I- Généralités sur les réseaux locaux

Qu’est-ce que le modèle OSI ?

Le modèle OSI (de l’anglais *Open System Interconnexion*) a été créé par l’organisation internationale de standardisation (*International Organization for Standardization* (ISO)) en tant que **modèle de référence pour une communication ouverte** via divers systèmes techniques. Ce programme prend tout son sens si l’on tient compte des débuts de l’Internet. En effet, à la fin des années 70, les principaux acteurs des nouvelles technologies ont été confrontés à la multitude de modèles existants de machines. Peu de constructeurs pensaient alors à leur mise en réseau et ne se souciaient guère de leur adaptabilité les uns des autres. Internet a entrainé la mise en place de standards de conformité entre ordinateurs dans le but de permettre une communication commune.

Le modèle OSI est le résultat d’une tentative de standardisation. Il dessine un cadre conceptuel pour la base de design des standards de communication entre différents ordinateurs. Le modèle ISO divise ce processus complexe de **communication en 7 couches** (de l’anglais *layer*). On parle alors de **modèle en couches**. La communication entre deux systèmes exige que chaque couche respecte une tâche. Nous pouvons citer par exemple le contrôle des communications, l’adressage du système cible ou la traduction de paquets de données en signaux physiques. Pourtant, ce modèle fonctionne seulement si tous les systèmes participant à la communication s’en tiennent aux règles. On constate que ces derniers se trouvent dans des protocoles qui sont configurés pour des couches précises ou qui interviennent entre les couches.

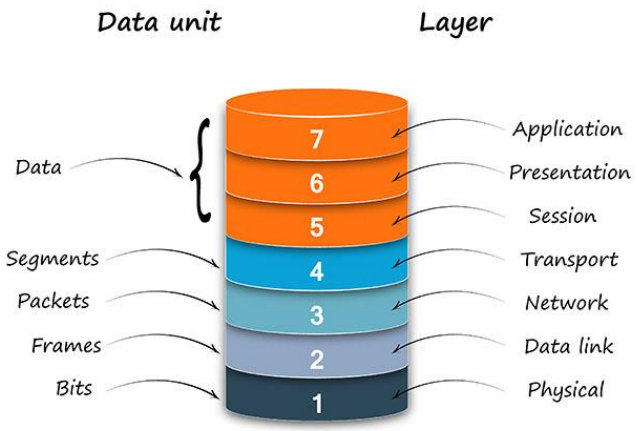
Néanmoins, le modèle de référence ISO n’est **pas une norme réseau concrète**. Au lieu de cela, il décrit sous forme abstraite quels processus doivent être régulés pour que la communication fonctionne via un réseau.

Les couches d'un modèle OSI

Il se peut que la communication entre deux ordinateurs s'avère triviale. En effet, lors d'un transfert de données via un réseau, il est normal que de nombreuses actions soient maîtrisées et qu'elles remplissent certaines exigences en termes de confiance, de sécurité voire d'intégrité. Cette pratique de diviser la communication en plusieurs couches a depuis fait ses preuves. Ainsi, chaque couche conduit à un domaine de fonctions bien définies. Une norme couvre en général seulement une partie du modèle en couches. Elle est construite de **manière hiérarchique**. En passant par une interface, chaque couche a recours à une autre située en dessous et est à son tour disponible pour les autres couches du niveau supérieur. Ce principe présente deux avantages décisifs :

- Les fonctions et les exigences qui doivent être maîtrisées remplies à l'intérieur d'une couche sont clairement définies. Les normes de chaque couche peuvent être développées indépendamment l'une de l'autre.
- Etant donné que les couches individuelles sont clairement séparées, les modifications apportées à une norme n'ont aucune influence sur les processus qui fonctionnent sur d'autres couches.

Au regard de leurs tâches, ces 7 couches du modèle OSI sont divisées en deux groupes : celles **orientées application** et celles **orientées communication**. Ces processus qui se produisent à chaque couche peuvent être illustrés par l'exemple suivant : le transfert d'un email partant d'un ordinateur vers le serveur de messagerie :



OSI model

Les couches orientées application

Les couches hautes du modèle de référence OSI sont décrites comme des couches orientées application. Elles présentent trois dénominations « application », « présentation » et « session ».

- **Couche 7 dite application :** ce domaine du modèle OSI dispose d'un contact direct avec les applications telles que les programmes de messagerie ou le navigateur. C'est là qu'ont lieu les entrées et les sorties des données. La couche application crée le lien avec les autres couches matérielles (ou basses) du modèle OSI et enclenche les fonctionnalités d'application. Pour illustrer notre propos, prenons l'exemple de transfert d'un email : un usager écrit un message dans le programme de messagerie de son ordinateur. Cet email est transféré sous la forme d'un paquet de données dans la couche application. En même temps, d'autres informations sont ajoutées aux données de l'email sous la forme d'un en-tête (ou header) d'application. On parle alors **d'encapsulation**. Cet en-tête contient entre autre l'information qu'il s'agit de données provenant d'un programme de messagerie. Par ailleurs, le protocole définit que le transfert de l'email est utilisé sur la couche application (dans le cas d'un email normalement SMTP).

- **Couche 6 dite présentation :** une des tâches courantes de la communication en réseau est de s'assurer que les données sont transmises sous format standard. La couche présentation présente des programmes locaux convertis en formats standards. Si nous prenons notre exemple, c'est la couche 6 qui définit la présentation du message transféré. En outre, le paquet de données est complété d'un en-tête de présentation. Celui-ci contient les informations sur le code de l'email, sur le format des pièces jointes et sur la manière dont les données sont comprimées voire chiffrées (par exemple SSL/TLS). C'est ainsi que l'on constate que le format d'un email est interprété par le système cible et qu'il est transmis en conséquence.

- **Couche 5 dite session** : la tâche centrale de la couche session est la gestion de la liaison entre les deux systèmes. On parle alors de couches orientées communication. C'est là que ces mécanismes de contrôle spécifiques prennent effet. Ce sont eux qui régulent la liaison, le maintien de la connexion et la déconnexion. En ce qui concerne cette **commande de communication**, d'autres informations sont ajoutées aux données de l'email transmis via un en-tête de session. La plupart des protocoles d'applications courants tels que SMTO ou FTP s'occupent eux-mêmes des sessions ou sont à l'instar de HTTP indépendants. Le modèle TCP/IP qui est concurrent au modèle OSI résume OSI 5, 6, 7 en une couche d'application. Les autres normes qui agissent sur la couche 5 sont NetBIOS, Socks et RPC.

Les couches orientées communication

Quatre couches orientées communication font suite aux trois couches orientées application du modèle OSI. Ces quatre couches sont les suivantes : « transport », « réseau », « liaison » et « physique ».

- **Couche 4 dite transport :** la couche transport sert de lien entre les couches orientées application et communication. A ce niveau du modèle OSI crée la connexion logique, soit **le canal de transmission** entre les systèmes communicants. Par ailleurs, certaines informations doivent être ajoutées aux données de l'email. Le paquet de données qui a déjà été ajouté à l'en-tête des couches orientées application est complété par un en-tête de transport sur la couche 4. C'est là qu'interviennent les protocoles de réseau standardisés tels que TCP. En outre, les ports sont définis sur la couche transport où les applications sont gérées sur le système cible. C'est sur la couche 4 que le paquet de données est attribué à application particulière.

- **Couche 3 dite réseau** : la couche 3 permet la transmission de données sur Internet et plus particulièrement l'adressage logique des terminaux. Une adresse IP unique est attribuée sur la couche 3. Un en-tête de réseau est ajouté au paquet de données qui contient des informations sur le routage et le contrôle des flux. Les systèmes informatiques ont recours aux standards de l'internet tels que IP, ICMP, X.25, RIP voire OSPF. En ce qui concerne les échanges d'emails, TCP

intervient en général avant IP.

- **Couche 2 dite liaison :** la couche liaison assure les fonctions telles que la détection des défauts, le dépannage et le contrôle des flux et a pour but **d'éviter des erreurs de transfert**. Par ailleurs, le paquet de données qui inclue les en-têtes d'application, de présentation, de session, de transport et de réseau est encadré par un en-tête et un parcours de liaison. Enfin, c'est sur la couche 2 qu'a lieu l'adressage physique.

- **Couche 1 dite physique :** c'est là qu'a lieu la conversion des bits d'un paquet de données en un signal physique approprié vers un support de transmission. Ce signal peut être transmis via un fil de cuivre, une fibre optique ou à l'air libre. L'interface de support de transmission est définie par des protocoles et des normes tels que DSL, ISDN, Bluetooth, USB ou Ethernet (tous deux des couches physiques).

Encapsulation et décapsulation

Les paquets de données passent à travers chaque couche du modèle OSI, sur le système émetteur et enfin le système cible. Tous les terminaux rencontrés sont configurés pour les couches allant de 1 à 3. L'email que notre exemple illustre passe en tant que signal physique d'abord par le routeur, avant de continuer sa route sur la Toile. Il est situé sur la couche 3 du modèle OSI et ne traite par conséquent que les informations des trois premières couches. Les couches allant de 4 à 7 ne sont pas prises en compte. Afin de recevoir un accès aux informations nécessaires, le routeur doit tout d'abord décompresser le paquet de donné encapsulé. On parle alors de **décapsulation**. Dans ce cas de figure, les couches du modèles OSI sont exécutées dans l'ordre inverse.

Tout d'abord, il faut décoder le signal sur la couche physique. Par la suite, il convient de lire les adresses MAC sur la couche 2 et l'adresse IP ainsi que le protocole de routage sur la couche 3. Avec ces informations, le routeur est en mesure de prendre la décision de transférer l'email. Le paquet de donnée peut ensuite être encapsulé et transmis en fonction des informations obtenues à la station suivante sur le chemin vers le système cible.

En règle générale, **plusieurs routeurs** sont impliqués dans un transfert de données. Le processus décrit de décapsulation et d'encapsulation expire jusqu'à ce que le paquet de données arrive sous la forme d'un signal physique auprès de la cible réelle (par exemple un serveur de messagerie). Le paquet de données est ici à nouveau décapsulé en parcourant les couches du modèle OSI de la couche 1 à 7. L'email envoyé via le client de messagerie arrive au serveur où il est prêt à être récupéré par un autre client.

Le jeu de protocoles *TCP/IP*

La famille de protocoles *Transmission Control Protocol* et *Internet Protocol* communément appelée pile de protocoles *TCP/IP* autorise l'échange de données en milieu hétérogène. Nous appelons milieu hétérogène un regroupement d'ordinateurs d'architectures ou des systèmes d'exploitation différents, par exemple des PC et des Apple Macintosh, des machines sous UNIX et de gros calculateurs.

IP est un protocole routable autorisant une communication en mode connecté au travers de *TCP*. Un fonctionnement en mode déconnecté est également possible. Est alors utilisé le protocole *UDP* qui fait partie de la pile des protocoles *TCP/IP*.

TCP/IP est actuellement un standard de l'industrie, évidemment en raison de son exploitation mondiale par Internet, mais aussi par son utilisation dans des réseaux d'entreprise de type Windows ou Novell.

Outre les protocoles de transport *IP* de la couche OSI 3 et *TCP* ou *UDP* de la couche OSI 4, le pile *TCP/IP* comporte des protocoles de niveau supérieur. Figurent, par exemple, le *Simple Mail Transfer Protocol* (*SMTP*), pour l'échange de messages électroniques, le *File Transfer Protocol* (*FTP*), pour le transfert de fichiers entre ordinateurs, le *Simple Network Management Protocol* (*SNMP*), de gestion de composants réseau tels les routeurs ou les répartiteurs, ou le *Hypertext Transfer Protocol* (*HTTP*), sur lequel repose le World Wide Web.

TCP/IP fut dans un premier temps réservé aux gros calculateurs et aux stations de travail, en raison de son importante pile de protocoles qui demandait une grande puissance de calcul. Les PC d'antan, sous MS-DOS, étaient inadaptés. Cela ne pose en revanche plus de problèmes aux PC de la génération actuelle et la plupart des ordinateurs communiquent donc sous *TCP/IP*.

TCP/IP est un jeu de protocoles dérivé du projet *ARPANet* (*Advanced Research Projects Agency*) du ministère de la Défense américain, pendant les années 60 et 70. Le but était de construire un réseau indestructible pouvant même résister à une frappe atomique. Après différentes étapes du développement auquel participèrent des militaires mais aussi des centres de recherche des universités, *ARPANet*, précurseur d'Internet, fut subdivisé en 1984 en deux sections, l'une pour la recherche et l'autre pour des applications militaires. Cette époque vit l'introduction d'une nouvelle famille de protocoles appelée jeu de protocoles *DARPA-Internet*, aujourd'hui connue sous le nom de *TCP/IP*.

Les caractéristiques intéressantes du jeu de protocoles *TCP/IP* sont :

- l'indépendance des fabricants, ce qui n'est pas le cas de tous les jeux de protocoles;
- presque tout système peut s'intégrer au réseau par *TCP/IP*;
- l'utilisable tant dans un *LAN* que dans un *WAN*;
- le fantastique essor d'Internet l'a élevé au rang de pile des protocoles la plus utilisée.

Les différents protocoles IP

Le protocole Internet propose les services de transmission en paquets de données. Ce type de transmission s'oppose à celui qui consiste à établir un flux continu de données et d'instructions de pilotage de flux. Les données à transmettre sont subdivisées en petits paquets qui sont déposés dans le réseau comme des messages que le destinataire décodera et assemblera de nouveau. Chacun de ces paquets peut emprunter un chemin différent pour parvenir au destinataire. *IP* fonctionne donc en mode non connecté et, tel quel, le protocole n'est pas fiable. Il n'existe aucun contrôle de flux et les paquets transmis sont considérés indépendamment les uns des autres. Une couche supérieure de protocoles doit donc assurer la sécurité des transmissions. Cela autorise le traitement individuel de chacun d'eux et leur transmission par le meilleur chemin existant à ce moment.

Bien qu'on évoque souvent *TCP/IP* comme s'il s'agissait d'une unique entité, il existe, outre *TCP*, d'autres protocoles qui reposent sur *IP*. Nous avons regroupé ces protocoles

dans un tableau et représenté leurs rapports à la figure 1, ainsi que leur emplacement dans le modèle *OSI*.

Modèle *OSI*	
Application	Couches application
Présentation	
Session	
Transport	
Réseau	Couches de flux de données
Liaison de données	
Physique	

Modèle *TCP/IP*	
Application	Protocoles d'application
Transport	Protocoles de transport
Réseau	Protocoles réseau
Accès réseau	

Figure 1 : Les protocoles *IP* et les couches *OSI*

Différents protocoles reposant sur *IP*	
UDP	*User Datagram Protocol* – Protocole de datagramme utilisateur. Paquet dont le destinataire n'accuse pas la réception; il est purement et simplement supprimé si le destinataire n'est pas joint. Ce protocole est de type non connecté, c'est-à-dire qu'expéditeur et destinataire ne sont pas reliés ensemble. Cela signifie qu'un problème de transmission n'est pas détecté au niveau du protocole. Cette détection et la solution sont à la charge de l'application exploitant le protocole. Ainsi, *TFTP* (*Trivial File Transfer Protocol*), *NFS* (*Network File System* sous UNIX) ou *SNMP* sont des exemples de telles applications.
ICMP	*Internet Control Message Protocol* – Protocole de messagerie Internet. Il assure l'échange de messages d'erreurs et de commandes entre passerelles et hôtes. Ces messages sont généralement générés et transmis par le logiciel réseau lui-même. Un utilitaire courant exploitant *ICMP* est *Tracert* (Windows) ou *Traceroute* (UNIX); ils permettent de suivre un message de routeur en routeur.
ARP	*Address Resolution Protocol* – Le protocole de résolution d'adresse transforme une adresse *IP* logique en une adresse physique. Cela n'est nécessaire que pour certains réseaux, par exemple *Ethernet* ou *token-ring*.
RARP	*Reverse Address Resolution Protocol* – Ce protocole de génération d'adresse transforme une adresse physique en adresse *IP* correspondante. Il est le symétrique d'*ARP* et n'est pas non plus exploité dans tous les réseaux.

Information

Passer du décimal au binaire

La façon la plus simple de convertir un nombre décimal en nombre binaire est de diviser répétitivement le nombre décimal par 2 et de retenir les résidus dans l'ordre indiqué.

Exemple : Conversion de 192 en binaire.

2	192	
2	96	+ 0
2	48	+ 0
2	24	+ 0
2	12	+ 0
2	6	+ 0
2	3	+ 0
	1	+ 1

192 (décimal) = 11000000 (binaire)

Passer du binaire au décimal

Le tableau suivant permet une conversion rapide entre ces systèmes :

Bit	7	6	5	4	3	2	1	0
Valeur	2^7=128	2^6=64	2^5=32	2^4=16	2^3=8	2^2=4	2^1=2	2^0=1

Dans la ligne du haut figure le rang du bit; la numérotation commence à 0. Dans le nombre binaire 00001010, les bits 1 et 3 sont à 1. Selon le tableau, il s'agit des valeurs 2 et 8 qu'il nous suffit d'additionner pour obtenir la valeur décimale, soit 10.

Information

Conversion du binaire à l'hexadécimal

Le système hexadécimal comprend les 16 symboles suivants :
(0, 1, 2,.....,9, A ,B ,C ,D ,E , F)
Chaque symbole est représenté dans le système binaire comme 4 bits.

Hexadécimal	Binaire
0	0000
1	0001
2	0010
3	0011
4	0110
5	0101
6	0110
7	0111

Hexadécimal	Binaire
8	1000
9	1001
A	1010
B	1011
C	1100
D	1101
E	1110
F	1111

Pour convertir un flux (*stream*) binaire de 8 bits en hexadécimal, il s'agit de diviser le flux en deux groupes de 4 bits et de trouver, dans le tableau sus-mentionné, le symbole hexadécimal correspondant. .

Exemple : Conversion de 11000000 en hexadécimal

11000000 = 1100 0000

C 0

192 (décimal) = 11000000 (binaire) = 0xC0 (hexadécimal)

Comparaison des modèles TCP/IP *et* OSI

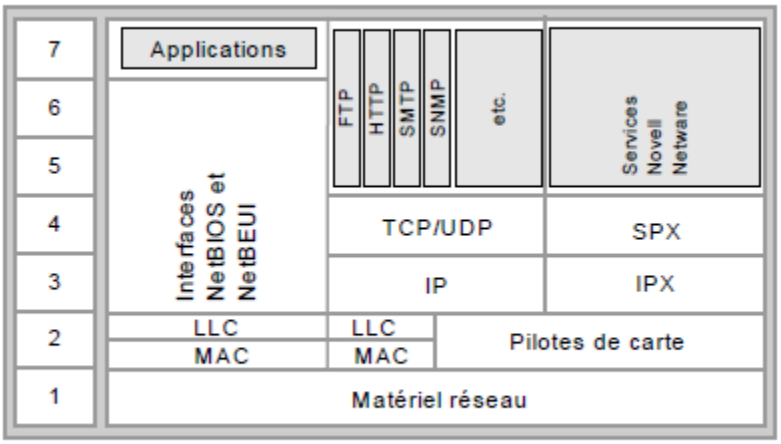


Figure 2 : Le modèle OSI appliqué aux protocoles

Les caractéristiques des protocoles d'application peuvent se résumer comme suit :

- Se situent au niveau des trois premières couches du modèle *OSI*.
- Permettent les interactions entre les applications et les échanges de données.
- *SMTP – FTP – SNMP – Telnet*.

Les protocoles de transport présentent les caractéristiques suivantes :

- Opèrent à la couche transport.
- Permettent les sessions de communication entre ordinateurs et le transfert fiable des données.
- *TCP – UDP – SPX – NWLink – NetBEUI*.

Les protocoles réseau se présentent de la façon suivante :

- Opèrent au niveau des trois dernières couches.
- Fournissent des services de liaison.
- Gèrent les informations de routage, d'adressage, de détection d'erreurs et des demandes de retransmission.
- Définissent des règles de communication au sein d'un environnement de réseau donné.
- *IP – IPX – NWLink – NetBEUI*.

L'adressage IP

Toute pile de protocoles identifie un ordinateur expéditeur et un ordinateur destinataire au moyen d'une adresse. Cette adresse fonctionne de la même manière qu'une adresse postale et permet d'identifier parfaitement un ordinateur précis au sein d'un réseau.

Une adresse *IP* est un code sur 32 bits généralement indiqué sous la forme de quatre nombres décimaux séparés par un point. Chacun de ces nombres correspond à un octet, autrement dit à 8 bits. Dans l'ordinateur, les valeurs sont traitées en tant que nombres binaires ou hexadécimaux par les programmeurs. Sous forme binaire, il s'agit d'une série de 32 uns et zéros, et en hexadécimal à 8 chiffres correspondant chacun à 4 bits.

L'utilisateur ne voit généralement que la représentation décimale mais nous verrons un peu plus loin que la représentation binaire facilite la compréhension des masques de sous-réseau.

La transposition de représentation des adresses peut s'expliquer par un exemple. Soit l'adresse 192.168.5.3, dont nous verrons qu'il s'agit d'une adresse dite de classe C. À noter que cette adresse est indiquée sous la forme de quatre nombres décimaux séparés par un point. La transposition se réalise comme dans la figure suivante.

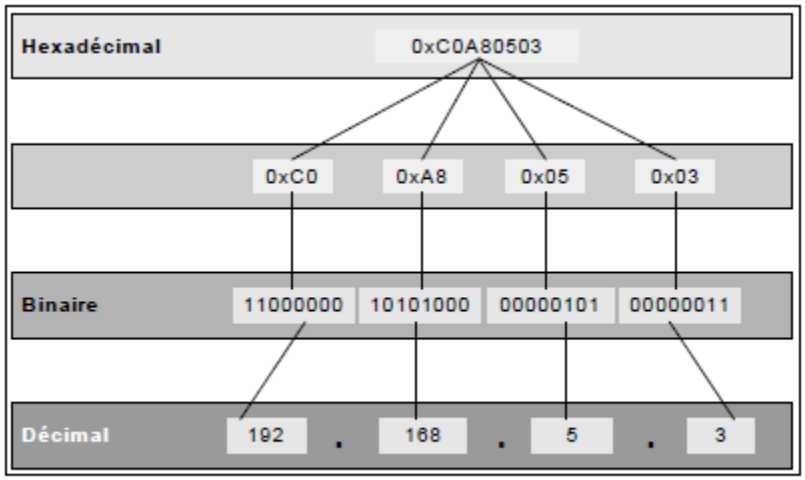
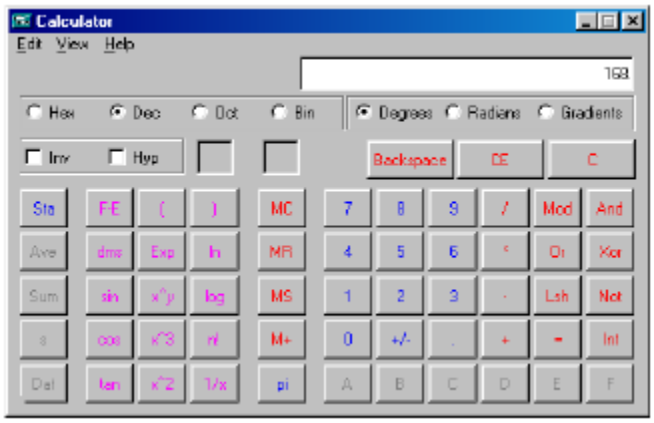


Figure 2 : Transposition d'une adresse *IP*

0xC0A80503 correspond à la suite d'octets 0xC0.0xA8.0x05.0x03, chacun d'eux étant séparé par un point du suivant. En exprimant ces octets en binaire, on obtient l'adresse *IP* telle qu'elle est traitée par la machine, soit 11000000101010000000010100000011. Pour faire soi-même une telle conversion, on doit employer une calculatrice scientifique, par exemple celle de Windows, qui autorise l'affichage d'un nombre selon différentes bases.



L'adressage dans un réseau exploite ce que l'on appelle un masque de sous-réseau. Ce masque permet à un ordinateur de déterminer si le paquet qu'il souhaite envoyer a pour destination le domaine auquel il appartient lui-même, ou s'il transitera par un routeur. Il orientera le paquet en conséquence.

Le masque de sous-réseau est également un nombre défini sur 32 bits mais qui se compose de deux blocs de chiffres binaires seulement. Le premier bloc ne comprend que des uns et celui qui le suit uniquement des zéros. Les uns définissent la partie réseau de l'adresse, et les zéros la partie hôte. Est ainsi également défini le nombre maximal d'hôtes dans un réseau donné.

Exemple de masque de sous-réseau : 11111111111111111111111100000000, ce qui correspond à 255.255.255.0 en décimal.

Conversion d'un masque binaire en décimal				
Masque en binaire	11111111	11111111	11111111	00000000
Masque en décimal	255	255	255	0

Tableau 2 : Conversion d'un masque binaire en décimal

Les classes d'adresses IP

Comme nous l'avons dit précédemment, dans une adresse IP, on distingue la partie réseau de la partie hôte. L'identification du réseau figure en début d'adresse et l'identificateur de l'ordinateur à la fin. Les premiers bits définissent la classe de l'adresse. Les combinaisons autorisées sont 0, 10, 110. La partie qui suit et qui identifie le réseau aura une longueur comprise entre 7 et 21 bits, cela en fonction de la classe. Enfin, la partie hôte aura une longueur de 8 à 24 bits, selon la longueur de la partie réseau précédente.

Il existe cinq classes d'adresses Internet, et nous n'en utilisons généralement que trois. Les classes A, B et C se distinguent par la longueur différente de leurs parties réseau et hôte. La classe D, spéciale, est réservée aux adresses dites de diffusion multipoint (un point vers plusieurs destinataires identifiés). Les premiers bits d'une adresse multipoint sont 1110.

Nous avons représenté, dans l'exemple ci-dessous, les formats possibles d'une adresse *IP*.

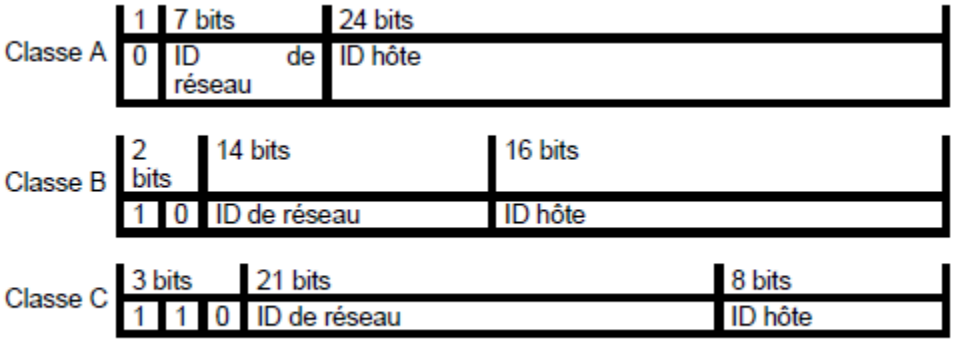


Figure 4 : Formats d'une adresse IP

La classe d'adresses A est destinée aux réseaux regroupant chacun de nombreux hôtes. Le premier bit de l'adresse est toujours un zéro. Le champ d'adressage théorique s'étend ainsi de 0.0.0.0 à 127.255.255.255. Toutefois, un octet d'une adresse ne peut se composer ni uniquement de zéros (00000000), ni uniquement de uns (11111111). Les adresses correspondantes n'existent donc pas. Cela s'applique tant à la partie réseau de l'adressage qu'à la partie hôte. Les réseaux 0.0.0.0 et 127.0.0.0 sont réservés à titre de routage par défaut (*default route*) et d'adresse de bouclage (*loopback address*) respectivement. Un réseau de classe A fournit ainsi 126 adresses réseau dans la plage 1.0.0.1 à 126.255.255.254. Les 24 autres bits de l'adresse constitueront la partie hôte, ordinateur, de l'adresse. Chaque réseau peut ainsi comporter jusqu'à 16 777 214 (2^{24}-2) ordinateurs.

Dans le cas des réseaux de classe B, 14 bits désignent le réseau et les 16 bits suivants, l'hôte. Une adresse de classe B débute toujours par 10, ce qui nous donne un champ d'adressage théorique de 128.0.0.0 à 191.255.255.255. Concrètement, seules les adresses 128.0.0.1 à 191.255.255.254 sont exploitables. Cela correspond à 16 834 (2^{14}) réseaux regroupant chacun 65 534 (2^{16}-2) ordinateurs.

La classe C définit le grand nombre de réseaux, trois octets étant attribués à leur adressage. Seuls 8 bits demeurent pour désigner les hôtes, ce qui correspond à 254 (2^8- 2) machines au sein de 2 097 152 (2^{21}) réseaux. Un tel adressage se distingue par ses trois premiers bits, dont la valeur est 110. La plage des adresses des réseaux de classe C débute à 192.0.0.1 et se termine à 223.255.225.254.

Le réseau de classe D est d'un usage spécial. Les paquets destinés à ce réseau sont reçus par tout hôte inscrit à cet effet. Le travail est effectué par le routeur le plus proche qui doit disposer d'une liste de destinataires et qui dupliquera et routera les paquets en conséquence. Si le routeur n'est pas dans le chemin de distribution, il tente de s'inscrire auprès du routeur suivant. Le processus est répété de routeur en routeur jusqu'à ce que tous les routeurs entre les expéditeurs et les destinataires aient transmis le paquet. Le fait que ce ne soit pas l'expéditeur mais les routeurs concernés qui dupliquent au besoin les paquets minimise la charge du réseau. Ce procédé est particulièrement bien adapté à la diffusion de contenus multimédias.

Chaque paquet *IP* contient l'adresse 32 bits de l'expéditeur et du destinataire. L'identificateur de classe indique le nombre de bits constituant l'adresse du réseau. Un routeur peut extraire son adresse mais aussi celle d'autres réseaux et router le paquet.

De nombreux espaces d'adressage sont réservés, notamment un par classe d'adresses. Les adresses de cette plage ne peuvent s'employer dans Internet car elles sont réservées aux réseaux locaux et privés. Cela garantit une frontière étanche et empêche le routage, vers Internet, de paquets à destination locale. La communication d'un tel réseau avec Internet s'effectuera à travers un routeur spécial, par la technique de l'*IP masquerading*. La méthode consiste à exploiter une adresse *IP* routable unique par la totalité des machines du réseau local. Elle est employée conjointement dans le cas de connexions intermittentes et s'utilise donc par le biais d'un classique fournisseur d'accès. Nous avons regroupé les plages d'adresses réservées dans le tableau suivant.

Plages d'adresses attribuées aux réseaux « privés »			
10.0.0.0	à	10.255.255.255	Réseau classe A
172.16.0.0	à	172.31.255.255	Réseau classe B
192.168.0.0	à	192.168.255.255	Réseau classe C

Tableau 3 : Plages d'adresses destinées aux réseaux privés

Notez que la première adresse d'un réseau le représente dans sa totalité. Ainsi, l'adresse 192.168.0.0, qui dépend du masque de sous -réseau, comme nous le verrons plus loin, ne peut pas être attribuée à un hôte. La dernière adresse de la plage, également dépendante du masque de sous-réseau, ne peut pas non plus être attribuée à une machine puisqu'il s'agit de l'adresse de diffusion générale (*broadcast*).

*Les sous-réseaux (***subnetting***)*

Un sous-réseau est une plage d'adresses *IP* d'une même adresse réseau. Ces sous-réseaux peuvent être réunis par des routeurs et former un réseau plus vaste. La plage d'adresses attribuée ne pouvant s'exploiter sans une structuration en un réseau de plusieurs sous-réseaux, il est nécessaire de réaliser une répartition ordonnée de cette plage. Cette subdivision s'appelle le sous-réseautage.

Le sous-réseautage, pourquoi?

Un réseau sera articulé en plusieurs ensembles logiques de façon à en équilibrer la charge. Cette articulation est souvent calquée sur l'organisation géographique des machines, leur répartition en étages ou en bâtiments, ou sur l'organisation de l'entreprise en services et en fonctions. Il convient de veiller au nombre d'ordinateurs gérés dans un sous-réseau puisque le but est également un équilibrage de la charge.

La subdivision du réseau en unités indépendantes réunies par les routeurs peut décharger les sections importantes pour l'entreprise, telles que la production ou le centre de calcul, et fiabiliser le fonctionnement.

Une entreprise aux nombreuses filiales peut recevoir, par exemple, une adresse unique de classe B que le gestionnaire de réseau découpera de manière appropriée au sein de la société. Selon cet exemple, il dispose de 16 bits pour l'adressage des hôtes. Si les services sont reliés par des routeurs, les données n'ont à transiter par eux que lorsque cela est réellement nécessaire.

Dans le cas contraire, la charge de la totalité du réseau pourrait croître de manière inacceptable en raison de transports inutiles. L'utilisation de routeurs ne se justifierait également plus. Cela ne peut être assuré que lorsque le routeur distingue le segment physique auquel les données sont adressées. Pour lui permettre la prise de décision, chaque segment recevra une adresse de sous-réseau propre qui se transcrira en masque de sous-réseau. Comme nous l'avons évoqué, le masque de sous-réseau sert à distinguer la partie de l'adresse de la partie hôte.

Il est possible de subdiviser la plage attribuée en adresses de réseaux, en adresses de machines. Dans le cas d'adresses de classe A ou B, cette subdivision est souvent, dans un but de simplification, réalisée à la limite d'un octet. Ainsi, un réseau de classe B peut employer le troisième octet en tant qu'identificateur de sous-réseau et le quatrième pour désigner les machines. Cela se transcrirait par 254 réseaux de 254 machines chacun.

Toutefois, avant de procéder à cette répartition, il faut réfléchir aux besoins des ordinateurs et des sous- réseaux, en fonction de leur nombre et de leur charge. Il est possible qu'une section du réseau requière plus de 254 stations ou qu'un nombre restreint d'ordinateurs génère un trafic intense qu'il conviendrait de circonscrire dans un petit sous-réseau.

De même, il peut se révéler nécessaire de subdiviser un réseau de classe C bien que nous n'y distinguons pas les frontières naturelles et claires d'un réseau de classe B. Il est donc possible de subdiviser un réseau au sein d'un octet.

Selon l'exemple de la classe C, nous disposons de 8 bits pour l'adresse de l'hôte. Deux méthodes nous permettent de raccorder 60 hôtes répartis dans 4 segments physiques du réseau :

- Sans sous-réseau : nous ignorons la structure physique et attribuons sans plan les adresses aux hôtes. La totalité des routeurs doit connaître les adresses qui correspondent aux différents segments de façon à pouvoir effectuer le routage ou à se contenter de faire suivre tous les paquets, ce qui remet en cause leur intérêt. Lorsqu'on ajoute un hôte, il faut mettre à jour la totalité des tables de routage, ce qui correspond à un travail de maintenance important.

- Avec sous-réseau : nous réservons 2 bits des 8 bits d'adressage d'hôte pour identifier les sous-réseaux. Ces 2 bits permettent de définir 2^2 = 4 sous-réseaux. Les 6 bits restants désigneront les hôtes de chaque sous-réseau, soit 2^6=64 ordinateurs. Les 2 bits internes d'identification de sous-réseau pourront être utilisés par les routeurs pour prendre d'autres décisions. Aucune modification des tables de routage n'est nécessaire en cas d'ajout d'un ordinateur tant que les bits réservés à cet adressage suffisent pour le nombre d'ordinateurs installés dans le sous-réseau.

L'exemple suivant représente la répartition d'un réseau de classe B en plusieurs sous-réseaux de classe C.

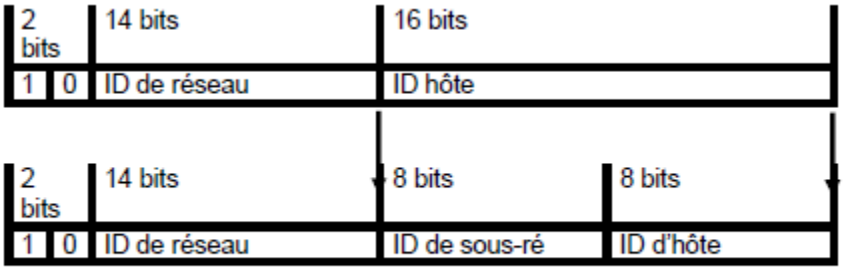


Figure 5 : Subdivision d'un espace d'adressage de classe B en sous-réseaux de classe C

Une adresse internet correspond ainsi à une structure de trois niveaux :

- identificateur du réseau;
- identificateur de sous-réseau;
- identificateur de l'hôte dans le sous-réseau.

Calcul des adresses réseau

Lorsque l'on sait quelle classe d'adresses doit être subdivisée et le nombre d'ordinateurs et de sous-réseaux qui doivent être adressés, l'étape suivante consiste à définir le masque de sous-réseau. Le but de ce masque est de permettre à un ordinateur de déterminer les paquets *IP* qu'il peut émettre directement dans le réseau. Les paquets dont le destinataire n'est pas dans le sous-réseau doivent être transmis à un routeur c'est-à-dire changer de sous-réseau. Ce masque déterminera également la taille du sous-réseau. Examinons pour cela la structure d'une adresse *IP*.

L'adresse réseau écrite sous la forme de 4 valeurs décimales correspond à une unique valeur de 32 bits contenant l'identificateur du réseau et celui de l'hôte. Le masque de sous-réseau distingue les bits désignant le réseau de ceux désignant le sous-réseau. Le masque de sous-réseau comporte donc également 32 bits, exprimés aussi sous la forme de 4 nombres décimaux. Un *ET* logique (opération booléenne) opéré entre les adresses cible et source et le masque de sous-réseau indique si la destination du paquet figure dans le même sous-réseau. Chaque bit de l'adresse est comparé au bit correspondant du masque. Si le bit du masque est à 1, le bit correspondant de l'adresse est transmit au résultat. Si le bit est à 0, le bit correspondant de l'adresse est également à 0 dans le résultat. Le résultat de l'opération correspond à la partie réseau de l'adresse.

Information

La fonction booléenne ET

Le tableau suivant montre l'opération de la fonction booléenne ET sur les différentes combinaisons possibles avec 2 bits:

Bit de l'adresse	Bit du masque de sous-réseau	Résultat
0	0	0 ET 0 = 0
0	1	0 ET 1 = 0
1	0	1 ET 0 = 0
1	1	1 ET 1 = 1

Nous avons regroupé quelques exemples d'adresses dans le tableau suivant.

Calcul de l'adresse réseau				
	Octet 1	Octet 2	Octet 3	Octet 4
Adresse source	192	170	11	32
(binaire)	11000000	10101010	00001011	00100000
Adresse cible	192	170	1	33
(binaire)	11000000	10101010	00000001	00100001
Masque de sous-réseau	255	255	255	0
(binaire)	11111111	11111111	11111111	00000000
Résultat pour l'adresse source :	192	170	11	0
(binaire)	11000000	10101010	00001011	00000000
Résultant pour l'adresse cible :	192	170	1	0
(binaire)	11000000	10101010	00000001	00000000

Tableau 6 : Calcul de l'adresse réseau

Les bits désignant l'hôte sont ainsi mis à zéro et ceux identifiant le réseau sont conservés.

Le résultat de l'opération appliquée aux adresses cible et source est ensuite comparé. Si les deux adresses de réseau sont semblables, le destinataire fait partie du même sous-réseau. Dans le cas contraire, le paquet est transmis à la passerelle par défaut, c'est-à-dire le routeur. Dans l'exemple du tableau 5, il est clair que réseau source (192.170.11.0) et réseau cible (192.170.1.0) sont différents et que les données doivent être routées.

Si on utilise les adresses préconisées par le *NIC* sans créer de sous- réseaux, il est possible de reprendre sans modification les masques de sous-réseau cités dans le tableau 6.

Masques de sous-réseau correspondant aux adresses préconisées par le *NIC*			
Espace d'adressage *IP*	Masque de sous-réseau	Nombre maximal de réseaux	Nombre maximal d'ordinateurs par réseau
Réseau classe A	255.0.0.0	126	16 777 21
Réseau classe B	255.255.0.0	16 384	65 534
Réseau classe C	255.255.255.0	2 097 152	254

Tableau 7 : Masque d'un sous-réseau

Masques de sous-réseau d'un réseau classe C			
Nombre de sous-réseaux	Masque de sous-réseau	Nombre maximal d'ordinateurs par réseau	Masque de sous-réseau en binaire
2	255.255.255.128	126	11111111.11111111.11111111.10000000
4	255.255.255.192	62	11111111.11111111.11111111.11000000
8	255.255.255.224	30	11111111.11111111.11111111.11100000
16	255.255.255.240	14	11111111.11111111.11111111.11110000
32	255.255.255.248	6	11111111.11111111.11111111.11111000
64	255.255.255.252	2	11111111.11111111.11111111.11111100

Tableau 8 : Exemple de masque de sous-réseau

Le tableau 8 illustre les masques de sous-réseau de classe C. Les sous-réseaux se composent des parties suivantes :

Masque de sous-réseau de la classe d'adresses;

Masque subdivisant la classe d'adresses en sous-réseaux.

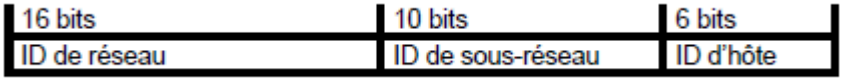
Le masque de répartition en classes d'adresses correspond au nombre de bits utilisés pour le masque de sous-réseau. Comment celui-ci est-il calculé?

Nous devons d'abord déterminer le nombre d'adresses hôtes nécessaires dans le sous-réseau. Si nous disposons, par exemple, d'un réseau de classe C et que nous ayons besoin au plus de 20 adresses d'ordinateur par sous-réseau, alors, selon le tableau 7, nous pouvons effectuer une répartition dans 8 sous-réseaux au plus. Cela nous donne 256/8 = 32 adresses *IP* par réseau, dont la première sera réservée à l'identification du sous-réseau et la dernière à la diffusion générale. Nous disposons donc en fait de 30 adresses *IP*.

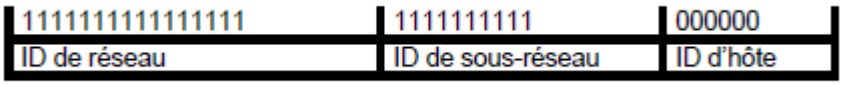
Malheureusement, plus le réseau est petit, plus la perte d'adresses augmente. Une subdivision en 64 sous- réseaux ne nous permettrait d'adresser que 64 * 2 = 128 ordinateurs. Le nombre d'ordinateurs par sous-réseau étant déterminé, nous en déduisons le nombre de bits attribués à l'identification du sous-réseau en soustrayant les bits utilisés de ceux disponibles.

Exemple :

Un réseau de classe B fournit 16 bits pour l'identification de l'hôte. Si nous n'utilisons que 6 bits pour identifier les hôtes, nous disposons de 10 bits pour identifier les sous-réseaux. Cela nous donne la répartition suivante :



Les bits du masque qui sont utilisés pour l'identification des hôtes doivent être à 0 alors que les bits qui ne sont pas employés pour l'identification des hôtes doivent être à 1. Le masque de sous-réseau aura donc la forme :

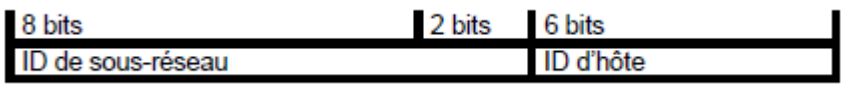


correspondant à 11111111.11111111.11111111.11000000 en binaire. La conversion en décimal donne le masque 255.255.255.192.

Le réseau de classe B est subdivisé en 2^{10} = 1 024 sous -réseaux puisque 10 bits sont réservés au masque de sous -réseau. L'identificateur des hôtes se définit sur 6 bits, ce qui fournit 2^6 adresses hôtes pour chaque sous -réseau. Leur plage s'étend de 1 à 62, soit 64 adresses au total, dont nous déduisons la première et la dernière adresse qui sont réservées. Les deux bits de poids fort du dernier octet font partie de l'identificateur de sous-réseau.

En admettant que l'entreprise ait reçu du *NIC* l'adresse de classe B 129.1.0.0, il s'agit maintenant d'obtenir l'adresse des 1 024 sous-réseaux et les adresses hôtes dans chaque sous-réseau.

Étant donné que les 16 premiers bits sont invariables dans une adresse de classe B, on va se concentrer sur les 16 derniers bits dont non dispose, soit 10 bits pour identifier les sous-réseaux et 6 bits pour identifier les hôtes dans chaque sous réseau.



Les sous-réseaux possibles donc varient de 0000000000 à 1111111111. Prenons comme exemple le sous-réseau 0000000000. Les 6 bits réservés aux hôtes peuvent, eux, varier entre 000001 et 111110.

00000000	00	000001 à 111110
ID de sous-réseau		ID d'hôte

Pour ce qui est du sous-réseau 0000000000, nous avons donc la plage d'adresses suivante :

- 00000000.00000001 à 00000000.00111110 (binaire),
- 0,1 à 0,62 (décimal).

Le sous-réseau 0000000000 propose la plage d'adresses effectivement exploitables 129.1.0.1 à 129.1.0.62.

Le même exercice peut être répété pour les sous-réseaux allant de 0000000001 à 1111111111. La plage complète d'adresses effectivement exploitables serait alors de 129.1.0.1 à 129.1.255.254.

Information
Le supernetting
Le masque de sous-réseau permet, à l'inverse du *subnetting*, de regrouper plusieurs réseaux de classe C jointifs en un réseau de classe B. Ce procédé s'appelle le *supernetting* et ne présente un intérêt que dans des cas très spéciaux. Par exemple, on pourrait combiner 8 réseaux de classe C jointifs, soit 202.61.0.0 à 202.61.7.0 en un seul réseau de 2^{11} - 2 = 2 046 hôtes en utilisant un masque de sous-réseau de 255.255.248.0 (décimal) ou 11111111.11111111.11111000.00000000 (binaire). Dans ce cas, 11 bits sont réservés pour identifier les hôtes; et le masque de sous-réseau comprend 21 bits.

Exemple de sous-réseau

Illustrons ces explications par l'exemple d'une entreprise de taille moyenne ayant des locaux sur deux sites.

Dans un réseau de classe C physiquement divisé en deux, il existe deux routeurs, un dans chaque partie, qui, dans un domaine, constituent la passerelle par défaut. Ils sont reliés et unissent, par conséquent, les deux parties du réseau. Cent ordinateurs au plus sont raccordés à chacun des sous-réseaux. Se posent alors les questions suivantes :

- À quoi ressemble le masque de sous-réseau des parties de réseau qui sont, rappelons-le, physiquement isolées?
- Quelles adresses pouvons-nous attribuer au sein des sous-réseaux?

Il s'agit ici d'une division en deux du réseau, chaque partie pouvant recevoir au plus 126 machines. La passerelle par défaut, c'est-à-dire le routeur, demandera une adresse dans chaque sous-réseau, par laquelle il est joignable à partir de chacun d'eux. L'identificateur de sous-réseau se composera donc d'un seul bit. En incluant les 24 bits du masque de réseau, le masque de sous-réseau comprend 25 bits.

24 bits	1 bit	7 bits
ID de réseau	ID de sous-réseau	ID d'hôte

Le sous- réseau 0 fournit la plage d'adresses 00000001 à 01111110, soit de 1 à 126 en décimal.

Le sous-réseau 1 fournit la plage d'adresses 10000001 à 11111110, soit de 129 à 254 en décimal.

Le masque de sous-réseau s'écrit en binaire : 11111111.11111111.11111111.10000000, soit 255.255.255.128 en décimal.