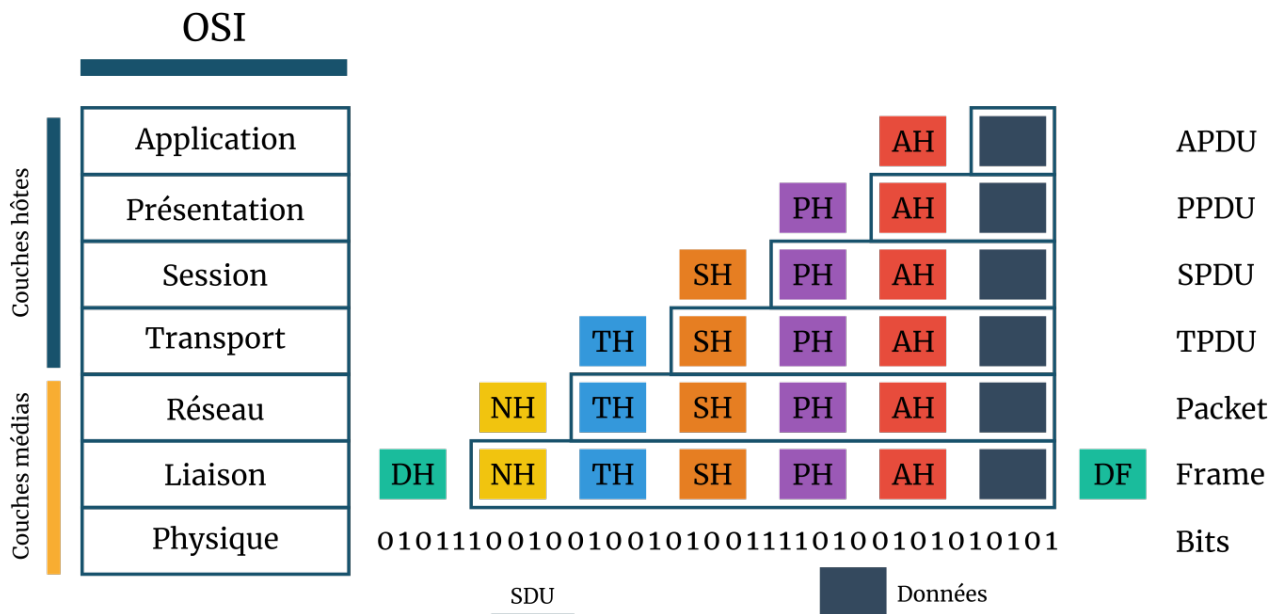


FIGURE II.2.9. – Encapsulation des SDU (CC BY)

zestedesavoir.com | Les réseaux de zéro



Dans le schéma ci-dessus, DF signifie *Data link Footer*. Le terme n'est pas exact, mais nous l'utilisons pour faciliter votre compréhension. Le vrai terme français qui équivaut au mot *trailer* est « remorque ». Une remorque est un genre de véhicule que l'on attèle à un autre véhicule ; la remorque est en quelque sorte la queue ou le *footer* du véhicule principal. Il est donc plus facile d'utiliser *footer* plutôt que *trailer*, le mot pied plutôt que remorque.



Tous les éléments encadrés en bleu forment un **SDU**, comme le stipule la légende.

Comme vous le voyez, au début nous n'avons que les données initiales, que l'on pourrait également appeler données d'application. La donnée initiale à ce stade est un **SDU**. Une fois dans la couche applicative, un en-tête AH (*ApplicationHeader* : « en-tête d'application ») est ajouté à cette donnée initiale. La donnée de la couche applicative est un APDU. La couche applicative transmet cela à la couche de présentation au-dessous. Cette donnée transmise est un **SDU**. Par l'encapsulation, cette couche ajoute un en-tête PH au **SDU** de la couche applicative. La couche de présentation envoie ce « nouveau » message à la couche de session et cette dernière encapsule son *header* avec le **SDU** obtenu de la couche présentation pour former son SPDU. Et ainsi de suite jusqu'à la couche liaison, qui a la particularité d'ajouter également un *trailer*. Finalement, toutes ces données sont converties en une série de *bits* et mises sur le média pour la transmission.



Une couche ne doit pas connaître (ne connaît pas) l'existence de l'en-tête ajouté par la couche au-dessus d'elle (la couche C + 1). En fait, cet en-tête, par l'encapsulation,

i

apparaît comme faisant partie intégrante de la donnée initiale. Par conséquent, la couche ignore qu'il s'agit d'un en-tête, mais elle le considère comme appartenant aux données à transmettre.

Vous pouvez également constater que toutes les informations ajoutées dans la couche supérieure se retrouvent dans la couche inférieure. Ainsi dans la couche réseau, par exemple, on retrouve la donnée initiale + l'en-tête d'application (AH) + PH + SH + TH. Toutes ces « informations » seront considérées par la couche réseau comme la donnée initiale. Dans cet exemple, la couche réseau ne s'occupe donc que de son propre en-tête.

?

Si, à chaque couche, l'en-tête est ajouté à la donnée initiale, ne serait-ce pas compromettre l'intégralité du message ?

Qui peut répondre à cela ? :D Très belle question, soit dit en passant. 🍊 Chaque couche ajoute à la donnée initiale un en-tête. De la sorte, tous les en-têtes sont réunis dans la couche de liaison. Lorsque ces informations seront converties en une suite de *bits*, le récepteur devrait recevoir des données erronées puisque la donnée initiale n'avait pas tous ces en-têtes, n'est-ce pas ? En principe. Mais le modèle OSI (ou le modèle TCP/IP) est assez intelligent. En effet, dans la procédure de réception, chaque en-tête est enlevé lorsque le message « grimpe » les couches, tel qu'illustré par le schéma ci-dessous. Cette « suppression » d'en-tête, c'est la **décapsulation** !

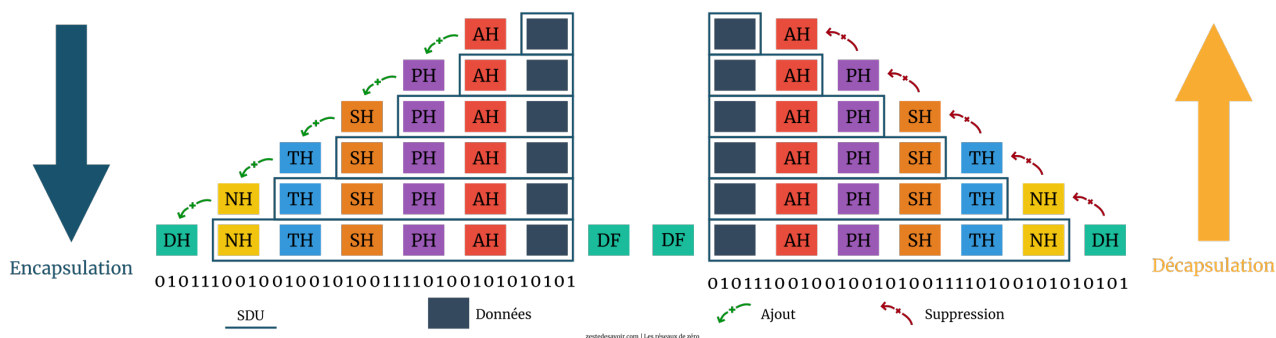


FIGURE II.2.10. – Représentation schématique de l'encapsulation et de la décapsulation (CC BY)

Comme vous le voyez sur le schéma, dans la procédure de réception, chaque couche supprime son en-tête correspondant **après l'avoir lu**. Par exemple, l'en-tête NH (réseau) est supprimé dans la couche réseau de l'hôte récepteur après que ce dernier l'a lu.

Maintenant que vous savez à quoi il sert, nous allons entrer dans les coulisses du modèle OSI par le haut. Pourquoi pas par le bas ? ~~Parce qu'il est plus facile de descendre des escaliers que de les monter.~~ Parce que nous estimons qu'il est plus intéressant de commencer par ce qui est plus proche de nous, à savoir les applications que nous utilisons.

Troisième partie

Veillez-vous identifier pour communiquer

III. Veuillez-vous identifier pour communiquer

Nous arrivons à un moment où l'identification et l'adressage deviennent des éléments clés pour pouvoir aller plus loin. C'est pourquoi nous allons y consacrer toute une partie!

III.1. Des adresses en folie!

Pour communiquer, il faut savoir à qui on veut s'adresser ! Lorsque nous avons parlé du commutateur (ou *switch*) dans le chapitre sur le matériel, nous avons évoqué des moyens d'identification au sein du réseau : l'adresse **IP** et l'adresse **MAC**. Il est temps de voir ce que c'est, et pourquoi on a besoin de ces 2 types d'adresses.

III.1.1. IP vs MAC

Il est temps de parler de l'identification et de la communication dans un réseau. Nous allons aborder 2 notions : il s'agit des adresses **IP** et des adresses **MAC**. Nous allons les aborder une par une, et comprendre pourquoi il y a des adresses **IP** et des adresses **MAC**.

III.1.1.1. Adresse IP : l'adresse relative au réseau

Dans le premier chapitre, nous avons vu un exemple simple de la transmission d'un livre entre humains. Mais, pour transmettre un livre à André, vous devez savoir où il habite.

Une adresse **IP** n'est «rien d'autre» que l'endroit où habite un ordinateur. Mais attention : cette adresse est *relative au réseau*. Une machine n'aura pas forcément la même adresse **IP** sur un réseau X et un réseau Y. Nous n'entrerons pas dans les détails pour le moment, l'adressage n'étant pas vraiment une base.

Les adresses **IP** sont le seul moyen d'identification des machines sur Internet. Mais il existe 2 versions du protocole Internet (la «manière» d'accéder à Internet en quelque sorte) : IPv4 et IPv6. Et chaque version utilise sa propre structure d'adresse **IP**.

Une «adresse IPv4» est constituée de 4 nombres correspondant à **4 octets** compris entre 0 et 255, séparés par des points. Exemple : 88.45.124.201. De nos jours, ce sont les plus connues. Les «adresses IPv6» sont encore plus complexes : elles sont représentées par une suite de 8 groupes de 2 octets représentés en hexadécimal (je vous avais prévenu que c'était complexe 🍌). Exemple (tiré de Wikipédia) : 1fff:0000:0a88:85a3:0000:0000:ac1f:8001.

Cette explication de ce qu'est une adresse **IP** est acceptable pour l'instant, mais vous verrez pourquoi une adresse **IP** est plus complexe que ça. En fait elle agit un peu comme un signe distinctif : si dans une rue toutes les maisons sont identiques, comment faites-vous pour reconnaître celle d'André ou de Pierre ? Dans notre exemple, c'est en se basant sur le numéro affiché devant la maison. Mais s'il existe plusieurs rues ? Plusieurs maisons peuvent avoir le même numéro sans être au même emplacement. Comment fait-on pour délimiter les rues ? On utilise pour cela un **masque de sous-réseau**, et l'adresse **IP** correspond au numéro de chacune des maisons.

i

Internet est une sorte de rue géante, comportant des croisements avec d'autres rues plus petites. Ces petites rues sont des sous-réseaux connectés à Internet, et chaque messager (chaque passerelle par défaut) aux carrefours possède une adresse **IP** spéciale relative au réseau Internet.

III.1.1.2. Adresses MAC : l'adresse relative à la carte réseau

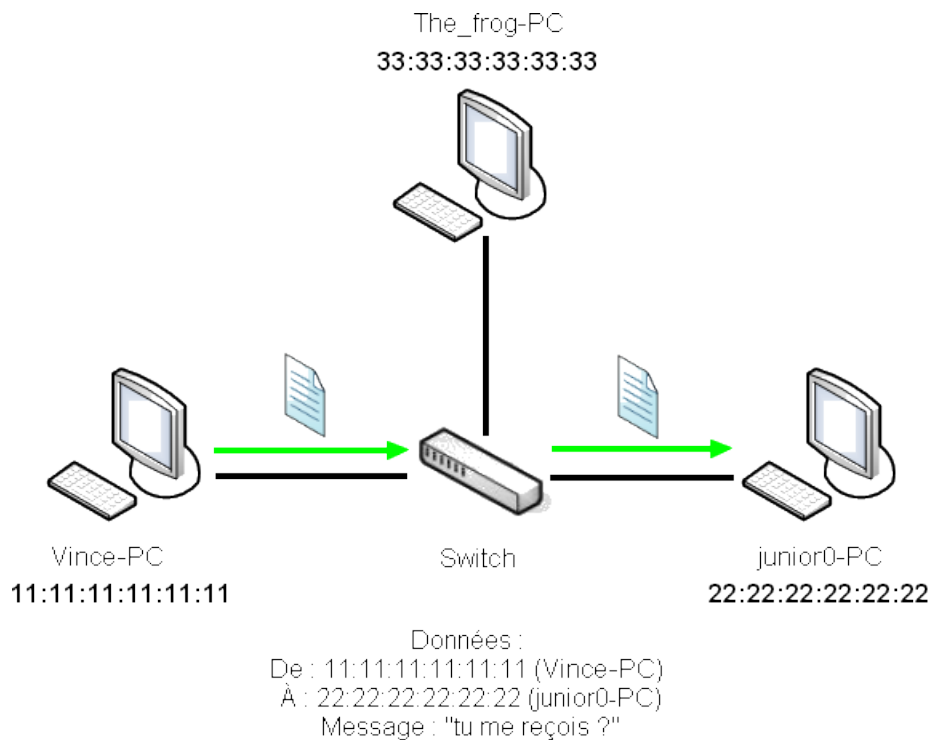
!

Précisons avant tout, le nom d'adresse **MAC** n'a rien à voir avec les ordinateurs Mac. Il vaut mieux prévenir, on ne sait jamais... 🍊

Comme dit brièvement lors du chapitre précédent, une adresse **MAC** est un identifiant unique attribué à chaque carte réseau. C'est une adresse **physique**. Concrètement, c'est un numéro d'identification composé de 12 chiffres hexadécimaux. Par convention, on place un symbole deux-points (:) tous les 2 chiffres. Une adresse **MAC** ressemble donc à cela : **01:23:45:67:89:AB**.

Imaginons un petit réseau de 3 ordinateurs connectés au même *switch*. Rappelez-vous, un *switch* est plus intelligent qu'un *hub*. Plutôt que d'envoyer ce qu'il reçoit par un port à tous les autres, il «filtre» les données renvoyées en se basant sur les adresses **MAC** des ordinateurs qui sont connectés. Prenons par exemple trois ordinateurs. Appelons-les Vince-PC, junior0-PC, et The_frog-PC (au cas où vous vous demanderiez pourquoi ces noms, ce sont les auteurs historiques du tuto 🍊). Si Vince-PC veut communiquer avec junior0-PC, il va envoyer au *switch* ce qu'il veut communiquer à junior0-PC. Le *switch*, ou commutateur, va regarder l'adresse **MAC** du destinataire et va lui envoyer ce qui lui est destiné sans l'envoyer aux autres machines (ici à The_frog-pc). En fait, le commutateur utilise une table de correspondance entre adresses **MAC** et ports pour savoir où envoyer les données.

Voici une illustration d'une communication basée sur les adresses **MAC** :



Les réseaux de zéro - zestedesavoir.com

FIGURE III.1.1. – Une trame contient au moins les adresses MAC du destinataire et de l'expéditeur



Mais pourquoi on n'utilise pas juste les adresses MAC?

Parce que dans un grand réseau, comme un WAN, ou même Internet, il n'y a pas d'élément central qui connaît l'emplacement du destinataire et qui peut renvoyer les données en conséquence. Par contre, le système d'adresses IP permet, grâce à un processus appelé **routing**, d'assurer que les données arrivent bien au destinataire. Le routage sera expliqué dès la prochaine partie.

III.1.1.3. En résumé...

La différence primordiale entre les adresses IP et les adresses MAC est que les adresses IP sont routables. Elles peuvent communiquer avec des machines au-delà d'un sous-réseau, contrairement aux adresses MAC. Le *switch*, au cœur du LAN, se base donc sur les adresses MAC pour assurer la communication entre plusieurs machines appartenant à un même sous-réseau. En revanche, les adresses IP sont utilisées pour faire communiquer des machines de sous-réseaux différents.

On espère que vous avez compris. 🍊

III.1.2. Masque de sous-réseau et passerelle

Afin de présenter ces notions, nous allons reprendre l'idée d'un «réseau» d'humains.

III.1.2.1. Les sous-réseaux et leurs masques

Considérons deux personnes, Jacques et Jean, et un gros réseau : leur ville. Nous allons établir des lois. Pour que deux personnes puissent se parler directement :

- Elles doivent parler la même langue;
- Elles doivent habiter dans la même rue;
- Chaque personne doit connaître l'adresse de l'autre (le numéro de la maison de l'autre).

Si Jacques habite la rue ClemStreet, et Jean aussi, alors ils peuvent facilement communiquer: ce n'est pas bien loin, ils vont marcher mais ils doivent, évidemment, parler la même langue. La rue est ici l'équivalent de ce qu'on appelle en informatique un **sous-réseau**, quant à la langue, c'est ce que l'on appelle un **protocole**. Si vous avez bien compris : une autre rue, par exemple juniorStreet (c'est le créateur du tuto qui a choisi ce nom 🍊), équivaut donc en informatique à... un autre sous-réseau!

?

Mais que vient faire un masque ici?

Ce serait très difficile d'expliquer directement cette notion, alors nous utiliser notre formule magique : analogie, magie! 🧙🏻‍♂️

Dans une adresse postale, il y a un numéro et un nom de rue. Par exemple : 17 rue des Coquelicots (au hasard 🍊). Un masque, c'est ce qui sépare le numéro du nom de la rue. Pour une adresse postale, ça se voit à l'œil nu (on sait reconnaître en un coup d'œil un numéro). Mais en réseau, c'est différent.

Prenons l'adresse IP 10.54.29.84 (au hasard, toujours). On ne peut pas, à première vue, reconnaître le numéro (de l'hôte) de la rue (le réseau, ou sous-réseau) : il n'y a que des chiffres! C'est pour ça qu'on a recours à un masque : c'est une suite de nombres qui dit que telle partie correspond au nom de la rue (au sous-réseau) et telle partie identifie l'hôte (le numéro de la maison). On verra dans les chapitres suivants comment se représente un masque. 🍊

Prenons un autre exemple : le téléphone (ce n'est pas pour rien qu'on a évoqué le réseau télécom avec le réseau Internet!).

Si vous souhaitez téléphoner, que faites-vous? C'est simple : vous prenez votre téléphone, vous tapez le numéro de votre correspondant, puis vous validez (en général, parce qu'il y a toujours des téléphones bizarres :p). Le numéro de votre correspondant peut, là encore, être assimilé à une adresse IP. 🍊

Cependant, si vous appelez à l'international, comment faire? Si votre ami habite le Cameroun par exemple, vous devez rentrer l'indicatif national de son pays. Dans notre cas, c'est 237 (vous rentrerez alors +237 sur les portables et 00237 sur les fixes généralement). Vous voyez le rapport avec les sous-réseaux? Un pays représente dans notre exemple un sous-réseau du réseau télécom mondial et l'indicatif de ce pays est équivalent au masque du sous-réseau.

On voit donc dans ces deux exemples que l'adresse IP (le numéro de la maison ou le numéro de téléphone) appartient à un sous-réseau.

En reprenant le parallèle que l'on vient de faire entre un réseau «humain» et un réseau informatique, et maintenant que l'on a tout le vocabulaire, vous devez être capable de transformer les trois lois précédentes en les appliquant à un réseau informatique...

La correction ? La voici :

👁 Contenu masqué n°1

?

Mais alors, comment faire pour que deux machines, appartenant à des sous-réseaux différents, communiquent ?

C'est là qu'intervient...

III.1.2.2. La passerelle

Celle-ci permet donc la communication entre deux sous-réseaux :

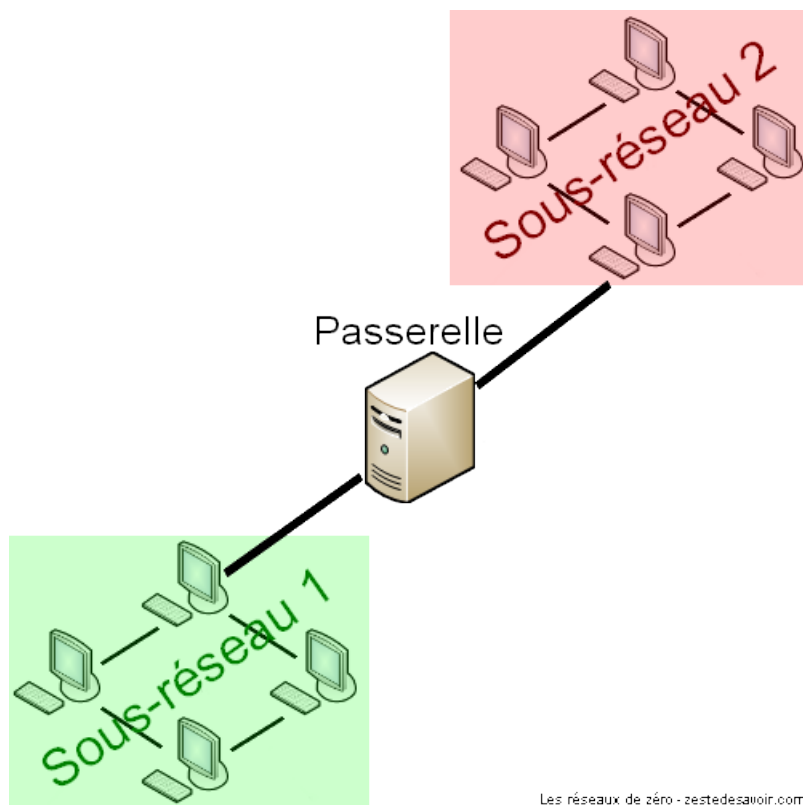


FIGURE III.1.2. – Une passerelle qui relie 2 sous-réseaux entre eux

Une passerelle est un autre ordinateur qui a plusieurs cartes réseau (en général, c'est un routeur). Cet ordinateur peut communiquer avec plusieurs sous-réseaux. On peut le comparer à une personne située à un carrefour, c'est-à-dire un croisement de plusieurs rues. La passerelle sert

ainsi de messager entre les habitants des différentes rues. Il faut un peu d'imagination pour comprendre...

i

On parle aussi de passerelle par défaut, de passerelle applicative ou de passerelle logique. Tous ces termes sont synonymes.

Un hôte communique avec la passerelle par défaut selon l'architecture **client-serveur**.

III.1.3. Le client et le serveur

Client et serveur, voici 2 mots que vous pouvez rencontrer dans la vie courante. Dans un café, par exemple. Un client est une personne qui demande quelque chose au serveur : le client demande un café au serveur, qui lui apporte. En informatique, le principe est le même : un client va demander quelque chose au serveur. Un exemple très simple : quand vous allez sur Zeste de Savoir, vous êtes un client qui demande au serveur du site une page. Dans la théorie, c'est aussi simple que ça.

Le mode de communication entre un client et un serveur est appelé **architecture client- serveur**.

Un autre exemple? Les serveurs IRC. Pour ceux qui ne connaissent pas, un serveur IRC est un serveur (eh oui 🍊) sur lequel des clients peuvent venir discuter («*chatter*») sur des salons. Un des clients ayant rejoint un salon peut envoyer un message au serveur en lui demandant de le transmettre aux autres, ce qu'il s'empresse de faire comme le montre cette animation :

Voir fichier serveursIRC.gif

FIGURE III.1.3. – Des clients connectés à un serveur IRC, sur le même salon, s'échangent des messages

i

Bien que cette architecture se retrouve dans beaucoup d'*applications* d'Internet (eh oui, il faut se remémorer le premier chapitre, dur), il existe un autre mode de communication : le pair-à-pair (P2P). Il s'avère très pratique dans certaines applications, comme le partage de fichiers notamment.

Vous devriez à ce stade comprendre l'identification dans un réseau. Mais n'allez pas vous imaginer que c'est si simple ! Maintenant, on va aller plus en profondeur...

III.2. Les masques de sous-réseaux : à la découverte du subnetting

Après avoir abordé la notion de masque de sous-réseaux, nous allons voir concrètement de quoi il s'agit. On va commencer à accélérer à partir de ce chapitre, alors soyez attentifs! 🍏



Ce chapitre sur les masques de sous-réseaux n'est valable que pour les adresses IPv4.

III.2.1. En bref

Un masque de sous-réseau, ça ressemble un peu à une adresse IP dans la forme, mais chaque octet ne peut prendre que certaines valeurs. Des exemples : 255.255.0.0, 255.255.255.0,... On les associe à des adresses IP et cela définit une **plage d'adresses** qui vont constituer un réseau. C'est donc le masque qui va définir avec qui on peut communiquer.

Prenons une adresse IP quelconque : 42.51.82.3. Associons à cette adresse un masque, par exemple 255.0.0.0. Ce masque va définir quelle partie de l'adresse IP **identifie le réseau** (cette partie est appelée **network ID**) et quelle partie **identifie l'hôte sur le réseau (host ID)**. C'est bien compris? Il vaudrait mieux, car nous allons maintenant voir comment cette définition du network ID et de l'host ID se fait. 🍏

III.2.2. L'importance des masques

Un masque de sous-réseau définit donc la plage d'adresses IP avec laquelle une carte réseau peut communiquer directement. Pour communiquer avec des adresses IP extérieures à cette plage, elle doit passer par une passerelle par défaut. Il est maintenant temps de voir la relation qui lie cette plage au masque.

III.2.2.1. Relation entre network ID et masques

Regardez bien cet exemple d'adresse IP et son masque de sous-réseau associé :

Adresse	129.51.3.5
Masque	255.255.0.0

III. Veuillez vous identifier pour communiquer

Les octets du masque ayant pour valeur 255 sont les mêmes que les octets de l'adresse IP définissant le network ID. De même, les octets du masque valant 0 correspondent aux octets de l'adresse IP définissant l'host ID. L'adresse IP ci-dessus est donc celle d'un **hôte 3.5 dans le réseau 129.51**. Cela est d'une importance capitale, et vous aurez l'occasion de vous en rendre compte quand nous verrons la personnalisation des masques. Avant d'introduire cette notion, voyons d'abord...

III.2.2.2. Des règles fondamentales à connaître absolument

Un masque de sous-réseau ne peut pas s'écrire n'importe comment. Voici quelques règles à connaître par cœur avant même d'aller plus loin :

On ne peut pas mélanger les zéros et les autres valeurs. En somme, tous les 255 doivent être à gauche et les zéros à droite. Pourquoi ? Parce que dans une adresse IP, c'est la partie gauche qui correspond à l'identité du réseau, et la partie droite qui correspond à l'identité de l'hôte. Ces exemples de masques sont donc **invalides** : 255.0.0.255, 255.255.0.255, 0.0.0.255,...

Un masque de sous-réseau ne peut pas avoir un octet qui vaut plus de 255, pour la bonne et simple raison qu'un octet ne peut prendre que 256 valeurs différentes, ici de 0 à 255. Par conséquent, un masque de sous-réseau ne peut pas prendre de valeur négative.

Ces règles sont simples, mais il faut absolument les savoir pour aller plus loin dans l'étude des masques de sous-réseau. Nous aurons l'occasion d'en voir d'autres par la suite, notamment lors de l'étude du **subnetting**.

III.2.3. Introduction au subnetting

Le **subnetting** est une technique qui consiste à diviser un réseau plus large en plusieurs sous-réseaux. Décomposons ce mot :

sub - net - ting sous - réseau - (suffixe d'action)

Il n'existe apparemment pas d'équivalent français. Si vous avez envie de dire « sous-réseautage », libre à vous, mais on risque de vous regarder bizarrement... 🍊 Vous l'aurez peut-être deviné, le **subnetting** est l'action de créer des sous-réseaux. Et pas n'importe comment : en **personnalisant les masques**.

Par exemple, admettons un réseau de 1000 ordinateurs. La gestion d'un tel réseau ne doit pas être évidente. Grâce au **subnetting**, on peut par exemple **diviser** ce grand réseau en 10 réseaux de 100 ordinateurs chacun (en gros). Et cela procure des avantages, voyez par vous-même !

III.2.3.1. Délégation de l'administration

Le **subnetting** permettant de diviser un grand réseau en plusieurs réseaux plus petits, il permet de décentraliser l'administration, et éventuellement de déléguer la gestion de chaque sous-réseau à une personne différente. Dans une entreprise possédant un réseau de 1000 machines, sa gestion sera simplifiée.

III.2.3.2. La réduction du trafic

Si 2 ordinateurs se trouvant dans un même sous-réseau communiquent, ils n'exploiteront que la bande passante allouée à leur sous-réseau, et non celle du réseau entier. Considérons une entreprise possédant un réseau de 500 machines. Il est divisé en 25 sous-réseaux de 20 machines. Ainsi, les machines appartenant à un même sous-réseau communiquant entre elles n'utilisent que la bande passante qui est allouée à leur sous-réseau, ce qui permet de ne pas réduire le débit des autres. Cela se remarque notamment lors du *broadcast* de données : elles ne sont transmises qu'aux ordinateurs du sous-réseau, et pas aux autres qui n'en ont probablement rien à faire. Si vous avez oublié de quoi il s'agit, c'est que vous n'avez pas fait attention au passage sur les envois de données du chapitre précédent ! Le *broadcast* est utilisé notamment par le protocole ARP qui permet d'associer adresses **MAC** et adresses **IP**. Nous aurons peut-être l'occasion de traiter ce sujet en annexes.

III.2.3.3. La facilité du diagnostic

Si par exemple un ordinateur consomme une quantité de bande passante inhabituelle, il est beaucoup plus aisé d'analyser son comportement pour régler le problème lorsqu'il se trouve dans un petit sous-réseau que lorsqu'il se trouve dans le même réseau que 1000 autres machines. C'est encore un avantage.

III.2.3.4. L'économie d'adresses

Prenons par exemple une adresse **IP** : 200.10.0.5. Le masque de sous-réseau par défaut est 255.255.255.0. Dans ce cas, on peut avoir jusqu'à 254 terminaux (clients) dans ce même réseau, donc 254 adresses **IP**. Ce qui veut dire que si vous avez un réseau de 10 ordinateurs, vous avez quand même 254 adresses **IP** disponibles. Mais comme vous ne les utilisez pas, vous les **gaspillez**. Toutefois, le *subnetting* ne nous permet pas d'économiser comme on le souhaite.



Vous avez 254 adresses **IP** disponibles uniquement lorsque vous utilisez un masque de sous-réseau par défaut.



Et ça sert à quoi d'économiser des adresses **IP** ? Ça ne va pas coûter plus cher de laisser 200 adresses **IP** vacantes, que d'en laisser 2...

Dans un réseau privé, certes. Mais cela peut être utile pour des raisons de sécurité, entre autres. Nous ne pouvons pas encore voir réellement l'intérêt, vous vous en rendrez compte en temps voulu. Sachez toutefois que sur Internet, les adresses **IP** publiques s'achètent. Et ce n'est pas le même prix d'acheter 2 adresses que 200 (ça doit faire environ 100 fois plus 🍌).



Donc le *subnetting* permet de diviser un réseau en plusieurs sous-réseaux, ça a plein d'avantages, mais ça se met en place comment, concrètement ?

III. Veuillez vous identifier pour communiquer

C'est le sujet du prochain chapitre ! Hé oui, « introduction au *subnetting* », ça veut dire « définition et du blabla » ! Vous croyiez quoi ? Que vous alliez « subnetter » sans savoir à quoi ça sert, juste pour dire « je suis trop fort, j'ai subnetté mon *home network* » ? 🤖

Avant de subnetter, voici des informations qui vous seront probablement utiles, notamment si vous débutez en tant qu'administrateur réseau.

III.2.4. Analyse des contraintes et plan d'adressage

Vous vous en doutez peut-être, les administrateurs réseaux passent beaucoup plus de temps à analyser qu'à implémenter. C'est d'ailleurs le rôle principal d'un administrateur réseau : apporter son expertise dans l'analyse et le design d'une infrastructure réseau. Quant à l'implémentation, c'est relativement simple une fois l'analyse terminée. En fait, c'est comme en programmation. Il y a le chef de projet qui analyse les contraintes et les demandes des clients, écrit éventuellement un cahier des charges, et le remet aux développeurs qui se serviront des contraintes de ce dernier pour créer une application. En réseau, c'est le même principe : vous, l'administrateur, allez réfléchir sur les contraintes du réseau, et vous allez proposer une solution en tenant compte de plusieurs critères (le prix, la facilité de mise en place, l'évolution de l'infrastructure, etc.).

III.2.4.1. Analyse des contraintes

Avant de « subnetter » un réseau, il faut donc faire une minutieuse analyse. Nous allons vous donner quelques pistes.

III.2.4.2. Le prix

Subnetter un réseau, c'est le subdiviser en plusieurs sous-réseaux. Ceci dit, il en résulte explicitement que l'achat de matériel additionnel est obligatoire, en effet il faudra un routeur pour que les sous-réseaux obtenus puissent communiquer. Qui dit nouveau matériel dit... câblage ;) Bref, il faut prendre en compte cette contrainte financière. Un client (ou votre patron) peut vous spécifier un budget pour l'infrastructure à mettre en place et il faudra trouver un compromis pour allier « meilleur prix » et « meilleure solution », ce n'est pas toujours évident, les boss sont trop exigeants. Parfois. 🤖

III.2.4.3. L'évolution du réseau

Un bon administrateur n'est pas celui qui offre une solution idéale à court terme. Les réseaux sont un domaine de l'informatique qui évolue très vite. Il ne faut jamais penser à une solution qui ne serait fonctionnelle que pendant 1 an. Il est préférable se poser la question : « dans 2-3 ans, à quoi ressemblera mon réseau ? Sera-t-il facile d'évoluer vers une nouvelle infrastructure si j'utilise telle infrastructure ? ».

III.2.4.4. Le nombre d'adresses IP

Il faut déterminer le nombre d'adresses IP dont on aura besoin. Les administrateurs débutants ont tendance à choisir pile-poil un sous-réseau qui leur offre exactement le nombre d'adresses IP dont ils ont besoin (c'est rare mais c'est néanmoins possible). Or cette pratique est une erreur, ou du moins, elle est fortement déconseillée. Si nous nous restreignons à un sous-réseau qui nous permet d'avoir 17 adresses IP par exemple, et que dans un futur proche nous ajouterons 400 autres ordinateurs... Vous rendez-vous compte de l'ornière dans laquelle nous nous trouverons ? Il faudra re-subnetter correctement et redéfinir les plages, et c'est... ennuyeux. 🍊

i

Il est recommandé de choisir un masque en se basant sur le maximum d'adresses IP qu'un réseau donné pourrait avoir, et non le minimum ou l'actuel. Par exemple, si vous avez besoin d'un sous-réseau de 10 adresses IP et qu'il peut y avoir agrandissement de réseau, que vous êtes sûrs que ça ne dépassera pas un maximum de 40 ordinateurs, il serait alors judicieux de commencer par choisir un masque qui vous donne d'ores et déjà 40 adresses IP. Cela peut être considéré comme du gâchis d'adresses mais c'est néanmoins pratique pour l'évolution.



III.2.4.5. L'organisation

L'une des choses les plus importantes, hormis les contraintes évoquées ci-dessus, est l'organisation du plan d'adressage.

i

Un plan d'adressage est un plan résultant d'une analyse de contrainte, qui servira de modèle pour gérer l'adressage / l'assignation des adresses dans un réseau donné.

« Comment allez-vous organiser vos sous-réseaux ? » Telle est la question qu'il faut se poser, niveau organisation. Plusieurs méthodes d'organisation sont courantes.

III.2.4.5.1. L'organisation par bâtiment

Certaines entreprises ont une organisation par architecture (physique). Par exemple, elles peuvent avoir le bâtiment A, qui regroupe le *staff* se chargeant du service après-vente. Elles peuvent également avoir le bâtiment C, qui regroupe le *staff* se chargeant du service des finances, etc. Vous pouvez par conséquent être amené à subnetter et organiser les sous-réseaux par bâtiment : créer un sous-réseau pour le bâtiment A, un autre pour le bâtiment B, etc. Donc cette organisation consiste à créer autant de sous-réseaux qu'il y a de bâtiments. 🍊 Elle a ses avantages, car elle offre une facilité de diagnostic grâce au repérage physique. Par exemple, on pourrait facilement dire que l'ordinateur D02 qui a des difficultés à communiquer est localisé dans le bâtiment D. C'est donc une méthode d'isolation utile en cas de diagnostic. 🍊

i

Dans ce genre d'organisation, les hôtes sont souvent nommés par un motif : nom du bâtiment + numéro d'hôte. Par exemple l'hôte 2 dans le bâtiment H serait nommé H02 (non, ce n'est pas une molécule). 🍊

III.2.4.5.2. L'organisation par fonctions

Cette organisation est différente de la précédente. On peut avoir un bâtiment B qui regroupe des employés du service de support informatique. Dans ce bâtiment, il y aura par exemple un chef de projet, des réceptionnistes et des techniciens. Mais il se peut que l'entreprise ait aussi des techniciens en électronique dans le bâtiment C, ou un chef de projet dans le bâtiment D qui s'occupe de la recherche. Dans ce genre de cas, vous pouvez alors subnetter par fonctions. C'est-à-dire, créer un sous-réseau qui n'hébergera **que** les ordinateurs des chefs de projets (tous services confondus), un sous-réseau qui n'hébergera que les secrétaires (tous services confondus), etc.

?

Ouh là, c'est pas de la ségrégation ça? 🍊

Que nenni. :D Cette organisation peut être très pratique. Imaginez que vous ayez plusieurs techniciens en informatique industrielle, qui communiquent constamment avec un serveur d'applications dans leur domaine. Les logiciels hébergés par le serveur sont lourds, et lorsque tous les techniciens travaillent à un rythme fou et multiplient les requêtes vers le serveur, cela ralentit le réseau. Avec une organisation par fonctions, vous aurez un sous-réseau alloué aux techniciens en informatique industrielle qui implémentera un débit assez élevé, uniquement pour assurer cette fonction. C'est pratique, on peut alors allouer une bande passante précise par sous-réseau en fonction des contraintes. Car, avouons-le, ça sert à rien d'allouer 512 Mo/s de débit aux secrétaires. 🍊 (Ah, on nous dit dans l'oreillette qu'on va se faire taper par des secrétaires fâchées. On finit le chapitre et on met les voiles! 🍊)

III.2.4.5.3. L'organisation par architecture

Le titre est assez évocateur, donc nous allons faire court (aussi parce que nous manquons d'inspiration :-°). Cette organisation consiste à subnetter avec une organisation par architecture. Dans la partie I du cours, souvenez-vous, nous avons parlé de la topologie logique « Token Ring ». Grâce à une organisation par architecture, vous pouvez créer un sous-réseau spécial Token Ring, un autre sous-réseau spécial Ethernet, et un autre spécial Wi-Fi, etc. 🍊

Voilà, nous avons fait le tour des techniques d'organisation. Cette phase d'analyse ne vous servira à rien en tant qu'étudiant, cependant quand vous entrerez dans le monde actif en réseau, elle vous sera d'une grande utilité. Et même en stage, ça peut servir... à impressionner le maître de stage! 🍊 (Assurez-vous quand même auparavant que le «m'as-tu vu» ne l'agace pas!)

III. Veuillez vous identifier pour communiquer

Le prochain chapitre sera donc dédié à la personnalisation des masques de sous-réseau, ce qui permet de faire du *subnetting*. Et par conséquent, de restreindre la commande à distance de la machine à café aux autres. 🍊

III.3. Le subnetting en pratique

Maintenant que vous savez ce qu'est le *subnetting*, nous allons voir comment cela se fait. Faites chauffer vos méninges, cela demande du calcul ! ... Hé ne partez pas ! C'est facile, vous allez voir ! Restez, voyons! 🍊



Pour pouvoir suivre ce chapitre, le binaire doit vous être familier. Si tel n'est pas le cas, [consultez cette annexe](#) avant de poursuivre votre lecture.

III.3.1. Comment?

Considérant que vous maîtrisez les conversions d'adresses IP du binaire au système décimal et vice versa, que vous connaissez les puissances de deux et que les yeux fermés vous pouvez dire que $2^3 = 8$, que vous savez ce qu'est le *subnetting*, et que vous comprenez parfaitement cette notion d'économie d'adresses, nous allons voir **comment** « **subnetter** », comment avoir un masque de sous-réseau personnalisé.



Cette section n'est que théorique, elle consiste simplement à vous dire comment on fait, et dans la suite de ce chapitre nous allons subnetter ensemble, pas à pas.

III.3.1.1. Le comment du pourquoi

Pour personnaliser les masques de sous-réseau, il faut **emprunter** des *bits* de la partie host (client) de notre adresse IP. Revoyons d'abord en douceur ce que sont le host ID et le network ID. D'abord, host ID signifie *identité de l'hôte* et network ID signifie *identité du réseau*. La règle de base à retenir est : plus on « monte » dans les masques (c'est-à-dire qu'on passe de 255.0.0.0 à 255.255.0.0, de 255.255.0.0 à 255.255.255.0,...), plus le network ID devient grand : il gagne un octet de plus à chaque fois, et le host ID en perd un.

Faire du *subnetting*, c'est subdiviser un réseau en plusieurs sous-réseaux, diminuer le nombre d'adresses IP par sous-réseau.

Avec le masque 255.0.0.0, le premier octet correspond à l'identité du réseau. Avec 255.255.0.0, ce sont les deux premiers octets, et avec 255.255.255.0, les 3 premiers. C'est l'inverse pour l'identité de l'hôte :

Masque de sous-réseau	Adresse IP
-----------------------	------------

III. Veuillez vous identifier pour communiquer

255.0.0.0	75.10.2.4
255.255.0.0	135.5.0.7
255.255.255.0	220.42.3.6

En **gras** : network ID, en *italique* : host ID

Donc, pour subnetter un réseau en plusieurs sous-réseaux, on se sert des *bits* disponibles du masque de sous-réseau, c'est-à-dire ceux qui valent 0.

Il est possible de procéder de plusieurs manières pour subnetter :

- En partant du nombre de sous-réseaux désirés;
- En partant du nombre d'adresses **IP** désirées par sous-réseau;
- En combinant les deux, c'est-à-dire en prenant en compte le nombre de sous-réseaux désirés et le nombre d'adresses **IP** désirées par sous-réseau.

Nous ne verrons pas tout de suite cette dernière possibilité, nous allons voir seulement les 2 premières. Ces méthodes sont valables pour toutes les adresses, néanmoins nous observerons quelques particularités relatives à l'ancienne classe C qui existe toujours.

III.3.2. À partir du nombre de sous-réseaux désirés

Avant de prendre un exemple concret, vous devez connaître quelques généralités. Tout d'abord, quelle que soit la classe d'adresses dans laquelle vous allez travailler, la formule est la même :

$$S = 2^n$$

Dans cette formule, S est le nombre de sous-réseaux désirés. À partir de S , vous devez déterminer n , qui est un nombre entier positif, et qui correspond au nombre de *bits* devant être **masqués**.



Masqués? Comme au bal?

Pasvraiment... 🍊 Masquer signifie, en gros, le mettre à 1 pour les besoins du *subnetting*. Comme vous le savez normalement, le *subnetting* consiste en l'emprunt des *bits* de la partie hôte pour créer des sous-réseaux. Ce processus d'emprunt de *bits* est appelé **masquage**. Masquer un *bit*, c'est donc l'emprunter, l'allumer.

En fait, S ne pourra pas toujours être égal au nombre de sous-réseaux désirés, nous verrons dans les exemples comment nous allons faire dans ce cas très courant.



Autrefois, dans ce cours, nous montrions une formule légèrement différente : $S = 2^n - 1$

1. Ce « -1 » venait d'une convention selon laquelle l'octet de l'adresse **IP** définissant le sous-réseau ne pouvait être supérieur ou égal à l'octet modifié du masque de sous-réseau personnalisé. Par exemple, si on avait un réseau 198.15.2.0 et qu'on applique aux hôtes un masque 255.255.255.192, on ne pouvait pas avoir de sous-réseau ayant pour identité

III. Veuillez vous identifier pour communiquer



198.15.2.192. Cette pratique étant dépassée, nous ne la prenons plus en compte dans ce cours. Sachez que vous pourriez encore tomber sur de vieux routeurs qui vont vous casser les pieds avec ça.

Au vu de vos têtes sceptiques (enfin, on ne voit pas mais on devine 🍊), nous allons tout de suite faire un exemple, parce que la théorie, ça ne semble pas digeste.

III.3.2.1. Exemple de subnetting

Passons tout de suite au plus intéressant. Considérons le réseau **40.0.0.0**. Nous voulons le diviser en 20 sous-réseaux. Déterminons le nombre de *bits* à masquer pour obtenir un masque de sous-réseau personnalisé, qui devra être appliqué à tous les hôtes.



...



Ah oui, on ne peut pas obtenir exactement 20 avec la formule 2^n ! Dans ce cas, nous allons prendre une valeur un peu supérieure pour avoir au moins 20 sous-réseaux. Allons-y doucement :

- $2^4 = 16$
- $2^5 = 32$
- On a suffisamment de réseaux en masquant 5 *bits*, on arrête là.



Quand vous avez trop d'adresses **IP** par sous-réseau, vous devriez empêcher l'assignation des adresses inutilisées. C'est une question de sécurité, pour empêcher qu'un intrus puisse s'octroyer une adresse **IP** libre. Nous n'entrerons pas dans les détails ici.

On ne peut pas mélanger les 1 et les 0, **tous** les 1 doivent être à gauche, et les 0 à droite. Cela veut dire le masquage se fait de la gauche vers la droite.

Nous allons donc masquer 5 *bits* de cette manière. Nous avons les puissances suivantes (les 8 premières puissances de 2) :

- $2^7 = 128$
- $2^6 = 64$
- $2^5 = 32$
- $2^4 = 16$
- $2^3 = 8$
- $2^2 = 4$
- $2^1 = 2$
- $2^0 = 1$

Voilà donc les 8 premières puissances de deux, par ordre décroissant. Nous allons masquer les 5 *bits* qu'il nous faut pour obtenir 20. Donc, nous allons **additionner** les valeurs des 5 premières puissances ci-dessus. Cela nous donne donc : $2^7 + 2^6 + 2^5 + 2^4 + 2^3 = 248$.

III. Veuillez vous identifier pour communiquer

Nous avons masqué 5 *bits* du 2ème octet de notre masque de sous-réseau. Schématiquement, ça nous donne ceci :

255	248	0	0
ssssssss	sssshhhh	hhhhhhhh	hhhhhhhh

s = subnet ; h = host

La valeur de notre **nouveau masque de sous-réseau** est à présent **255.248.0.0**. Si vraiment vous n'avez pas compris comment le 248 a été obtenu :

👁️ Contenu masqué n°2

Et voilà, soyez très heureux, nous avons réussi à personnaliser un masque de sous-réseau! 🍊

Maintenant il faudrait définir les limites de chaque sous-réseau, sinon ça ne va pas être très utile. Dans le cas présent, nous en avons 31, on va vous donner les 5 premières plages et la dernière, vous allez faire le reste vous-mêmes, il faut bien que vous fassiez quelque chose! 🍊

?

Comment calculer les plages?

C'est relativement simple. Pour calculer les plages, il faut retrancher le nombre calculé du nouveau masque de sous-réseau à 256. Ce nombre est 248, $256 - 248 = 8$. Donc nos sous-réseaux seront séparés par un intervalle de 8.

Concrètement, reprenons le réseau que nous avons choisi à la base : 40.0.0.0. Le premier sous-réseau est 40.0.0.0, le deuxième est 40.8.0.0, le troisième 40.16.0.0, le quatrième 40.24.0.0, etc. 🍊

Pour calculer les plages, il faut savoir que la dernière adresse d'un sous-réseau donné est toujours égale à l'adresse de l'identité du prochain sous-réseau moins 1. 🍊 Un exemple concret? Dans notre cas, notre premier sous-réseau est 40.0.0.0. La première adresse **IP adressable** (pouvant être donnée à un hôte) est donc 40.0.0.1 et la dernière... 40.7.255.254. o_O

!

Techniquement, la dernière adresse dans un réseau est réservée pour la diffusion dite de *broadcast*. Cela sert à envoyer la même chose à tous les hôtes du réseau. Ici, l'adresse 40.7.255.255 est réservée pour le *broadcast* et n'est pas donc pas assignable à un hôte. Nous y reviendrons un peu plus tard.

Nous avons donc :

Ordinal	Adresse du sous-réseau	Première adresse IP d'hôte	Dernière adresse IP d'hôte
1er	40.0.0.0	40.0.0.1	40.7.255.254

III. Veuillez vous identifier pour communiquer

2ème	40.8.0.0	40.8.0.14	40.15.255.254
3ème	40.16.0.0	40.16.0.1	40.23.255.254
4ème	40.24.0.0	40.24.0.1	40.31.255.254
5ème	40.32.0.0	40.32.0.1	40.39.255.254
...
Dernier	40.240.0.0	40.240.0.1	40.247.255.254

i

Remarquez que le 2ème octet de la dernière adresse IP du dernier sous-réseau est inférieure à la valeur du 2ème octet du masque personnalisé (248). Nous avons vu cela dans la partie théorique, mais vous aviez l'air de ne pas comprendre (c'est pas évident à expliquer).



Vous êtes capables de faire le reste maintenant. 🍊

Si vous pouvez encore suivre, nous allons voir comment subnetter à partir du nombre d'adresses IP d'hôtes désiré. N'hésitez pas à faire une pause si vous pensez en avoir déjà beaucoup fait pour le moment.

III.3.3. À partir du nombre d'adresses d'hôtes désirées

Parés à affronter cette méthode ? Dans ce cas, voici une bonne et une mauvaise nouvelle. On commence par la mauvaise ? La formule et la méthode changent ! La bonne nouvelle ? Elles ne changent que très peu, et vous allez comprendre pourquoi !

III.3.3.1. Explications sur l'adresse de *broadcast* et l'identité du réseau

La nouvelle formule est :

$$S = 2^n - 2$$

Cette fois, S correspond au nombre d'hôtes désiré par sous-réseau. La raison du changement dans la formule est simple : on retranche une première unité pour l'identité du réseau car elle n'est pas assignable. Si l'adresse 40.16.0.0 identifie un réseau, elle ne peut pas identifier un hôte ! Une autre unité est retranchée car on ne peut pas non plus assigner l'adresse de *broadcast*.

i

Il est théoriquement possible de supprimer le *broadcast* dans un réseau, mais dans des cas très particuliers uniquement, que nous ne verrons pas ici. Il est indispensable dans la plupart des cas : par exemple, les requêtes ARP permettant d'établir des correspondances entre adresses IP et MAC sont envoyées en *broadcast* ! Si vous décidiez de la supprimer et d'utiliser cette adresse pour un hôte, la formule deviendrait $2^n - 1$.

III. Veuillez vous identifier pour communiquer

Ces explications sont facilement vérifiables lorsqu'on détermine les plages d'un réseau subnetté dans l'ancienne classe C, vous aurez l'occasion de le voir. Pour le moment, voyons une application de cette méthode avec un exemple.

III.3.3.2. Un autre exemple de subnetting

Prenons le réseau 158.37.0.0. Commençons par décider du nombre d'adresses IP que l'on souhaite avoir par sous-réseau.



Si vous choisissez un nombre inférieur à 255, vous vous retrouverez avec un masque sous la forme 255.255.255.xxx. Ce n'est pas interdit, ça reste du *subnetting*, mais ne soyez pas surpris si le dernier octet du masque est également modifié.

Considérons que nous voulons 1800 hôtes par sous-réseau. Déterminons n :

$$- 2^{10} - 2 = 1022$$

$$- 2^{11} - 2 = 2046$$

n vaut donc 11. C'est là que ça change : comme nous voulons un nombre précis d'adresses par sous-réseaux, 11 *bits* doivent être **libres pour les hôtes** ! Ce qui signifie que 11 *bits* doivent valoir 0. Il y a 32 *bits* par masque, pour connaître le nombre de *bits* devant valoir 1, on fait $32 - 11 = 21$. Notre nouveau masque doit donc comporter 21 *bits* allumés, écrivons-le en binaire :

```
11111111 . 11111111 . 11111000 . 00000000
```

Ce qui nous donne en décimal **255.255.248.0**. L'intervalle entre chaque sous-réseau est de $256 - 248 = 8$.

Dressons maintenant le tableau des plages :

Ordinal	Adresse du sous-réseau	Première adresse IP d'hôte	Dernière adresse IP d'hôte
1er	158.37.0.0	158.37.0.1	158.37.7.254
2ème	158.37.8.0	158.37.8.1	158.37.15.254
3ème	158.37.16.0	158.37.16.1	158.37.23.254
...
Dernier	158.37.240.0	158.37.240.1	158.37.247.254

Voilà donc un certain nombre de sous-réseaux avec 2046 adresses d'hôtes dans chaque. On n'en voulait que 1800, mais ce n'était pas possible de les avoir précisément, donc on a pris la valeur possible immédiatement supérieure.

Faisons maintenant un autre exemple, mais cette fois, il doit y avoir moins de 254 hôtes par sous-réseau. La méthode reste la même, mais nous allons voir quelques particularités, dont une qui permet de vérifier facilement la formule de départ.

III.3.3.3. Exemple de subnetting avec moins de 254 hôtes par sous-réseau

Procédons de la même manière. Dans le réseau 203.68.5.0, nous voulons 14 hôtes par sous-réseau. $2^4 - 2 = 14$! Super, on tombe juste sur 14 ! (Bon d'accord, c'était fait exprès pour éviter de chercher... 🍊)

Attention tout de même quand ça tombe juste comme ça... Si par malheur vous deviez rajouter un hôte dans un sous-réseau, vous seriez ~~dans la m~~ bien embêté car vous devrez reconfigurer **toutes les machines du réseau!**

On a donc $n = 4$, il nous faut 4 *bits* valant zéro. Comme ici on ne va modifier que le dernier octet, on peut faire directement $8 - 4 = 4$ pour connaître sa nouvelle valeur. $11110000_{(2)} = 240_{(10)}$, notre nouveau masque est donc **255.255.255.240**. L'intervalle est de $256 - 240 = 16$, on détermine les plages :

Ordinal	Adresse du sous-réseau	Première adresse IP d'hôte	Dernière adresse IP d'hôte	Adresse de <i>broadcast</i>
1er	203.68.5.0	203.68.5.1	203.68.5.14	203.68.5.15
2ème	203.68.5.16	203.68.5.17	203.68.5.30	203.68.5.31
3ème	203.68.5.32	203.68.5.33	203.68.5.46	203.68.5.47
...
Dernier	203.68.5.224	203.68.5.225	203.68.5.238	203.68.5.239

Vous remarquez probablement une différence : la dernière adresse IP d'hôte de chaque sous-réseau ne se termine pas par 254 ! De plus, vous voyez bien maintenant l'intérêt du -2 de la formule : l'adresse du réseau et celle de *broadcast* sont bien visibles ici.

Remarquez que le masque de sous-réseau ne peut être 255.255.255.255. En effet, dans ce cas, il n'y a que l'adresse même du sous-réseau dans chaque sous-réseau, donc aucune adresse disponible pour les hôtes ! D'ailleurs, si on prend comme masque 255.255.255.254, il n'y a qu'une adresse disponible par sous-réseau, on est donc obligé de supprimer le *broadcast* (ce qui n'est pas grave vu qu'il n'y a qu'un hôte).

Allez, vous avez bien travaillé, vous avez droit... à la suite ! 🍊

III.3.4. La notation du masque

Cette sous-partie ne comportera rien de fameux, nous allons juste vous fournir quelques explications sur les éventuelles notations que vous rencontrerez probablement dans le monde du réseau.

III.3.4.1. La notation « classique »

Cette notation dite « classique » est la notation « normale » d'une adresse IP. C'est en fait une notation qui couple l'adresse IP et son masque de sous-réseau associé. Par exemple, vous pourrez rencontrer une expression telle que **192.168.1.45/255.255.255.0**. C'est assez évident à comprendre, n'est-ce pas ? Cela veut simplement dire qu'à l'adresse IP 192.168.1.45 est attribué un masque 255.255.255.0. C'est une notation que nous pourrions qualifier de « obsolète » car elle a laissé sa place à...

III.3.4.2. La notation avec un slash (/)

Cette notation suit le même modèle que la notation classique. C'est-à-dire que c'est un couplage de l'adresse IP d'un hôte à son masque de sous-réseau. Mais le point particulier ici, c'est qu'au lieu de donner l'expression « brute » du masque de sous-réseau dans la notation, on se contente de spécifier le nombre de *bits* masqués pour obtenir ce masque. La notation précédente en notation avec un slash devient **192.168.1.45/24**. Cela veut dire que l'adresse IP 192.168.1.45 est associée à un masque ayant 24 *bits* de masqués.

La notation avec un slash semble devenir la plus courante et la plus utilisée aujourd'hui notamment avec le succès du CIDR (*Classless Inter Domain Routing*) que nous allons aborder très bientôt. En fait, la notation avec un slash n'est rien d'autre que ce qu'on appelle officiellement la **notation CIDR**. 🍊

i

Concernant les expressions « notation classique » et « expression brute », ce sont des expressions propres au tutoriel, n'allez pas croire qu'elles sont conventionnées. 🍊

Voilà, vous savez tout sur les notations que vous pouvez rencontrer. Il va sans dire que vous préférerez sûrement utiliser la notation avec un slash. C'est plus pratique : on peut se contenter d'écrire 130.14.56.3/16 par exemple au lieu de 130.14.56.3/255.255.0.0. 🍊

Reposez-vous avant de passer au chapitre suivant. Car vous allez encore faire des calculs, d'un autre genre cette fois! 🍊

III.4. L'adressage par classes (obsolète)

Il y a fort fort longtemps, dans l'antiquité l'adressage se faisait **par classes**. En gros, pour chaque adresse IP, un masque de sous-réseau était assigné par défaut en fonction de sa classe. Cela ne se fait plus depuis bien des années, néanmoins, il peut être intéressant de voir ce que c'était pour mieux comprendre certaines notions. Nous allons donc voir quelque chose d'**obsolète**.

?

Ça sert à quoi de voir ça si ça date de Malthusalem? 🍊

Retournez voir votre professeur d'histoire que vous aviez au lycée ou au collège et posez-lui la question. 🍊 Vous voyez où on veut en venir? Savoir le passé permet de comprendre le présent et d'appréhender l'avenir ! Allez hop, on y va ! (Le chapitre d'après portera sur l'adressage utilisé actuellement, ne vous inquiétez pas.)

III.4.1. C'est quoi une classe?

Au début de la deuxième partie, nous avons vu rapidement la notion d'adresse IP, sans rentrer dans les détails. Nous l'avons décrite comme un numéro d'une maison faisant partie d'une rue, ou encore comme un numéro de téléphone. Cette définition n'est pas très précise mais c'était le meilleur moyen de vous faire comprendre le principe. Dans ce chapitre, nous allons pousser notre investigation un peu plus loin à propos des adresses IPv4.



Hum, avant de commencer, pourriez-vous me montrer quelle est mon adresse IP?

Pour voir votre adresse IP sous Windows, utilisez le raccourci clavier **Windows** + **R**, tapez `cmd` et validez. Sous Linux ou Mac OS, ouvrez votre terminal. Vous allez voir une belle boîte noire style DOS apparaître :

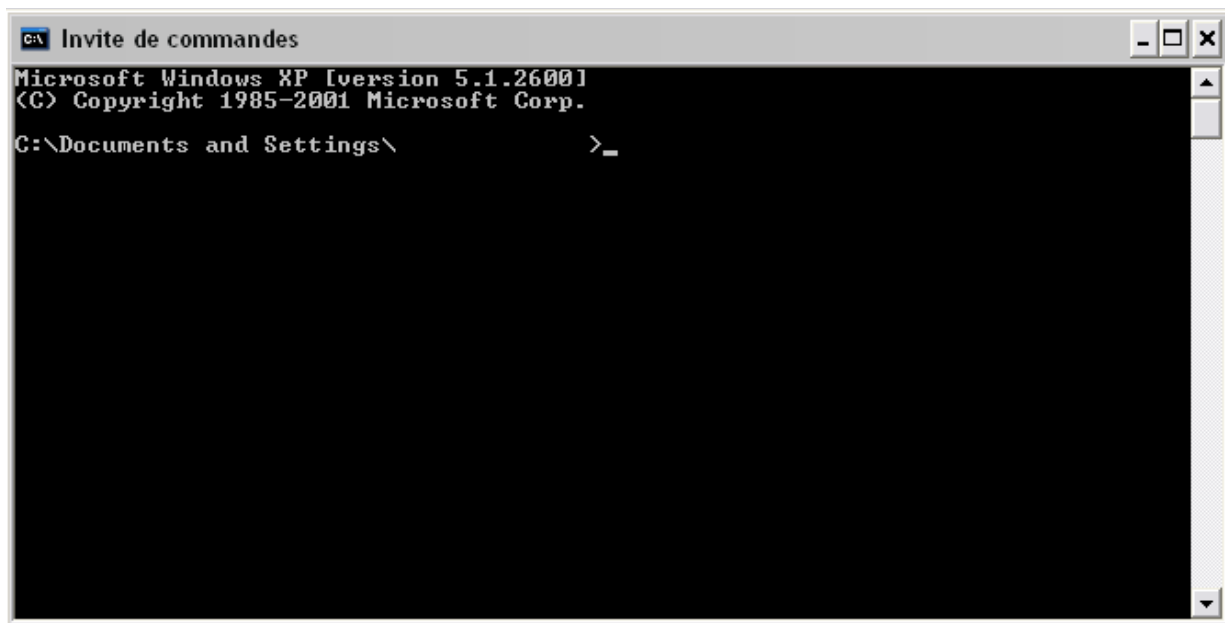


FIGURE III.5.1. – Invite de commandes sous Windows

Elle sera votre meilleure amie dans le monde du réseau, et nous allons beaucoup l'utiliser dans la suite du cours.

Ça fait peur hein ? 🍊 Il n'y a pas de quoi pourtant, même si ça paraît vieux, la console reste utile. Vous en aurez presque toujours besoin pour détecter les problèmes de réseau, vous vous en rendrez compte en temps voulu.

Dans l'invite de commandes, tapez `ipconfig` sous Windows ou `ifconfig` sous Linux et Mac OS.

Vous obtenez alors un résultat tel que celui-là :

```

1 Configuration IP de Windows
2
3
4 Carte Ethernet Connexion au réseau local:
5
6     Statut du média . . . . . : Média déconnecté
7
8 Carte Ethernet Connexion réseau sans fil:
9
10     Suffixe DNS propre à la connexion :
11     Adresse IP. . . . . : 192.168.1.17
12     Masque de sous-réseau . . . . . : 255.255.255.0
13     Adresse IP. . . . . :
14         fe80::218:deff:fe39:75c%5
15     Passerelle par défaut . . . . . : 192.168.1.1

```

Vous n'avez peut-être pas les mêmes rubriques : cela dépend de votre type de connexion (Wi-Fi, filaire...). Repérez la rubrique où vous avez des informations sur l'IP, le masque de sous-réseau, etc. Il est possible que, comme dans le résultat ci-dessus, vous ayez deux adresses IP. Si c'est

le cas, cela veut dire que votre système d'exploitation supporte le protocole IPv6 : vous avez donc une adresse IPv4 et une IPv6. D'après les descriptions que nous vous avons donné dans la partie 1, vous devriez les reconnaître. 🍊

Bien évidemment, vous verrez probablement une ou des adresse(s) IP différente(s), avec peut-être un autre masque de sous-réseau. Vous pourrez aussi avoir une passerelle par défaut différente.

Voilà, vous savez donc à quoi ressemble votre adresse IP. 🍊

Mais, vous n'avez pas tous la même, c'est un fait. Alors, comme nous ne sommes pas des ségrégationnistes en matière d'adresses IP, nous allons nous occuper de toutes les classes.



Toutes les classes? En réseau, les adresses sont *fashion*?



Eh non, c'est bien un cours de réseau informatique, pas de mode. 🍊 Une classe en réseau est, en fait, un ensemble d'adresses IP. Chaque adresse IP appartient à une classe principale (on dit aussi *une plage*). Chaque classe a un masque de sous-réseau **par défaut**. Que vous le vouliez ou non, dès que vous donnez à votre carte réseau une adresse IP, votre système d'exploitation lui assigne directement un masque de sous-réseau par défaut selon la classe à laquelle appartient votre adresse IP.

Par convention, les créateurs du protocole IP disent qu'il existe 5 classes d'adresses IP. En d'autres termes, on peut choisir notre adresse IP dans ces cinq classes (théoriquement hein, parce que la pratique, c'est encore autre chose 🍊). Ces classes sont : A, B, C, D et E.



Rappel : actuellement, les classes sont obsolètes. On peut considérer qu'elles n'existent plus. Néanmoins, il peut être utile de voir de quoi il s'agissait. Gardez à l'esprit, lors de la lecture de la suite de ce chapitre, cette information. Comment ça, on radote? 🍊

Maintenant que les présentations sont faites, étudions-les un peu plus en détail!

III.4.2. Classe A

Commençons par l'étude de la classe A.

III.4.2.1. Présentation

Nous vous avons dit qu'une classe d'adresses IP est en fait un ensemble d'adresses. Dans le cas de la classe A, ces adresses IP se situent entre 1.0.0.0 et 127.255.255.255. Son masque de sous-réseau par défaut est 255.0.0.0. En pratique, les adresses IP de la classe A se situent entre 1.0.0.0 (compris) et 126.255.255.255.



Mais alors, à quoi servent les adresses IP entre 127.0.0.0 et 127.255.255.255?

En fait, les adresses **IP** commençant par 127 sont utilisées pour faire des tests particuliers. Faisons un test. Reprenez votre invite de commandes, ou terminal (on vous l'avait bien dit que vous alliez beaucoup l'utiliser ;)). Sous Windows, tapez:

```
ping 127.0.0.1
```

Ou sous Linux :

```
ping -c 4 127.0.0.1
```

i

On verra plus tard ce qu'est ping en détails. Pour faire court : c'est un outil de diagnostic.

Si le protocole **TCP/IP** est correctement implémenté, c'est-à-dire si votre système d'exploitation est capable de se connecter à un réseau (on peut supposer que c'est le cas vu que vous êtes en train de lire cette page 🍌), vous aurez une suite de réponses de votre carte réseau, généralement 4 lignes. Nous vous laissons lire et comprendre ces quelques lignes, vous en êtes largement capables. 🍌

Revenons à l'étude de l'adresse 127.0.0.1. On l'appelle **loopback address**. D'accord, c'est de l'anglais... Cependant, dans le monde du réseau, c'est la langue principale, c'est donc important d'apprendre ce vocabulaire. 🍌

On va traduire ça ensemble. Le mot *loopback* signifie «boucle de retour». Donc, lorsque vous faites `ping 127.0.0.1`, vous faites en réalité un ping vers... votre ordinateur ! En fait, votre système d'exploitation crée automatiquement un réseau spécial composé uniquement de lui-même. Sous Linux, ce réseau spécial est représenté par l'interface **lo**.

?

Quel intérêt?

Eh bien, cela permet de tester des applications réseau sans être connecté réellement à un réseau. Par exemple, vous voulez tester un script PHP (ce qui nécessite un logiciel serveur Apache, généralement) mais vous n'êtes pas connecté à Internet, vous ne pouvez pas l'envoyer sur un serveur pour le tester. Votre ordinateur peut faire office de serveur, grâce à WAMP ou un logiciel de ce genre. Pour tester votre script, votre ordinateur se connecte à lui-même et s'envoie lui-même des requêtes. Ça paraît tordu comme ça, mais en fait c'est logique. 🍌

i

Notez que, sous Windows et Linux, toute adresse de la forme 127.XXX.XXX.XXX marchera à la place de 127.0.0.1. Si vous voulez, vous pouvez tester avec ping.

À travers cet exemple (certes un peu long), vous voyez qu'on ne peut pas utiliser les adresses 127.XXX.XXX.XXX.

Revenons maintenant à l'étude de la classe A. Ses adresses **IP** sont généralement utilisées dans de très très grandes entreprises et chez les **FAI**. Vous vous demandez pourquoi? Pour répondre, il va falloir nous intéresser à la structure d'une adresse **IP**.

III.4.2.2. Structure d'une adresse IP de la classe A

Prenons une adresse IP de la classe A. Au hasard : 110.0.0.1. Si vous avez bien retenu ce que nous avons dit plus haut, son masque de sous-réseau par défaut est 255.0.0.0. 🍊

Schématiquement, ça donne ceci :

110 . 0 . 0 . 1
└─
1 octet = 1 byte = 8 bits

Les réseaux de zéro - zestedesavoir.com

FIGURE III.5.2. – Décomposition d'une adresse IP de classe A

Une adresse IP est constituée de 4 octets. Votre ordinateur, lui, ne «voit» pas une adresse IP, comme vous et nous. Nous voyons des nombres décimaux tandis qu'il «voit» des nombres binaires, une suite de 0 et de 1 (à supposer que les ordinateurs «voient» 🍊).

Dans notre exemple, c'est-à-dire dans le cas d'une adresse IP de la classe A, le premier octet est l'**identité du réseau**, soit en anglais **network ID**.

?

Qu'est-ce que c'est ?

Cela indique simplement que l'**adresse client 0.0.1 se trouve dans le réseau 110**. Donc, à ce niveau, vous avez dû comprendre que la partie **0.0.1** est l'adresse de votre carte réseau. On l'appelle l'**adresse client**, ou, en anglais, **host ID**.

i

Si vous avez une adresse IP de 110.0.0.1, vous pouvez communiquer avec tous les hôtes de votre réseau (qui auront donc pour adresse IP 110.XXX.XXX.XXX). Par contre, si vous voulez communiquer avec un hôte dans le réseau 122, il vous faudra passer par... **une passerelle (un routeur)**. Vous ne l'aviez pas oublié, si ? 🍊

!

Notons une règle d'or : dans un réseau, deux clients (ordinateurs, imprimantes, etc.) ne peuvent pas utiliser une même adresse IP, de même que, dans un pays, 2 lignes téléphoniques ne peuvent pas avoir le même numéro attribué. 🍊

Bref, cela explique pourquoi ce sont les très grandes entreprises et les FAI qui utilisent ce type d'adresses. En effet, grâce à la répartition des octets entre network ID et host ID, vous pouvez avoir 16 777 214 adresses IP **par** réseau. De plus, vous pouvez avoir un total de 126 réseaux. Vous comprenez donc que ça intéresse les FAI qui doivent donner des adresses IP à un très grand nombre de personnes. 🍊

Un sous-réseau en anglais se dit **subnet** qui est le diminutif de *subnetwork*.

A priori, vous ou nous n'aurons pas vraiment affaire à cette classe, même dans le monde professionnel sauf si, bien sûr, vous travaillez pour des FAI. Dans ce cas, des formations telles que CISCO CCNA, CCNP, voire CCIE vous seront utiles. 🍊

III.4.3. Classes B et C

À présent, intéressons-nous aux classes B et C.

III.4.3.1. Classe B

Premièrement, parlons de la classe B. Il n'y a pas grand-chose à dire, voilà pourquoi elle vient en premier (et aussi parce que c'est l'ordre alphabétique 🍊).

III.4.3.1.1. Présentation

Les adresses IP de la classe B sont celles entre 128.0.0.0 et 191.255.255.255. Le masque de sous-réseau par défaut de cette classe est 255.255.0.0.

Seules des grandes ou moyennes entreprises vont utiliser ce type d'adresses IP pour raccorder plusieurs ordinateurs car dans la classe B, on a une possibilité de 65 534 ordinateurs **par** réseau. Comme pour la classe A, ce nombre vient de la structure des adresses IP de la classe B que nous allons étudier maintenant plus en détails. 🍊

III.4.3.1.2. Zoom sur la structure d'une adresse IP de la classe B

Prenons une adresse de la classe B pour notre étude. Par exemple : 172.40.0.5 (en fait, vous n'avez pas vraiment le choix 🍊).

La partie **172.40** est l'identité réseau et la partie **0.5** est l'identité client. On dit que l'adresse 0.5 se trouve dans le réseau 172.40. C'est le même principe que pour la classe A. 🍊

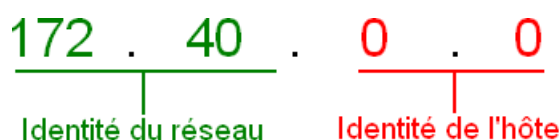
Mais pourquoi l'identité réseau prend deux octets dans ce cas?

C'est déjà bien si vous vous êtes posé cette question. Sinon, relisez ce chapitre : vous devez absolument bien comprendre les notions de *bits*, octets et *bytes*.

Bref, c'est grâce à une question comme celle-là que l'on se rend compte de l'importance d'un masque de sous-réseau. En effet, celui-ci **définit** quelles parties des 4 de votre adresse IP (ou quels octets de votre adresse IP) correspondent à l'identité réseau et quelles parties correspondent à l'identité client.

Quand on était dans la classe A, on avait un masque de sous-réseau de 255.0.0.0. Or, dans une adresse IP de la classe A telle que 110.0.0.1, seul le premier octet (ici 110) correspond à l'identité réseau. Maintenant, regardez son masque de sous-réseau, seul le premier octet est à 255, les autres sont à 0. Nous pensons que là vous avez compris comment ça fonctionne. 🍏

Reprenons notre adresse de la classe B. Comme dans notre masque de sous-réseau les deux premiers octets sont à 255, dans notre adresse IP, les deux premiers octets correspondent à l'identité réseau :



Les réseaux de zéro - zestedesavoir.com

FIGURE III.5.3. – Décomposition d'une adresse IP de classe B

III.4.3.2. Classe C

Abordons maintenant la classe C. Nous allons en parler un peu plus longtemps.

III.4.3.2.1. Présentation

Les adresses de la classe C sont entre 192.0.0.0 et 223.255.255.255. Le masque de sous-réseau par défaut est 255.255.255.0.

Cette classe est celle qui nous intéresse le plus. En effet, la plupart de nos adresses IP que nous avons vues en début de chapitre sont dans cette classe. 🍏 Après cela dépend aussi de votre FAI. Certains vous donneront des adresses privées et utiliseront des services comme NAT pour vous donner accès à Internet. Vous aurez plus d'informations sur les classes privées dans une autre sous-partie. 🍏

Dans cette classe on peut avoir 254 adresses IP par réseau, et 2 097 152 réseaux.



Pourquoi seulement 254 adresses IP par réseau? De 1 à 255, ça en fait 255, non?

Bonne remarque. 🍏 Pour répondre, il va nous falloir faire un passage rapide et indolore sur...

III.4.3.2.2. Les envois de données

Dans un réseau informatique, il y a plusieurs moyens d'envoyer des données.

- **L'unicast** : dans ce cas, on envoie des données à un seul ordinateur;
- **Le multicast** : l'envoi des données se fait vers un groupe d'ordinateurs;
- **Le broadcast** : on envoie des données à tous les ordinateurs du réseau.

Ce qu'il faut savoir, c'est que l'adresse 255 dans les réseaux de la classe C est **une adresse de broadcast réseau**. Nous avons déjà vu cette notion : si vous envoyez des données à cette adresse, les données seront envoyées à tous les ordinateurs du réseau. On a la même chose pour les adresses des classes A et B. Par exemple, l'adresse 255.255.255 du réseau 110 (pour la classe A) est une adresse de *broadcast*, ainsi que l'adresse 255.255 du réseau 140.20 (pour la classe B).

Un hôte ne peut donc pas prendre cette adresse IP, puisqu'elle sert, en quelque sorte, de support à l'envoi de données, ce qui explique qu'on ait seulement 254 adresses IP par réseau.

III.4.3.2.3. Structure d'une adresse IP de la classe C

Bon, je pense que vous avez compris le principe. 🍊 Pour vérifier ça, vous allez faire un exercice : prenez une adresse IP de la classe C au hasard, écrivez son masque de sous-réseau et dites quelle partie correspond à l'identité réseau et quelle partie correspond à l'identité client. Honnêtement, si vous n'êtes pas capables de faire cet exercice, nous insistons, **relisez tout le chapitre depuis le début**!

Voici une correction :

👁️ Contenu masqué n°3

Passons maintenant aux dernières classes qui nous restent à étudier.

III.4.4. Classes D et E

Consacrons-nous à présent aux classes D et E qui sont, en fait, sans grand intérêt pour nous.



En effet, vous pouvez vous permettre d'ignorer ces deux classes. Mais bon, pour le principe, nous allons quand même vous expliquer pourquoi on choisit de les ignorer et vous donner quelques informations.

?

Pourquoi ignorer les classes D et E?

Dans la pratique, vous n'utiliserez pas la classe D. Quant à la classe E, vous ne pouvez pas l'utiliser : c'est une classe **expérimentale**. Seuls les développeurs du protocole TCP/IP l'utilisent pour des expériences. 🍊

III.4.4.1. Quelques informations...

Pour la culture, nous allons vous donner quelques informations. Pour les classes D et E, le nombre de réseaux possibles et le nombre d'identités client ne sont pas connus : c'est **non-assigné**. On pourrait supposer qu'on utilisera des adresses de ces deux classes dans les années à venir mais, en réalité, cela ne risque pas d'arriver. En effet, il y a déjà une nouvelle version du protocole IP : IPv6. Il est donc logique que l'on migre vers le protocole IPv6.

Voici les portées de ces deux classes :

- Classe D : de 224.0.0.0 à 239.255.255.255 ;
- Classe E : de 240.0.0.0 jusqu'à 255.255.255.255.

Vous pouvez vous amuser à compter le nombre d'adresses IP possibles dans ces deux classes, cependant officiellement cela n'a pas été assigné. 🍏

III.4.5. Notion de classe privée

Nous allons à présent aborder la notion de classe privée.

?

Qu'est-ce qu'une classe privée?

Une classe privée est une portée d'adresses IP d'une certaine classe publique (A, B, C), mais réservée pour un usage particulier par des standards ou conventions. 🍏

Par exemple, la portée 169.254.0.0 - 169.254.255.255 fait partie d'une classe privée réservée par Microsoft pour le protocole APIPA. Celui-ci permet la configuration automatique des adresses IP lorsque aucune autre configuration n'est apportée. Sous Windows, quand aucune adresse IP n'est explicitement définie, le protocole DHCP tente de contacter d'autres éléments sur le réseau afin d'en obtenir une. Si aucune réponse n'est obtenue, c'est alors qu'on se retrouve avec une adresse de cette plage.

Microsoft a réservé cette classe pour l'assignation automatique des adresses IP lorsque aucune autre configuration manuelle ou dynamique n'a été faite. Cette portée d'adresse (ou classe privée) est très importante dans le diagnostic des ordinateurs qui tournent sous Windows.

Il y a aussi la portée des adresses IP de 192.168.0.0 - 192.168.255.255. Beaucoup d'entre vous ont une adresse IP commençant par 192.168, nous y compris. Comme ces adresses IP sont issues d'une classe privée, il faut donc utiliser un service particulier pour pouvoir accéder à Internet : ce service, c'est PAT que nous verrons plus tard. 🍏


Si vous avez une adresse IP d'une classe publique, vous n'avez éventuellement pas besoin d'une passerelle car vous avez accès au réseau public qu'est Internet. Mais un FAI « sérieux » vous donnera une adresse IP privée même si elle est dans la classe A, B, ou C. Oui, il y a aussi des classes privées dans les classes A et B. Dans la classe A les adresses IP allant de 10.0.0.0 à 10.255.255.255 sont des adresses privées, et dans la classe B, celles allant de 172.16.0.0 à 172.31.255.255 sont des adresses privées aussi.

Maintenant que vous connaissez cela, revenons dans notre époque et intéressons-nous à l'adressage CIDR, beaucoup plus actuel !

III.5. L'adressage CIDR

Nous vous avons dit en parlant des classes que ces dernières avaient presque entièrement disparu parce qu'elles étaient devenues obsolètes. Elles ont été remplacées par un système d'adressage plus fiable et plus performant, à savoir l'adressage CIDR, qui est l'objet de ce chapitre. Étant donné qu'il s'agit du système d'adressage utilisé actuellement, vous feriez mieux d'être plus concentrés que lorsque nous avons traité des classes. 🍏 C'est parti !

III.5.1. Révision de l'adressage par classes

Pour ne pas transformer cette section en cours d'histoire, nous vous proposons de lire [cet article](#) pour plus d'informations à caractère « culturel » sur la naissance et l'évolution d'Internet. 

Nous allons revoir le fonctionnement de l'adressage par classes pour vous préparer à la suite du tutoriel.



Les cours sur les réseaux ne sont pas toujours à jour, aussi la majorité des étudiants continue à croire que les classes sont toujours d'actualité. Elles ont bien existé, mais elles ont été supplantées par l'adressage CIDR.

L'adressage par classes est un système utilisant une architecture réseau appelée en anglais *classful network*, c'est-à-dire « réseau dans les classes ». Cette expression traduit que le principe — majeur, sinon l'unique — de ce système d'adressage est de répartir les adresses IP par classes.

Le réseau ayant pour but de permettre la communication entre machines, les créateurs de la pile TCP/IP se sont inspirés du monde réel pour créer un système de communication informatique. Dans une société donnée, des individus vivent dans des maisons (en général), parlent une langue, ont un nom unique, vivent parfois dans des villes, occupent certaines fonctions, ont des responsabilités et des droits. Un système de communication informatique reprend plus ou moins ces grandes lignes : des individus (hôtes) parlent des langues (protocoles), ont des noms uniques (adresses IP), vivent dans des maisons (sous-réseau ou réseau, c'est selon), occupent des fonctions (clients, serveurs, passerelles, routeurs, etc.), ont des responsabilités (transmettre des données à la demande du client, distribuer automatiquement des adresses IP pour le cas d'un serveur DHCP) et des droits (réclamer un renouvellement d'adresse IP, demander l'identité d'un autre hôte, exiger un mot de passe, etc.).

Ainsi, pour rendre opérationnel ce tout nouveau système de communication, il y avait une espèce de sac plein de noms (adresses IP). Alors, on assignait un nom (une adresse IP) à chaque individu (hôte). Pour gérer cette distribution d'adresses, on a créé l'adressage par classes.

Pour reprendre un exemple, imaginez que vous ayez 10 000 prénoms à attribuer à autant de nouveau-nés. Pour mieux gérer cette attribution, vous pourriez classer les prénoms par ordre

III. Veuillez-vous identifier pour communiquer

alphabétique : tous les prénoms commençant par un A sont réunis dans un dossier « Prénoms A », etc. Alors, si une dame venait vous demander dix prénoms pour ses dix futurs enfants, vous n'auriez qu'à lui demander les contraintes auxquelles doivent répondre ces prénoms :

« Je veux des prénoms qui commencent par la lettre A et constitués de cinq lettres », dirait-elle. Rien de plus simple que de regarder votre dossier « Prénoms A » et d'en choisir dix. Mais que se passerait-il si, après avoir reçu les dix prénoms, la dame ne mettait au monde qu'un seul enfant (exemple très original 🍊)? Quid des neuf autres prénoms? Ce serait du gaspillage!

Le système d'adressage par classes fonctionne selon le même principe : les adresses IP sont rangées par classes et dans chacune d'elles se trouvent des plages. Si une entreprise demandait des adresses pour cent ordinateurs, on choisirait la classe lui offrant ce nombre d'adresses et on lui offrirait des adresses IP issues de cette classe.

Le problème de ce système d'adressage est le pourcentage assez élevé de perte d'adresses. Nous avons vu que toutes les adresses IP de la classe A, par exemple, nous permettaient d'obtenir 16 777 214 adresses IP par réseau en utilisant les masques par défaut. Cela dit, l'entreprise qui voudrait une adresse IP pour un réseau de 10 000 hôtes aurait quand même 16 767 214 d'adresses en surplus. Quelle perte!

Si l'adressage par classes n'avait pas été remplacé depuis les années 1990, aujourd'hui nous utiliserions presque exclusivement les adresses IPv6, car nous aurions très vite connu une pénurie d'adresses. C'est pourquoi un nouveau système d'adressage, capable de réduire au minimum le gaspillage d'adresses IP et de faciliter considérablement le routage, a été mis en place. Nous allons le voir dans la prochaine section.

III.5.2. CIDR et le supernetting



La compréhension de cette section demande beaucoup de concentration. Vous êtes priés de rester attentifs tout au long de votre lecture. Prenez une feuille et un crayon pour faire les calculs au même moment que nous. Si vous vous contentez de les lire comme on lirait un roman, vous les trouverez très difficiles. Nous ne nous attendons pas non plus à ce que vous maîtrisiez le *supernetting* du premier coup : même des étudiants ont du mal à appréhender cette notion.

L'adressage sans classes (ou adressage CIDR) est le système de gestion et d'allocation d'adresses IP le plus utilisé aujourd'hui. Ce système, qui est régi par les RFC 1518 et 1519, a été conçu pour remplacer l'adressage par classes pour les raisons que nous avons évoquées dans les chapitres précédents. Le but de ce nouveau système s'articule autour de deux points :

- Économiser les adresses IP.
- Faciliter le routage.



CIDR? Ce ne serait pas plutôt cidre?



III. Veuillez-vous identifier pour communiquer

Nous parlons de réseaux et non de boissons... 🍷 CIDR est l'acronyme de *Classless Inter Domain Routing* (« routage sans classes entre domaines »). Plutôt bizarre, non ? En bref, par CIDR comprenez « routage effectué entre domaines qui n'utilisent pas les classes ». On comprend alors que le réseau Internet est fondé sur ce système d'adressage. Logique, quand on y pense... Sinon, comment un système d'adressage par classes aurait-il pu supporter plus de 2 milliards d'internautes ? Depuis les années quatre-vingt-dix, nous n'aurions plus d'adresses IP disponibles.

i

En anglais, les adresses IP utilisant l'adressage CIDR sont appelées *classless addresses* par opposition aux *classful addresses*, qui désignent celles qui utilisent l'adressage par classes. Habituez-vous à ce vocabulaire qui est très présent dans les documentations en anglais.

Quand nous parlons d'assignation d'adresses IP, en tant qu'administrateur d'un réseau, nous devons examiner deux choses :

- Les contraintes administratives pour obtenir et allouer les adresses.
- L'aspect technique (sous-entendu le routage, le plus souvent) que cela implique.

CIDR répond mieux aux contraintes techniques.

III.5.2.1. CIDR : le comment

Nous allons maintenant nous focaliser sur le comment, étant donné que vous savez déjà pourquoi ce nouveau système a été créé.

Soit l'adresse 192.168.10.0/23. À ce stade, vous êtes censés savoir que le nombre après le slash (/) équivaut au nombre de bits masqués. Si vous avez encore des difficultés, nous vous recommandons la relecture de la sous-partie sur la notation du masque.

Bien ! 192.168.10.0/23 applique un masque de 255.255.254.0 au réseau 192.168.10.0. Grâce à cette notation, nous pouvons calculer (et vous êtes censés savoir le faire seuls à présent) l'étendue du sous-réseau qui ira donc de 192.168.10.0 à 192.168.11.255 (si nous incluons l'adresse de diffusion ou *broadcast address*), dans un réseau sans classes. Par contre, si nous étions dans un réseau n'utilisant pas l'adressage CIDR, **192.168.10.0/23** représenterait **une fusion de deux sous-réseaux de la classe C, 192.168.10.0 et 192.168.11.0 ayant chacun un masque de sous-réseau de... 255.255.255.0**.

Cela dit, avec l'adressage CIDR, le masque /23 nous donne l'équation suivante :

192.168.10.0/23 (adressage CIDR) = 192.168.10.0/24 (ou 255.255.255.0) + 192.168.11.0/24 (ou 255.255.255.0)

Vous voyez ? Nous avons la possibilité d'utiliser un seul réseau qui fusionne plusieurs sous-réseaux. Cette fusion de sous-réseaux, dite aussi *supernetting*, est l'essence même de CIDR. Cette technique est également appelée résumé de routes (*route summarization* en anglais).



Cette pratique viole la règle d'or du *subnetting*. Vous vous en souvenez ? Celle des 0 (*network ID*) et des 1 (*broadcast address*). Tous les routeurs qui supportent ce type d'adressage ignorent également cette règle.

Pour implémenter un réseau fondé sur l'adressage CIDR, il faut utiliser un protocole qui puisse le supporter. Il en existe plusieurs, tels que BGP et OSPF. Si le protocole ne supporte pas ce type d'adressage, le routage échouera dans ce réseau. En général, les petits LAN et les réseaux « maison » n'implémentent pas l'adressage CIDR.

III.5.2.2. Comment résumer une route

Dans l'adressage par classes, nous utilisons le *subnetting* pour réduire la congestion d'un réseau en le subdivisant en plusieurs sous-réseaux. Toujours est-il que nous perdions quelques adresses IP, étant donné que plusieurs sous-réseaux utilisaient un même masque. Cela dit, chaque sous-réseau avait le même nombre d'adresses. « Supernetter » un réseau est exactement le contraire de « subnetter » un réseau, sauf qu'ici, il ne s'agit plus de l'adressage par classes mais de l'adressage CIDR. Tous ces sous-réseaux peuvent donc être fusionnés et rassemblés sous un seul préfixe.

Un exemple vaut mieux que tout ce pavé. 🍌 Si nous avons quatre *subnets* tels que :

- Subnet 1 : 192.168.0.0/24 soit 11000000. 10101000. 00000000.00000000/24
- Subnet 2 : 192.168.1.0/24 soit 11000000. 10101000. 00000001.00000000/24
- Subnet 3 : 192.168.2.0/24 soit 11000000. 10101000. 00000010.00000000/24
- Subnet 4 : 192.168.3.0/24 soit 11000000. 10101000. 00000011.00000000/24

Nous remarquons que ces quatre *subnets* ont bien le même préfixe /24 : nous pouvons les fusionner sous un seul préfixe. Par conséquent, nous obtenons la route suivante : **192.168.0.0/22 soit 11000000.10101000.00000000.00000000/22.**



Comment avez-vous obtenu le /22 alors que nous avions /24 au départ ?

Très belle question ! Nous avons simplement appliqué la technique d'agrégation de routes. Supernetter, c'est la même chose qu'agréger des routes. Le résultat obtenu est donc appelé route agrégée ou route résumée. Pour obtenir le /22, nous avons suivi quatre étapes bien précises que vous devez suivre également.

- **Étape 1 : détecter les réseaux ayant le même préfixe**

Dans cette étape, nous avons pris quatre réseaux ayant le même préfixe (/24) : il s'agit de 192.168.0.0, 192.168.1.0, 192.168.2.0 et 192.168.3.0.

III. Veuillez-vous identifier pour communiquer

- **Étape 2 : convertir des réseaux en binaire**

Ensuite, nous avons converti chaque adresse réseau en binaire. Pourquoi ? Parce que c'est important pour l'étape 3.

- **Étapes 3 et 4 : détecter les motifs entre les sous-réseaux en binaire (étape 3) et les compter (étape 4)**

Ne paniquez pas : ce n'est pas difficile. Quand nous avons converti les quatre sous-réseaux en binaire, qu'avons-nous obtenu ?

```
— Subnet 1 : 11000000. 10101000. 00000000.00000000
— Subnet 2 : 11000000. 10101000. 00000001.00000000
— Subnet 3 : 11000000. 10101000. 00000010.00000000
— Subnet 4 : 11000000. 10101000. 00000011.00000000
```

Y a-t-il quelque chose de commun à ces quatre sous-réseaux ? Rien ? Vous en êtes sûrs ? Nous allons vous faciliter la tâche.

```
— Subnet 1 : 11000000.10101000.00000000.00000000
— Subnet 2 : 11000000.10101000.00000001.00000000
— Subnet 3 : 11000000.10101000.00000010.00000000
— Subnet 4 : 11000000.10101000.00000011.00000000
```

Et maintenant? 🍊

Tous ces sous-réseaux ont 11000000.10101000.000000 en commun. Comptons le nombre de bits : il y en a 22, n'est-ce pas ? 22 sera donc notre nouveau préfixe. Le *network ID* sera la plus petite adresse IP parmi les quatre, soit 192.168.0.0. Enfin, la nouvelle route, la route résumée ou agrégée, sera **192.168.0.0/22**.

Voilà, vous savez tout sur le *supernetting* et le *subnetting*. Nous y reviendrons certainement une fois que vous maîtriserez le routage, afin que nous constations combien il est efficace de résumer les routes pour ne pas alourdir la table de routage.

III.5.2.3. Quelques exercices pour la route

Ne croyez pas que nous allons vous laisser filer comme ça, il vous faut pratiquer et encore pratiquer !

- **Exercice 1 : supernetting**

Votre premier exercice est relativement simple. Plus haut, nous avons pris le cas d'un réseau 192.168.10.0/23 et nous avons évoqué l'équation suivante :

```
192.168.10.0/23 (adressage CIDR) = 192.168.10.0/24 (ou 255.255.255.0) + 192.168.11.0/24 (ou 255.255.255.0)
```

III. Veuillez-vous identifier pour communiquer

Prouvez que **192.168.10.0/23** est bel et bien une fusion (une route agrégée) de **192.168.10.0/24** et **192.168.11.0/24**. C'est très simple, il suffit de respecter les étapes que nous avons définies plus haut.

• Exercice 2 : stagiaire chez Link it Technology

Vous êtes stagiaire dans une entreprise éditrice de logiciels nommée *Link it Technology*. Le réseau de ladite entreprise est constitué de 192 hôtes parmi lesquels 4 serveurs :

- **Srvprog** est le serveur que les programmeurs (au nombre de 47) de l'entreprise utilisent. Il héberge un nombre important d'applications.
- **Srvcomp** est le serveur des 76 comptables. Il héberge également un nombre important d'applications de comptabilité.
- **Srvprint** est le serveur d'impression des secrétaires. On en compte 33 qui effectuent un nombre considérable d'impressions par jour, ce qui alourdit le réseau et empêche aux autres services de communiquer plus rapidement avec leurs serveurs respectifs.
- **Srvboss_backup** est le serveur sur lequel sont sauvegardés tous les fichiers des chefs de division. Le système de *backup* de l'entreprise est automatique : chaque fois qu'un fichier est modifié et sauvegardé, une copie est sauvegardée aussitôt sur ce serveur. Les chefs de division, tasse de café à la main chaque matin, modifient plusieurs fichiers. Ils sont au nombre de 36 (les chefs, pas les fichiers!).

Votre chef se plaint de la congestion du réseau et vous demande de mettre en place un plan d'adressage pour minimiser le trafic. Vous devrez donc, à partir de l'adresse réseau 120.12.0.0/18, aboutir à une solution satisfaisante. Dans ce cas précis, il vous est demandé de subnetter ce réseau en quatre :

- Un réseau **netprog** pour tous les développeurs de l'entreprise et leur serveur.
- Un réseau **netcomp** pour tous les comptables et leur serveur.
- Un réseau **netsecr** pour tous les secrétaires et leur serveur de fichiers.
- Un réseau **netbackup** pour tous les chefs de division et leur serveur *backup*.

i

Pour cet exercice, nous supposons que les sous-réseaux ne communiquent pas entre eux, donc que le trafic reste interne. Par conséquent, nous n'avons pas besoin de routeurs et de calculer les plages pour leurs interfaces. Dans la réalité, ce calcul est obligatoire mais ignorez cette étape pour cet exercice. Vous le ferez dans la prochaine sous-partie.

À partir de l'énoncé ci-dessus, votre mission est triple :

- Déterminer le network ID de chaque *subnet* et leur masque.
- Déterminer les plages d'adresses de chaque *subnet* en incluant leur *broadcast address*.
- Appliquer la technique du supernetting pour avoir une route résumée des *subnets* que vous aurez obtenus.

i

Les sous-réseaux n'ont pas le même nombre d'hôtes ; ainsi pour déterminer combien de bits il faut masquer, vous devrez vous focaliser sur le plus grand sous-réseau. Dans notre

i

scénario, il s'agit du sous-réseau netcomp (76 hôtes pour les comptables). Trouvez un masque qui vous permette d'avoir au moins 76 hôtes par sous-réseau. Nous perdrons des adresses, certes, mais vous ne savez pas encore comment implémenter des masques à longueur variable. Pour cela, rendez-vous à la prochaine sous-partie.