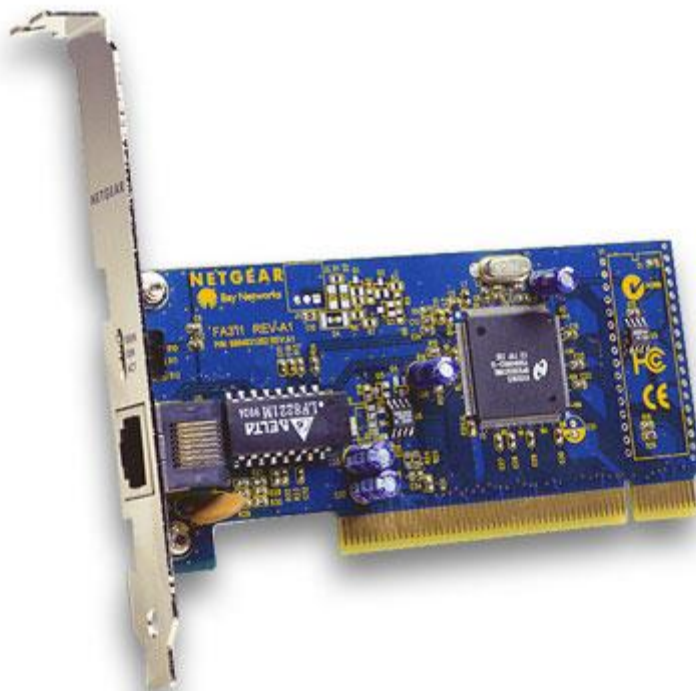


FORMATION RESEAUX

I - MATERIEL

LA CARTE RESEAU :

Est le composant le plus important, elle est indispensable. C'est par elle que transitent toutes les données à envoyer et à recevoir du réseau dans un ordinateur. Il n'y a pas grand-chose à dire sur cet appareil. La seule chose que vous devez connaître, c'est la notion **d'adresse MAC** : c'est l'adresse physique de la carte. Elle permet d'identifier la machine dans un réseau, un peu comme l'**adresse IP**. Nous ne devrions pas encore en parler, mais il serait bien difficile de comprendre le fonctionnement de certains matériels... Pour faire court et ne pas vous embrouiller si tôt, l'adresse physique est relative à la carte réseau. Elle lui est attribuée à sa fabrication et ne peut pas changer (ce n'est pas tout à fait vrai, mais l'idée est là). L'adresse IP est relative au réseau : elle change tout bonnement suivant le réseau. Vous comprendrez mieux ce que sont ces adresses dans la sous-partie sur le commutateur (switch). La carte réseau est aussi appelée NIC en anglais, pour Network Interface Card. Voici à quoi peut ressembler une carte réseau :



CONCENTRATEUR (HUB)

Un hub est un dispositif en réseau qui permet de mettre plusieurs ordinateurs en contact. Définition pas très précise, puisque tout dispositif en réseau (ou presque) a le même but. Bref, ce qu'il faut retenir est qu'un hub est très bête, enfin, moins intelligent que les autres. **Ce qu'il fait est tout simple : il reçoit des données par un port, et envoie ce qu'il reçoit aux autres. Il a une interface de réception (un port) et une interface de diffusion (plusieurs autres ports par où les autres ordinateurs sont connectés).**

Attention, une interface permet la réception ET la diffusion. Comme vous pouvez le voir sur la photo ci-dessous, le hub n'a pas juste deux interfaces physiques, où on entre par la gauche et on ressort à droite, non ! L'interface de réception est logique.

Exemple : j'ai un hub à 4 ports, avec 4 ordinateurs connectés. J'ai le port 1, 2, 3, 4 (ici, interface = port). Si l'ordinateur 4 (au port 4) veut communiquer avec les autres, moi le hub, je reçois les données au port 4 (c'est mon port de réception), je renvoie les données aux ports 1, 2, et 3 : ce sont les ports de diffusion.

Je ne renvoie plus les données au port 4, car c'est mon port de réception.



Un hub, ou concentrateur

Ce qu'on lui reproche est le manque de confidentialité, et oui, le hub ne garde pas de secret : tout ce qu'un ordinateur lui dit, il l'envoie aux autres. Heureusement, les autres vérifient bien si ça leur est destiné, et si ça ne l'est pas, ils laissent tomber les données et ne les lisent pas.

C'est toujours sécurisant, non ?

Non, pas du tout, à partir du moment où les données arrivent jusqu'à la carte réseau, elles peuvent toujours être lues (mais on n'est pas là pour un cours de sécurité informatique).

COMMUTATEUR (SWITCH) ET ROUTEUR :

Le commutateur (ou switch) et le routeur sont 2 appareils fondamentalement différents, et pourtant, leurs rôles se ressemblent tellement ! Au-delà de leur architecture, il faut comprendre leur différence au niveau d'un réseau.

LE COMMUTATEUR : JUSTE UNE HISTOIRE D'ECHANGE DE DONNEES

Un commutateur fonctionne à peu près comme un hub, sauf qu'il est plus discret et intelligent. **Il n'envoie pas tout ce qu'il reçoit à tout le monde, mais il l'envoie uniquement au destinataire.** Si l'ordinateur 1 envoie des données à l'ordinateur 2, seul ce dernier les recevra et pas les autres connectés. Afin de déterminer l'ordinateur à qui il faut renvoyer les données, **le switch se base sur les adresses physiques (adresses MAC) des cartes réseau.** Pour faire une analogie avec la vie réelle, une adresse MAC est un peu comme une adresse postale. C'est une suite de **6 nombres hexadécimaux**, par exemple **00-16-D4-C7-6E-D3**. Si vous ne savez pas ce qu'est l'[hexadécimal](#), ce n'est pas bien grave mais vous devriez quand même en prendre connaissance, on en a souvent besoin en informatique (et pas qu'en réseau). Nous n'étudierons pas les adresses MAC dans ce chapitre, elles seront étudiées à partir de la partie 3, lorsque nous aborderons réellement la communication dans un réseau.

Un commutateur transmet donc des données aux autres ordinateurs en se basant sur leurs adresses MAC. Les transmissions sont plus confidentielles, les autres ne savent rien des données ne leur étant pas destinées.



LE ROUTEUR, UN VERITABLE ORDINATEUR

Un routeur ressemble à un switch sur le plan de l'utilisation : en effet, il permet de mettre plusieurs ordinateurs en réseau. Mais cela va plus loin : il permet de mettre en contact 2 réseaux fondamentalement différents. **Dans une petite installation, avec un ou plusieurs ordinateurs connectés à une "box" (qui est en fait un routeur), il est la frontière entre le réseau local et Internet.**

Un routeur a plusieurs interfaces. Pour continuer dans notre exemple de frontière avec Internet, il possède une interface connectée à Internet (généralement, cela se traduit par un câble branché sur la prise téléphonique) et plusieurs autres interfaces sur lesquels se connectent des ordinateurs voulant accéder à Internet (ce qui se traduit généralement par des câbles Ethernet ou des connexions Wi-Fi). Notez aussi que le routeur n'est pas uniquement utilisé pour aller sur Internet, on l'utilise aussi dans un réseau strictement local.

REPETEUR

Un répéteur (repeater en anglais) agit un peu comme un hub, mais ce dernier n'a que 2 interfaces. Son intérêt est de renvoyer ce qu'il reçoit par l'interface de réception sur l'interface d'émission, mais plus fort. On dit qu'il régénère et réémet le signal. En transmission sans fil (radio, téléphone) on parle aussi de relais. Un répéteur permet de couvrir des distances plus grandes que les distances maximales fixées par le matériel que l'on utilise : par exemple, dans un réseau sans fil (Wi-Fi), la portée maximale entre 2 appareils est d'environ 50 mètres en intérieur. En plaçant un répéteur peu avant ces 50 mètres, vous pouvez connecter 2 appareils à 100 mètres de distance. Le fait que les informations soient renvoyées "plus fort" peut dégrader la qualité du signal dans les réseaux sans fil. Pour prendre un exemple parlant, en radiophonie, si l'on se trouve trop loin d'un relais, la qualité du son que l'on entend est dégradée.

II - TOPOLOGIE DES RESEAUX

LAN : LE RESEAU 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. Ce type de réseau utilise généralement une configuration de type domaine comme nous l'avons vu précédemment.

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).

WAN : LE RESEAU ETENDU



WAN signifie Wide Area Network, en français, on peut le traduire par "réseau étendu". Un WAN est en fait une association de plusieurs LAN. Supposons 3 LAN formés par des switches : le "branchement" des 3 switches sur un autre switch forme un WAN, car on associe plusieurs LAN entre eux. Nous pourrions l'utiliser pour obtenir un seul réseau virtuel dans deux endroits géographiques différents. Cet exemple peut être contesté, car on parle plus souvent de WAN pour des réseaux très grands, à échelle régionale voire nationale, mais l'idée est là.

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.

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.

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.

RESEAU 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. 😞

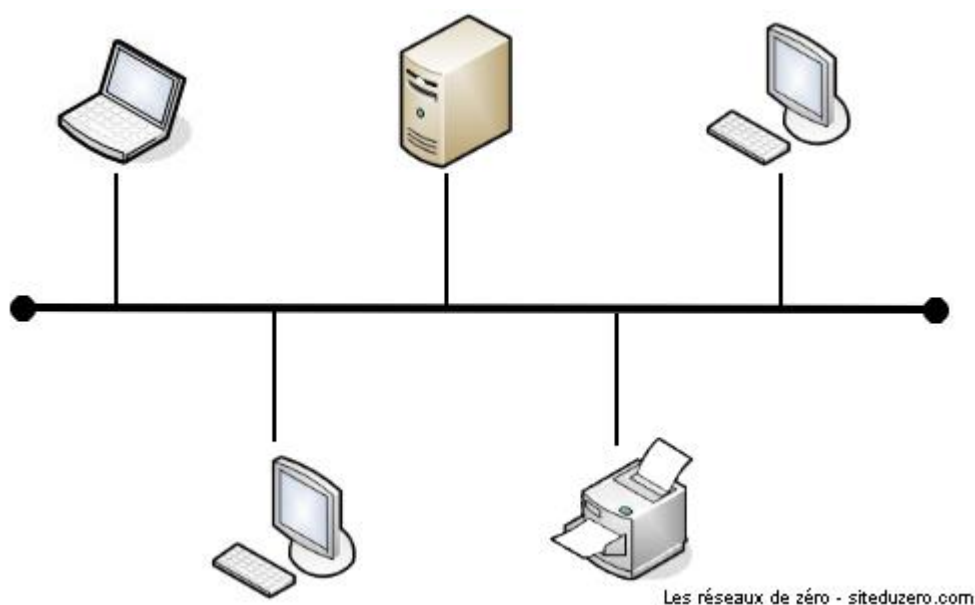
Il y a d'autres raisons qui font que cette topologie est très peu utilisée.

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 !

Représentation schématique d'un réseau en bus

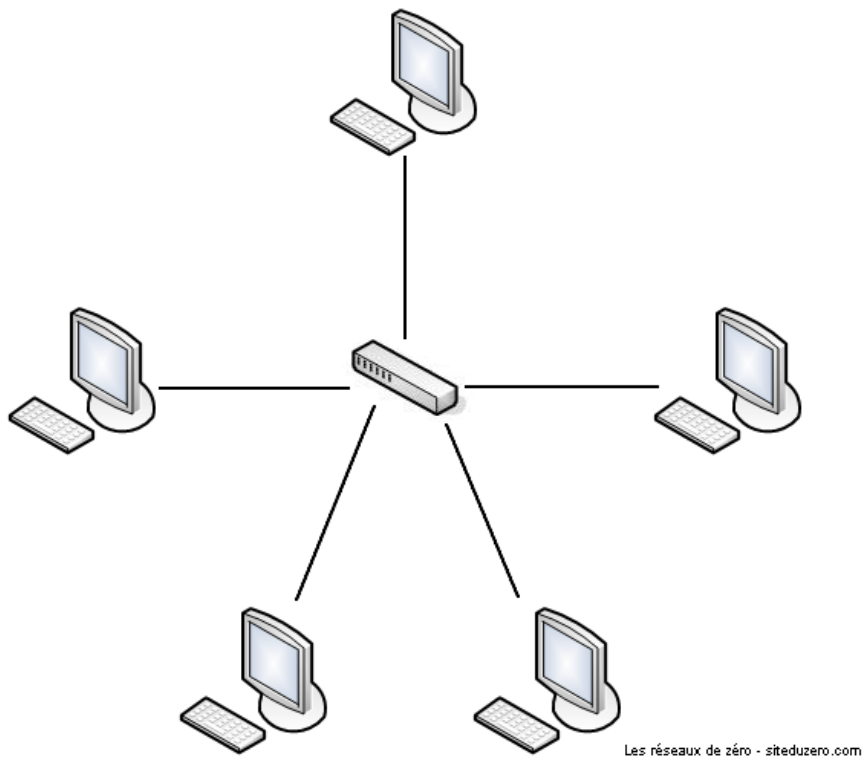


TOPOLOGIE DE TYPE ETOILE

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

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 !

RESEAU 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.

Je vais 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.

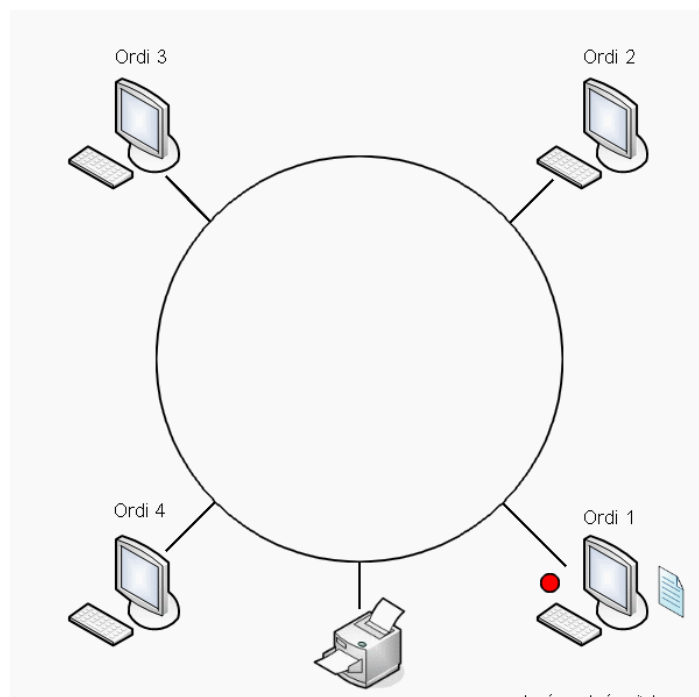
Le principe est assez simple : une machine connectée au réseau possède un jeton virtuel. Ce jeton, c'est une **autorisation de communiquer**. Une fois que la machine a transmis ce qu'elle voulait, elle passe le jeton à la machine suivante, et ainsi de suite. Si le détenteur du jeton n'a rien à dire,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

On va me dire que je radote, mais je 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.

*Réseau en anneau.
attendent le jeton
transmettre des*



*Des ordinateurs
(token) pour
données.*

III - LES PROTOCOLES RESEAUX

INTRODUCTION AUX PROTOCOLES

Un protocole est **un ensemble de règles qui définissent comment se produit une communication dans un réseau**.

Pour mieux appréhender cela, nous allons considérer deux analogies.

LE PROTOCOLE : UN GENRE DE PILOTE

Un protocole joue un peu le même rôle qu'un pilote : ils ont beaucoup de similitudes. **Un pilote permet au matériel de communiquer avec le système**. En d'autres termes, un pilote c'est le protocole de communication entre le matériel et le système.

Sans un pilote, votre souris ne peut pas fonctionner, elle ne peut pas **communiquer** avec le système. Vous comprenez donc que le pilote est l'interface de communication entre le système et le matériel, il en est de même pour le protocole.

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

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.

L'UTILITE 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.

Citation

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 **ICMP packet (Internet Control Message Protocol)** qui enverra une **erreur de type 3 (destination unreachable, destination inaccessible) et de code 7 (Destination host unknown, destinataire inconnu)**.

ICMP est un protocole dans la suite protocolaire TCP-IP utilisé pour envoyer des messages D'ERREURS DANS UN RESEAU.

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.

Citation

« 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.

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.

Citation

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.

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 DONNEES :

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'une entête et du contenu. L'entête sera un peu « réservée » 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 couvrir cette notion avec plus de détails que ce que l'on vous a appris dans la partie II du cours. Mais vous êtes sans ignorer que le routage consiste à « diriger » les données entre deux réseaux d'un plan d'adressage différent.

DETECTION 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.

ACCUSE DE RECEPTION :

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**". 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'essiera 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.

CONTROLE DE SEQUENCES :

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 ! 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. C'est une suite (une pile pour être précis) de protocoles en fait, **protocol stack** en anglais.

IV - MODELE OSI - MODELE TCP/IP

LE MODELE 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 pourvoit des services que la couche en cours utilise, et la couche en cours pourvoit 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 :

Image utilisateur

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 t1, 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.

► Pour Le Réseau, Tout Se Passe Automatiquement.

Le modèle OSI par l'exemple : le facteur

Oui, nous le savons, vous êtes impatient(e) ; 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 :



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 !

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.

COUCHE PRESENTATION

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 cryptage/décryptage, 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).

Le cryptage 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 cryptage a ses avantages.

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.

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.

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.**

COUCHE RESEAU

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.**

LA COUCHE LIAISON

1. Présentation effectuée ? O.K. !
2. Session établie ? O.K. !
3. Transport en cours ? O.K. !
4. 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.

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.

FINALEMENT : LA COUCHE PHYSIQUE

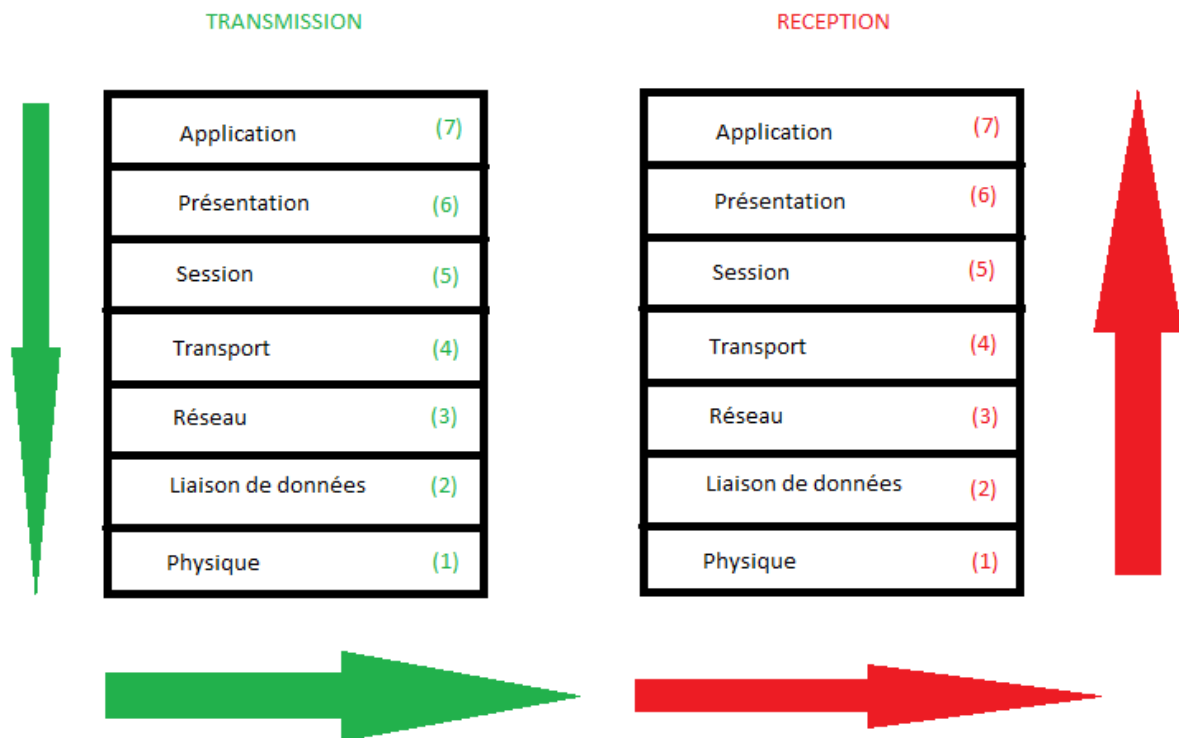
Notre mail est en cours de transport, mettons-le sur le média. La couche physique reçoit les trames de la couche de liaison de données et les « convertit » en une succession de bits qui sont ensuite mis sur le média pour l'envoi. Cette couche se charge donc de la transmission des signaux électriques 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.**

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

POSITION DANS LE MODELE OSI

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, cryptage, 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 bits et transmission physique des données sur le média.

☀ 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.



TCP/IP vs OSI : le verdict ?

Vous vous êtes peut-être posé la question de savoir pourquoi le titre de cette partie était Les modèles de communication et les protocoles plutôt que Le modèle OSI et les protocoles. En effet, nous allons 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.

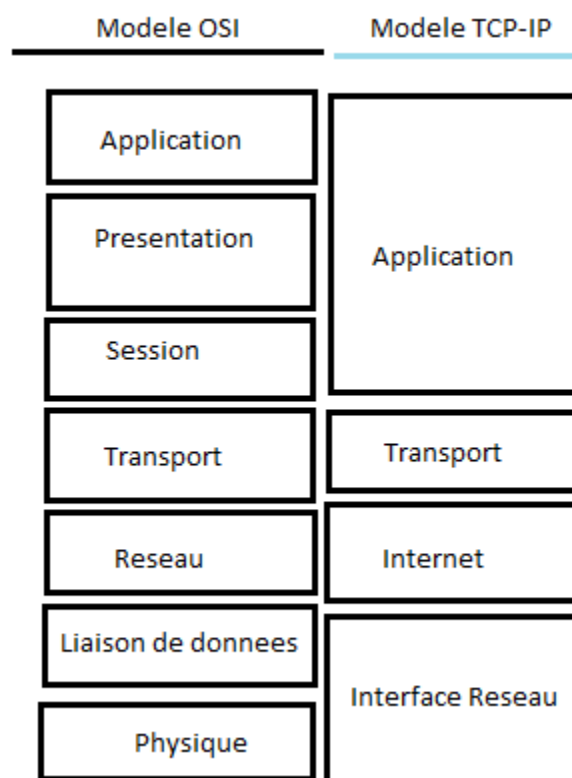
LE MODELE TCP/IP

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.

Voici un schéma comparatif des deux modèles.



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.

Le formatage des données dans le modèle TCP/IP peut également se faire via des bibliothèques.

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 :

POINT VOCABULAIRE : LES UNITES DE DONNEES PDU

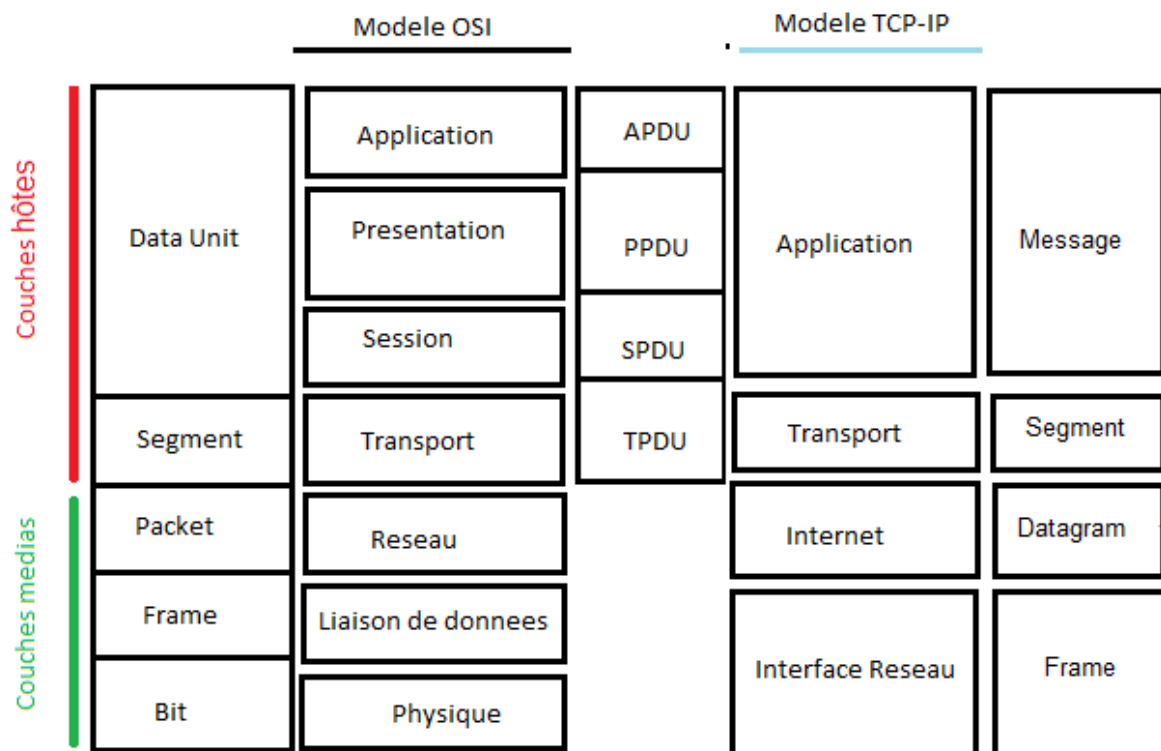
Au début de la communication entre deux hôtes, chaque information qui sera transmise est une donnée. Cependant, cette donnée a plusieurs unités selon la couche dans laquelle elle se trouve : il s'agit de la même donnée, mais sous plusieurs appellations. Prenons un exemple : votre père, vous l'appellez 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. :D Les acronymes dans l'image ci-dessous sont en anglais parce qu'ils sont plus courts. :p Vous ne devriez pas avoir de difficulté à les comprendre puisque leurs équivalents français sont juste plus haut.



www.siteduzero.com

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). Faites-nous confiance, ne demandez pas de détails pour l'instant.

Tout au long du tutoriel, nous ne ferons pas forcément usage du nom approprié pour une couche donnée. Nous utiliserons souvent les mots données et paquets pour faire référence à toute information qui se transmet. L'utilisation du mot approprié interviendra lorsqu'elle sera de rigueur.

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.

CRITIQUES DU MODELE 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.

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 ! :D 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.

L'IMPLEMENTATION

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

LA DUREE 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.

CRITIQUES DU MODELE 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.

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. Cependant, je (ce n'est peut-être pas l'avis de tous les rédacteurs de ce tutoriel, mais de celui qui rédige en ce moment) 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.

Même si je suis pro-OSI, cela ne veut pas dire que ce modèle remportera définitivement le duel. Au train où vont les choses, TCP/IP régnera pendant encore très longtemps. Mais sait-on jamais ?... Il se pourrait qu'un meilleur modèle voie le jour...

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

Oui : apprendre l'anglais ! :p

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 :

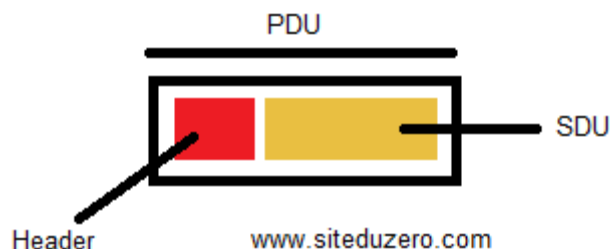
Image utilisateur

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 :



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 chambre en ce moment. Soyez concentrés. ;)

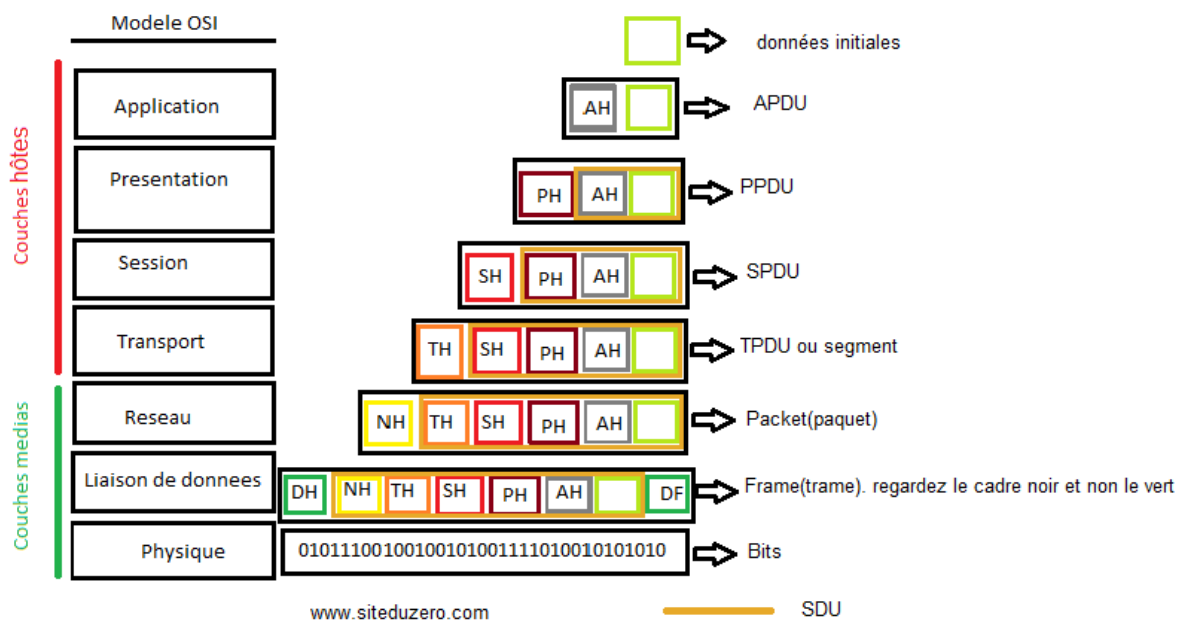
Quand vous écrivez une (vraie) lettre, c'est un SDU. Vous la mettez dans une enveloppe sur laquelle est écrite une adresse. Cette lettre qui n'était qu'un SDU devient un PDU du fait qu'elle a été enveloppée (encapsulée). Votre lettre arrive à la poste. Un agent du service postal regarde le code postal 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/attentive. 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 :



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 or 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 (Application Header : « 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, 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 !

Image utilisateur

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.

V - ADRESSAGE

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.

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.

ADRESSE IP : L'ADRESSE RELATIVE AU RESEAU

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 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. On peut comparer le nom d'une rue à un **masque de sous-réseau**, et l'adresse IP au numéro de chacune des maisons.

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 messenger (chaque *passerelle par défaut*) aux carrefours possède une adresse IP spéciale relative au réseau Internet.

ADRESSES MAC : L'ADRESSE RELATIVE A LA CARTE RESEAU

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 Vinc14-PC, junior0-PC, et The_frog-PC (au cas où vous vous demanderiez pourquoi ces noms, regardez la liste des auteurs du tuto). Si Vinc14-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.

Un paquet 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 **routage**, d'assurer que les données arrivent bien au destinataire. Le routage sera expliqué dès la prochaine partie.

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. L'élément central (switch, ...) se base donc sur les adresses MAC pour assurer la communication entre plusieurs machines appartenant à un même sous-réseau, mais utilise les adresses IP pour faire communiquer des machines de sous-réseaux différents.

On espère que vous avez compris. 😊

MASQUE DE SOUS-RESEAU ET PASSERELLE

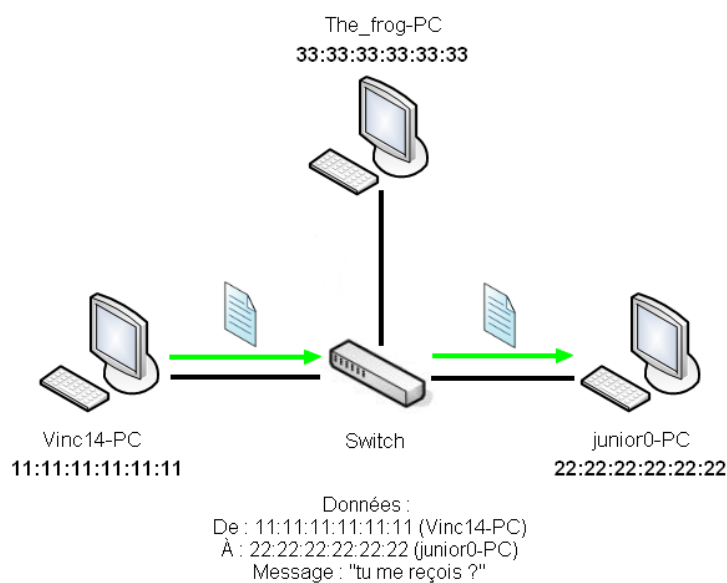
Afin de présenter ces notions, nous allons reprendre l'idée d'un "réseau" d'humains.

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 ZozorStreet, 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



Les réseaux de zéro - siteduzero.com

!

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

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 voit 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). 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 :

Pour que 2 hôtes (machines connectées) communiquent :

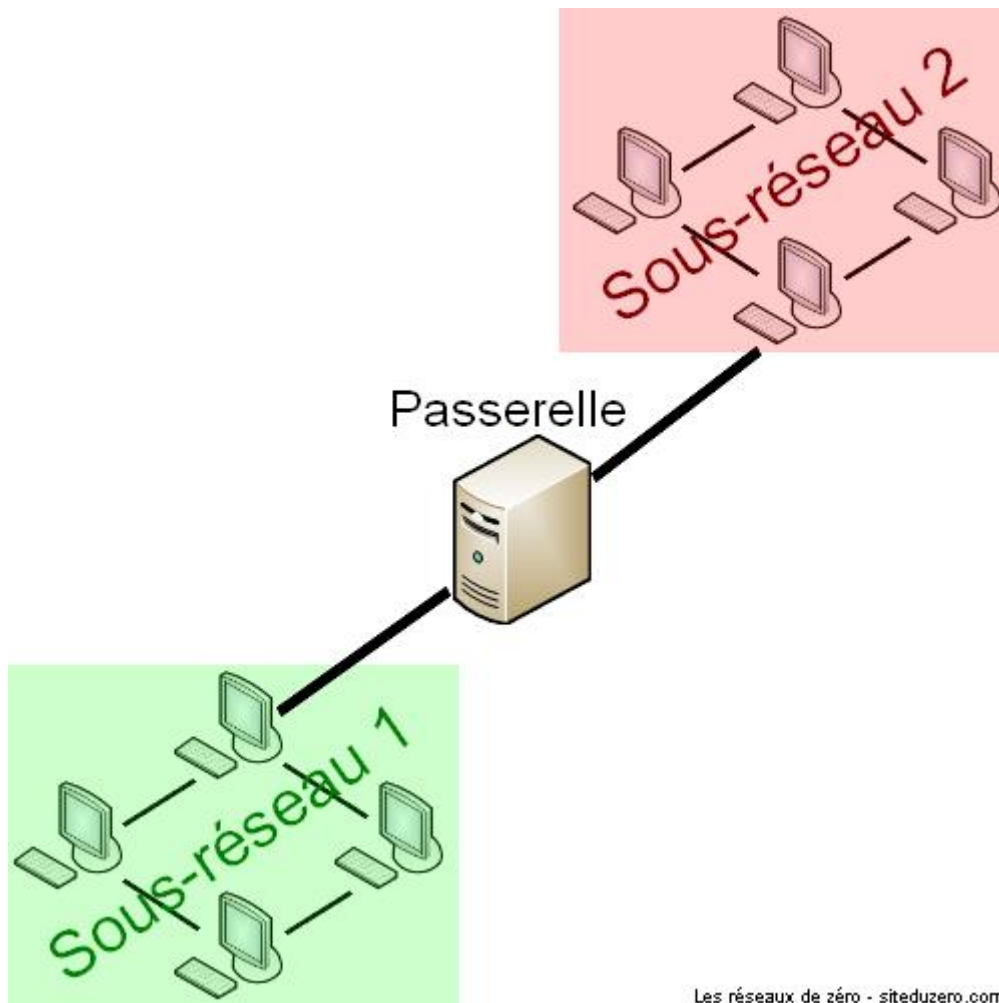
- Ils doivent utiliser le même protocole ;
- Ils doivent appartenir au même sous-réseau ;
- Chaque hôte doit connaître l'adresse IP de l'autre.

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

C'est là qu'intervient...

... LA PASSERELLE

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



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...

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**.

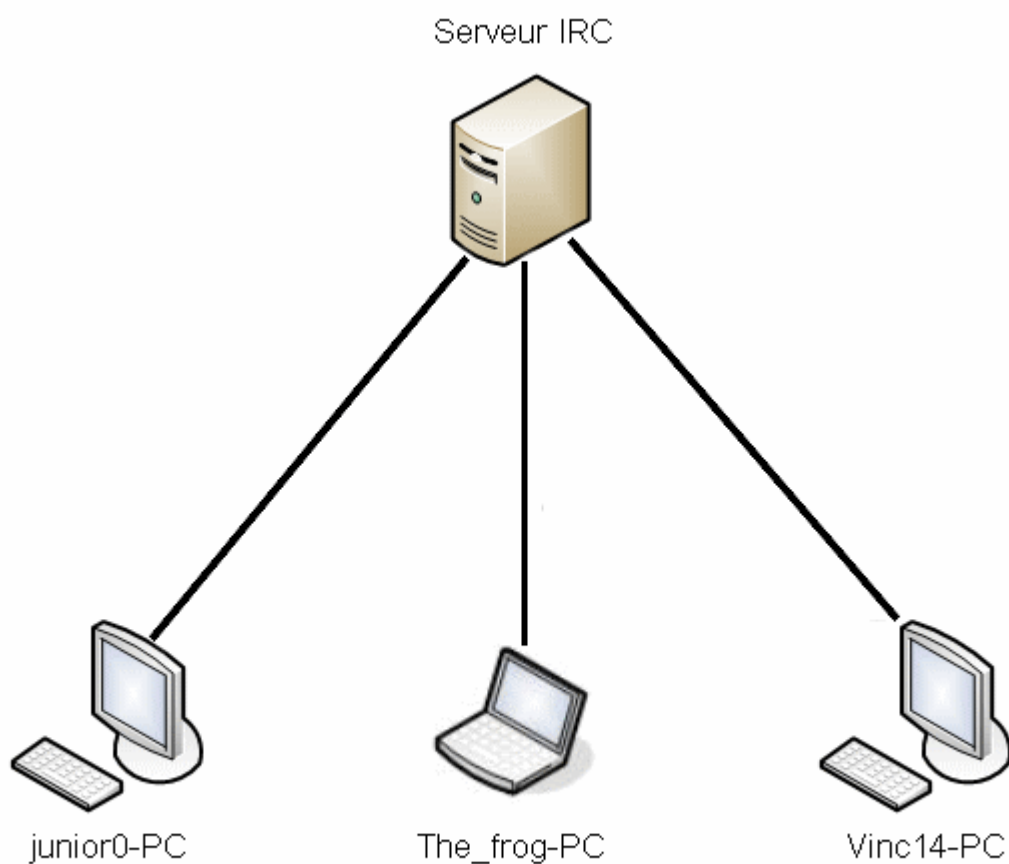
LE CLIENT ET LE SERVEUR

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 le Site du Zéro, 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 (hé 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 :



Les réseaux de zéro - siteduzero.com

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

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.

VI - LES MASQUES DE SOUS-RESEAUX

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.

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.

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.

Relation entre network ID et masques

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

129.51.3.5

255.255.0.0

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...

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**.

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 !

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.

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.

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.

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...

En effet. 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.

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 ?

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.

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.).

Analyse des contraintes

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

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, car 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. 🤔

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 ? ».

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. 🤔

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. 😊

L'organisation

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

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 :

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. 😊

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 🤔).

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. 😊

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 ! 😊)

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 !)

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. 😊

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.

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...

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**. 😊

LA PASSERELLE : LES BASES DU ROUTAGE

Deux hôtes ne se situant pas dans le même sous-réseau ne peuvent pas communiquer directement. Il faut que quelque chose intervienne entre les deux pour transmettre à l'un, les données au nom de l'autre. Ce « quelque chose » est la passerelle : un service qui connecte plusieurs sous-réseaux. Cette position fait donc que la passerelle se situe à califourchon entre plusieurs sous-réseaux, faisant ainsi office d'intermédiaire.

Dans un réseau comprenant plusieurs routeurs, la passerelle par défaut (default gateway, en anglais) est l'interface du routeur vers laquelle sont dirigés tous les **paquets** dont on ne connaît pas la route à emprunter pour atteindre le réseau dans lequel se trouve le destinataire. Chaque routeur a **une table de routage**. Pour faire simple, les paquets représentent des parties de vos données. En fait, lorsque vous envoyez des données sur un réseau, celles-ci sont découpées en plusieurs portions et chaque portion est appelée un paquet. **Quant à la table de routage, il s'agit d'une liste des différentes "routes" (chemins) vers d'autres sous-réseaux.** Ces définitions sont très simplifiées pour vous permettre de comprendre le principe, nous reviendrons dessus plus tard.

MODE DE FONCTIONNEMENT

Prenons un exemple concret pour illustrer le mode de fonctionnement d'une passerelle. Voici 2 ordinateurs : Azur-PC et Safran-PC (on se demande bien d'où ces noms sont inspirés). Leurs cartes réseau sont configurées ainsi :

Nom	Adresse IP	Masque de sous-réseau
Azur-PC	192.0.1.5	255.255.255.0
Safran-PC	72.40.2.1	255.0.0.0

Ils n'appartiennent pas au même sous-réseau et ne peuvent donc pas communiquer : il leur faut une passerelle. Voyons comment elle fonctionne.

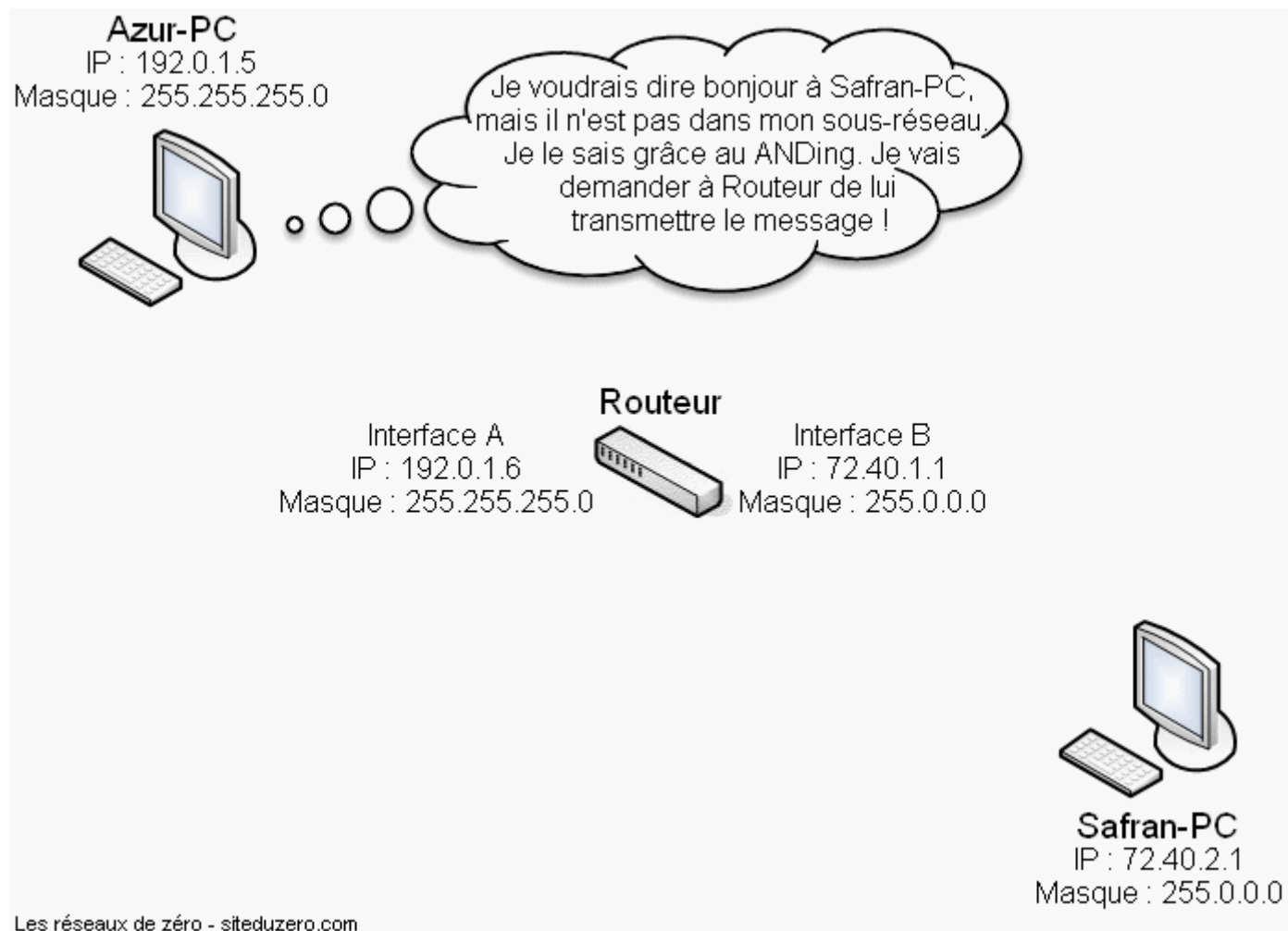
Azur-PC a recours à un processus nommé ANDing, que nous allons voir juste après, pour déterminer si Safran-PC, avec qui il veut communiquer, est dans le même sous-réseau que lui. Il réalise que ce n'est pas le cas, il va donc transférer son message à la passerelle en lui indiquant l'adresse du destinataire.

Supposons que ce soit un routeur qui offre ce service. Il a 2 interfaces. Pour que la communication puisse avoir lieu, une de ses interfaces doit être dans le même sous-réseau que Azur-PC et l'autre dans le même que Safran-PC. Voici une configuration possible pour ce routeur :

Interface	Adresse IP	Masque de sous-réseau
A	192.0.1.6	255.255.255.0
B	72.40.1.1	255.0.0.0

Avec une telle configuration, Azur-PC et Safran-PC peuvent à présent communiquer. Quand Azur-PC voudra parler à Safran-PC, il vérifiera grâce au ANDing si le destinataire est dans le même sous-réseau. Si oui, il enverra son message directement à son adresse IP, sinon, il l'envoie à la passerelle en lui demandant de transmettre à bon port. La passerelle étant entre les 2, cela ne pose pas de problème.

Voici un schéma récapitulatif :



Il est maintenant temps de voir ce qu'est le ANDing dont on a parlé sans expliquer ce que c'était.

ANDING (CONJONCTION LOGIQUE)

Dans la sous-partie précédente, nous avons évoqué un processus utilisé par Azur-PC pour déterminer dans quel sous-réseau se trouve le destinataire : le ANDing.

Le ANDing ou conjonction logique (ET logique) en français, est utilisé pour le [calcul des propositions](#). Nous n'allons pas entrer dans les détails de ce que c'est concrètement. Nous allons simplement voir en quoi cela peut nous servir dans les réseaux en général, mais surtout quelle est son application dans le cas du routage. Quand votre hôte veut communiquer, il fait un calcul logique pour déterminer si votre destinataire se trouve dans un même sous-réseau ou pas. Si le destinataire est dans un sous-réseau différent, les données seront envoyées à la passerelle qui se chargera de les *router* au destinataire.

Les règles du ANDing

C'est là que l'on voit l'intérêt d'être à l'aise avec la conversion des adresses IP du décimal au binaire. En effet, le ANDing se base sur la notation binaire des adresses IP. Si vous n'êtes pas à l'aise avec les conversions décimal/binaire, nous vous conseillons de relire [la sous-partie qui traite ce sujet](#). Le ANDing est relativement simple à comprendre, il faut juste assimiler les règles suivantes :

- $0 \text{ AND } 0 = 0$
- $0 \text{ AND } 1 = 0$
- $1 \text{ AND } 0 = 0$
- $1 \text{ AND } 1 = 1$

Déterminer si l'adresse IP du destinataire est dans le même sous-réseau que celle de l'émetteur est assez simple. La carte réseau de l'émetteur connaît son adresse IP, son masque de sous-réseau et l'adresse IP du destinataire. On va alors faire un ET logique (AND) entre l'adresse IP de l'émetteur et son masque de sous-réseau pour trouver son network ID. Ensuite, on va faire un ET logique entre l'adresse IP du destinataire et le masque de sous-réseau de l'émetteur et comparer le résultat avec le network ID obtenu précédemment. Si les deux valeurs sont identiques, alors l'émetteur et le destinataire sont dans le même sous-réseau. Sinon, ils sont dans des sous-réseaux différents.

Pas de panique, un exemple vaut mieux que tout ce pavé. 😊

Le ANDing par l'exemple

Paul veut communiquer avec Éric. L'ordinateur de Paul (Paul-PC) a pour adresse IP 142.20.1.15 et pour masque de sous-réseau 255.255.0.0. Celui d'Éric (Éric-PC) a pour adresse IP 92.40.1.14.

Étape 1 : déterminons le network ID de l'émetteur

Convertissons l'adresse IP de Paul-PC en binaire, ce que vous savez normalement faire. 😊

Voici ce que vous devez obtenir :

10001110.00010100.00000001.00001111

Convertissons à présent le masque de sous-réseau de l'adresse IP de Paul-PC en binaire :

11111111.11111111.00000000.00000000

Nous allons à présent faire un AND entre ces deux groupes de nombres binaires en appliquant les règles précédentes :

$$\begin{array}{r} 10001110 . 00010100 . 00000001 . 00001111 \\ \text{AND } 11111111 . 11111111 . 00000000 . 00000000 \\ \hline = 10001110 . 00010100 . 00000000 . 00000000 \end{array}$$

Le network ID de Paul-PC est donc 10001110.00010100.00000000.00000000.

Étape 2 : AND entre l'adresse IP du destinataire et le masque de sous-réseau de l'émetteur

Convertissons l'adresse IP d'Éric-PC en binaire :

01011100.00101000.00000001.00001110

Le masque de sous-réseau de Paul-PC ayant déjà été converti en binaire, il ne nous reste plus qu'à faire un ET logique :

$$\begin{array}{r} 01011100 . 00101000 . 00000001 . 00001110 \\ \text{AND } 11111111 . 11111111 . 00000000 . 00000000 \\ \hline = 01011100 . 00101000 . 00000000 . 00000000 \end{array}$$

On obtient donc 01011100.00101000.00000000.00000000.

Étape finale : comparaison des résultats

Au cours des deux étapes précédentes, nous avons obtenu :

```
Émetteur : 10001110.00010100.00000000.00000000
Destinataire : 01011100.00101000.00000000.00000000
```

Nous n'obtenons pas les mêmes valeurs. Par conséquent, ces deux adresses IP (142.20.1.15 et 92.40.1.14) ne sont pas dans le même sous-réseau.

Toujours à titre d'exemple, nous allons cette fois-ci choisir l'adresse IP du destinataire dans le même sous-réseau que celle de l'émetteur pour prouver que cette technique fonctionne bel et bien.

Si l'adresse IP d'Éric-PC est 142.20.20.10, sa notation en binaire sera **10001110.00010100.00010100.00001010**.

Nous avons déjà converti le masque de sous-réseau de l'émetteur en binaire donc nous pouvons passer directement au ET logique :

```
10001110 . 00010100 . 00010100 . 00001010
AND 11111111 . 11111111 . 00000000 . 00000000
= 10001110 . 00010100 . 00000000 . 00000000
```

Faisons une comparaison entre ce résultat et celui obtenu à l'étape 1 :

```
Émetteur : 10001110.00010100.00000000.00000000
Destinataire: 10001110.00010100.00000000.00000000
```

Comme vous pouvez le constater, le résultat est bien le même, donc ces deux adresses IP (142.20.1.15 et 142.20.20.10) sont dans le même sous-réseau. 😊

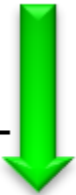
Voilà, vous comprenez maintenant comment ça se passe. Chaque fois que vous communiquez, ce calcul logique est effectué. Si le destinataire est dans le même sous-réseau, l'hôte lui transmet directement les paquets, sinon, il les envoie au routeur (la passerelle) qui se charge de les **router au destinataire**.

Un raccourci

La conversion du masque et de l'adresse IP en binaire n'est pas obligatoire, nous vous l'avons fait faire exprès pour vous faire pratiquer. 😊 En fait, il est tout à fait possible de faire un ET logique directement en décimal. L'astuce c'est de se servir d'une analogie : une adresse IP représente de l'eau qui coule en direction du masque de sous-réseau. Chaque 255 du masque représente une membrane perméable qui laisse couler l'eau et chaque 0 représente un barrage qui bloque l'eau. Ainsi seuls les octets d'une adresse IP au-dessus d'un 255 se retrouveront dans le résultat du ANDing.

Ce raccourci ne marche que si on utilise les masques de sous-réseaux par défaut, c'est-à-dire composé uniquement de 255 et de 0 (ce qui est le cas dans notre exemple). Mais, si vous devez faire un ET logique qui implique des masques de sous-réseau personnalisés, souvent représentés par une notation CIDR, cette technique ne marchera pas. 😊

Un schéma étant plus parlant qu'un discours, voici une illustration d'un ET logique entre l'adresse 192.168.30.4 et le masque 255.255.255.0 en utilisant le raccourci évoqué :

	192 . 168 . 30 . 4			Membrane perméable
AND	255 . 255 . 255 . 0			Barrage
	192 . 168 . 30 . 0			

Les réseaux de zéro - siteduzero.com

Remarquez que chaque octet de l'adresse IP au-dessus d'un 255 "coule", c'est-à-dire se retrouve dans le résultat de la conjonction logique. Par contre, l'octet qui se trouve au-dessus d'un zéro est bloqué. 🤨

Dans ce chapitre, nous avons fait un zoom sur la passerelle par défaut : nous avons étudié comment elle fonctionne, vu à quoi ça sert, et nous avons même étudié les opérations logiques qui s'effectuent au niveau de votre carte réseau. Tout ceci va vous servir de base pour l'étude de la fonction principale de la passerelle par défaut : le routage, que nous étudierons lorsque nous arriverons à la couche 3 du modèle OSI. Mais pour l'instant, nous sommes toujours coincés devant la couche 4 ! Courage, il ne nous reste plus que l'adressage par classes et l'adressage CIDR à voir...

VII - LES CLASSES DE RESEAUX

CLASSE A

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
```

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 ?

Et 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. 😊

Notez que 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.

A 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.

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. 🤔

Une adresse IPv4 est constituée de 32 bits séparés en 4 portions par des points. Donc, si vous voyez une adresse IP comme 120.0.2, ... Ben ce n'en est pas une. 🤔 Une adresse IPv4 est toujours composée de 4 blocs, pareil pour le masque de sous-réseau. 🤔

Chaque bloc contient 8 bits, soit un octet (ou byte en anglais). Ça, c'est à retenir par cœur car on va beaucoup utiliser ces termes : **8 bits = 1 octet = 1 byte**.

Schématiquement, ça donne ceci :

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

Les réseaux de zéro - siteduzero.com

Une adresse IP est donc constituée de 4 octets, ou 4 bytes soit 32 bits. Votre ordinateur, lui, il 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**.

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 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. 😊

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 😊).

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 😊

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. 😊

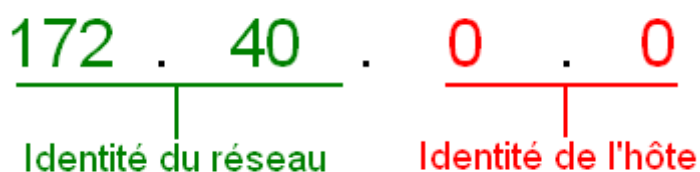
Mais pourquoi l'identité réseau prend deux octets dans ce cas ?

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 tous 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 - siteduzero.com

CLASSE C

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

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 sous-partie après. 😊

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...

LES ENVOIS DE DONNEES

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**. En bref, 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.

Retenez bien ce qu'est une adresse de broadcast, nous allons en avoir besoin dans cette partie !

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 quelles parties correspondent à l'identité réseau et quelles parties correspondent 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 :

Adresse IP : 194.220.34.1

Masque de sous-réseau par défaut : 255.255.255.0

Identité réseau : 194.220.34

Identité client : 1