LES PROTOCOLES DES RÉSEAUX

Introduction

Afin d'échanger des données de manière structurée au sein d'un réseau, il faut avoir recours à des règles qui commandent le déroulement des communications : les protocoles.

Le choix d'un protocole pour un réseau est dicté par l'environnement. Il faut prévoir la possibilité d'une extension, évaluer la taille que le réseau peut atteindre, cibler les ordinateurs à relier et, au préalable, déterminer le but de la mise en réseau.

On distingue généralement deux grands types de protocoles : les protocoles routables et les protocoles non routables. Dans les sections suivantes, nous allons étudier ces deux types de protocoles. Ensuite, nous examinerons les protocoles routables les plus utilisés sur les réseaux locaux – *TCP/IP*, *IPX/SPX.* Pour commencer, nous allons citer quelques caractéristiques du jeu de protocoles *TCP/IP*. Nous évoquerons les différents protocoles reposant sur *IP* et nous traiterons les bases théoriques de l'adressage *IP* et son articulation en sous-réseaux. Nous présenterons ensuite les protocoles non routables tels que *NetBIOS* (*Network Basic Input/Output System*) et *NetBEUI* (*NetBIOS Extended User Interface*).

Protocole routable et protocole non routable

Un protocole routable peut transmettre ses paquets de données à un routeur. Ce dernier doit évidemment gérer ce protocole. Les protocoles routables couramment employés sont *IP* (*Internet Protocol*) et *IPX* (*Internet Packet Exchange*). En revanche, *NetBEUI*, contemporain d'anciens produits Microsoft comme MS-DOS et Windows pour Workgroup, est un exemple de protocole non routable. Il est encore le protocole par défaut des réseaux Microsoft sous Windows 95 et 98.

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*.

Tableau comparatif de protocoles routables et non routables

NetBEUI (Microsoft)	Non routable	Facile à configurer. Fonctionne par diffusion (*broadcast*).
NWLink IPX/SPX Protocole Microsoft compatible à celui de Novell *IPX/SPX*	Routable	Pour cohabitation avec Novell Netware.
IPX/SPX (Novell)	Routable	Utilisé principalement dans les réseaux Novell (version 4 et inférieure).
TCP/IP	Routable	Très largement répandu. Protocole d'Internet.

Tableau 1 : Tableau comparatif des protocoles routables et non routables

Les caractéristiques du jeu de protocoles 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**

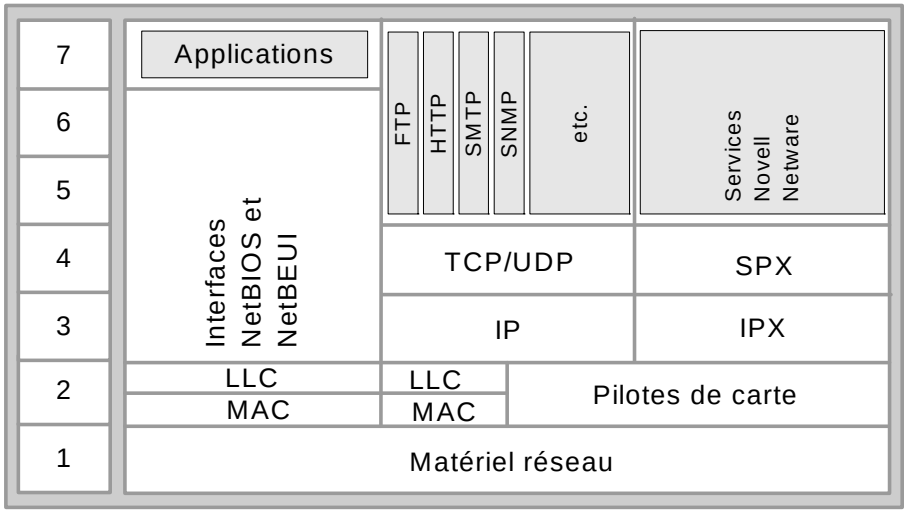


Figure 2 : Le modèle OSI appliqué aux protocoles

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.

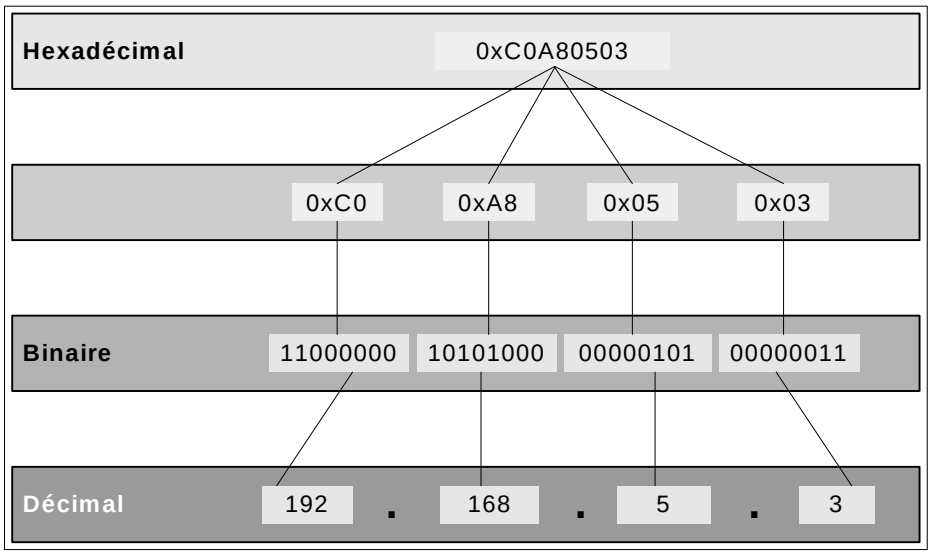
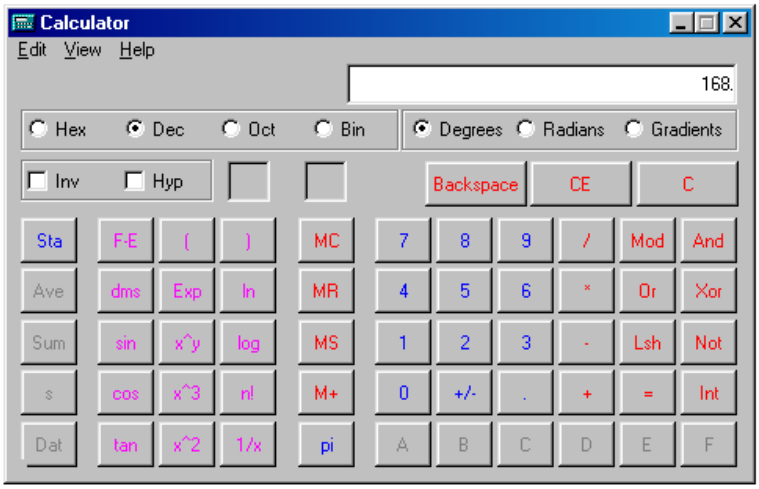


Figure 2 : Transposition d'une adresse *IP*

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.



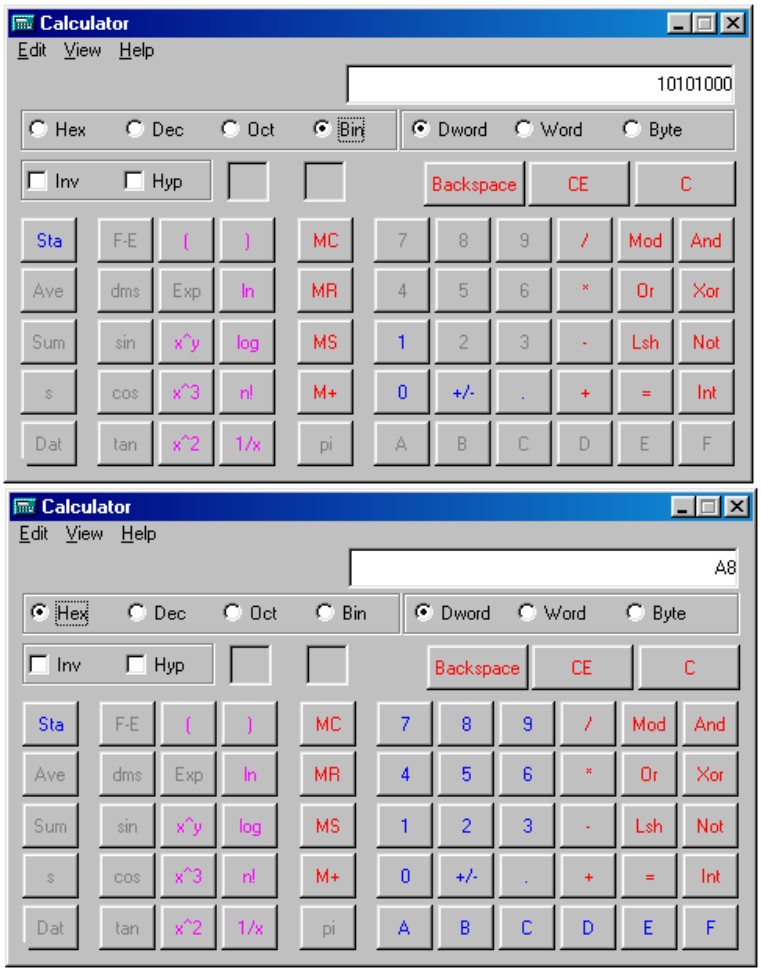


Figure 3 :
Conversion du nombre 168 en binaire et en hexadécimal

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*.

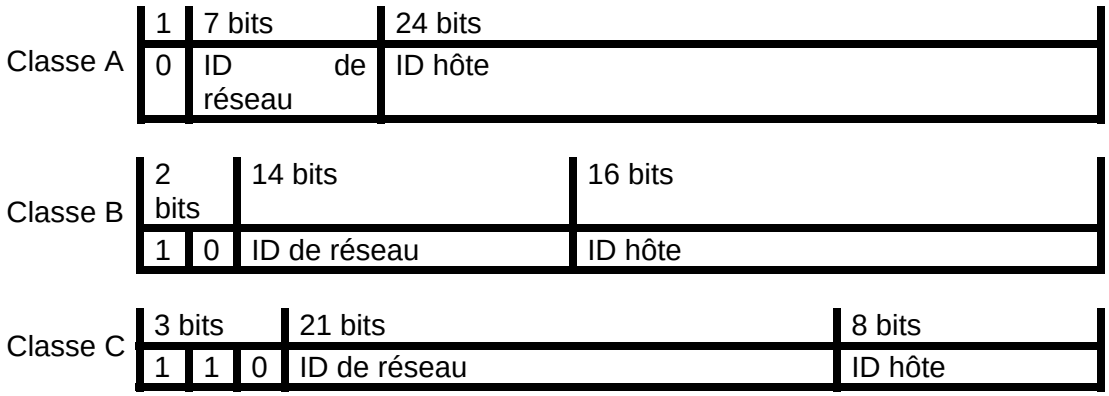


Figure 4 : Formats d'une adresse IP

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 adresses *IP* sont attribuées de manière centralisée par le *Network Information Center* (*NIC*). Un réseau connecté en permanence à Internet doit donc recevoir une plage d'adresses attribuées par le *NIC*. La demande est généralement effectuée par le fournisseur d'accès qui attribue la ligne fixe. Il est alors possible de n'acheter que des parties de réseaux de classe C.

Tout hôte d'Internet se devant de posséder au moins une adresse unique, mais pouvant en posséder plusieurs, et les adresses ayant été attribuées, aux débuts d'Internet, de façon très libérale, il n'existe actuellement plus de réseaux disponibles en classe A ou B.

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 par le *NIC* 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. Le *NIC* tente d'attribuer des classes d'adresses appropriées, mais cet organisme ne peut évidemment pas prendre en compte les caractéristiques physiques d'un réseau dont il ignore tout. Il tente plutôt, en fonction de l'équipement prévisible, d'affecter une identification de réseau cohérente avec le but du réseau.
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 par le *NIC* 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.

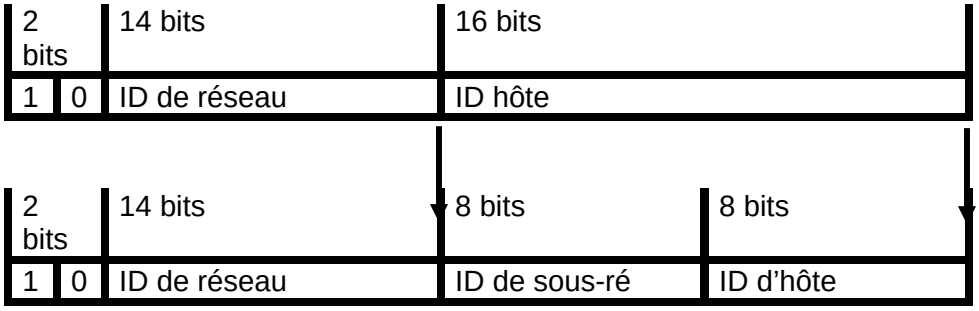


Figure 5 : Subdivision d'un espace d'adressage de classe B en sous-réseaux de classe C

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.

Description d'un paquet IP

Le protocole *IP* exploite une structure divisée en deux, définie dans la *RFC* 791. La première partie se compose de l'en-tête et la seconde des données. L'en-tête reçoit toutes les informations dont *IP* et les mécanismes qui lui sont reliés, routeur ou ordinateur, ont besoin pour transporter les données au sein du réseau.

Information
Que sont les RFC*?*
RFC est l'abréviation de *Request For Comment*. Il s'agit d'écrits donnant les détails techniques et le fonctionnement interne de la famille de protocoles *TCP/IP*. Les standards *IP* ne sont pas développés par un comité central mais élaborés en accord avec la communauté internet. En principe, tout membre de la société internet peut participer au développement en rédigeant une *RFC*. Celle-ci sera traitée et classifiée par des experts techniques ou un éditeur de *RFC*. Les niveaux de classification sont :

- nécessaire,
- recommandée,
- facultative,
- usage limité,
- non recommandée.

La classification lui attribue un numéro d'ordre. Lorsqu'une *RFC* est mise à jour, l'ancienne version n'est ni supprimée ni modifiée. Une nouvelle *RFC* est créée et diffusée sous un nouveau numéro. De ce fait, lorsque l'on consulte une *RFC*, on doit toujours vérifier s'il n'en existe pas une version plus récente.

Les paquets *IP* s'appellent des datagrammes. Un datagramme peut se comparer à l'ancien télégramme, d'où la similitude de nom. Il s'agit d'une certaine quantité de données, avec adresses et expéditeur, transmise à un destinataire. Il n'existe pas de mécanisme de sécurité tel un accusé de réception ou un retour à l'envoyeur confirmant la réception à l'expéditeur.

L'en-tête d'un paquet *IP* comporte plusieurs zones de données regroupées en blocs de 32 bits, ce qui est très pratique pour la présentation et la description des différentes zones. Ces données s'alignent verticalement et donnent au datagramme *IP* la structure représentée dans la figure 7.

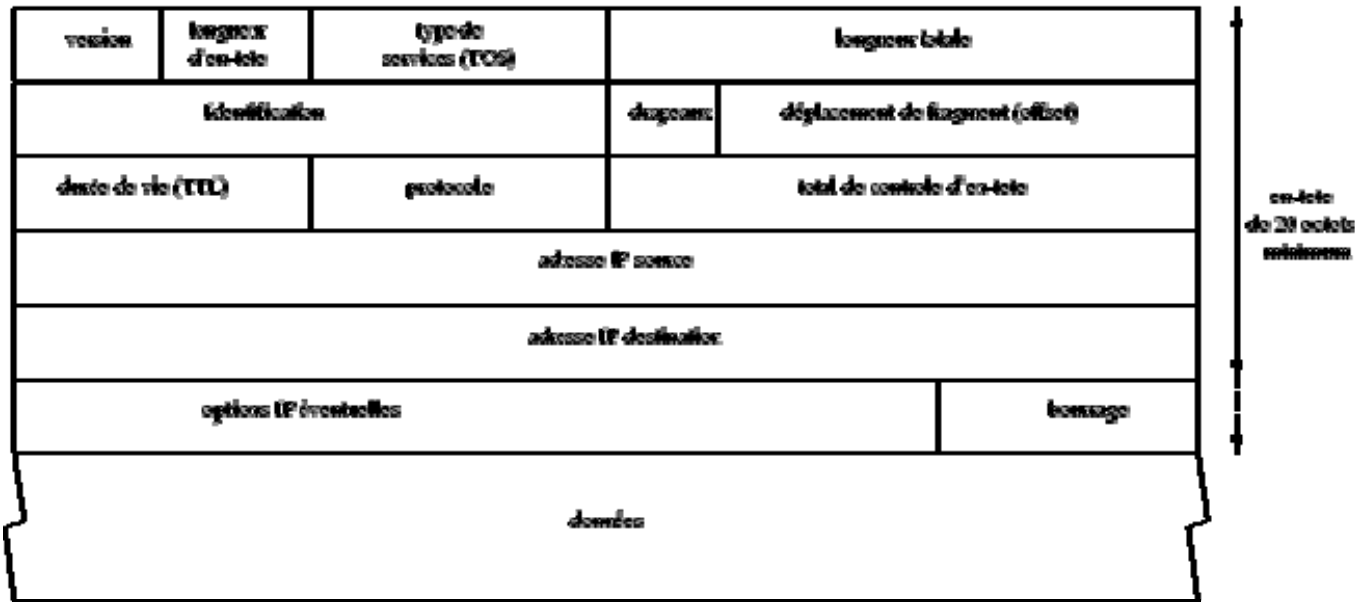


Figure 6 : Datagramme *IP*

Les champs de l'en-tête *IP* :
Version (4 bits) : ce champ désigne la version avec laquelle le paquet *IP* a été créé. Sont actuellement utilisées les versions 4 et 6, principalement la version 4. Il s'agit des adresses *IP* que nous connaissons, formées de 4 octets, 32 bits. Cette version est encore la plus courante. La version 6 devrait la remplacer sous peu.
Les datagrammes des deux versions sont gérés par le Net, mais ils sont incompatibles et ne peuvent s'employer dans un même flot de données.

IHL – Internet Header Length (4 bits) : *IHL* indique la longueur de l'en-tête *IP* en tant que multiple de 32, autrement dit de 4 octets. La section Options de 4 octets est également comptée. Une longueur de 5 indique que seul l'en-tête fixe de 5 x 32 bits = 160 bits = 20 octets est utilisé, c'est-à-dire qu'aucune option n'est définie. La valeur maximale binaire, 1111 = 15 décimal, correspond à une longueur d'en-tête de 15 x 32 bits = 480 bits = 60 octets.

Type de services (8 bits) : ce champ définit la qualité de service requise. Le champ de 8 bits s'analyse de la manière suivante :

Priorité (3 bits)	Délai (D) (1 bit)	Débit (T) (1 bit)	Fiabilité (R) (1 bit)	Coût (C) (1 bit)	0 (1 bit)

Priorité (3 bits) : ces bits attribuent au datagramme un niveau de priorité parmi 8. Une valeur forte correspond à une priorité également forte. Les anciens routeurs n'en tiennent pas compte. Les autres bits définissent d'autres priorités, activées lorsque le bit est à 1. Les routeurs récents les utilisent.

Les 8 niveaux de priorité d'un datagramme		
Binaire	**Décimal**	**Description**
000	0	Normal
001	1	Priority
010	2	Immediate
011	3	Flash
100	4	Flash override
101	5	Critical
110	6	Internet Control
111	7	Network Control

Tableau 4 : Les niveaux de priorité

Définition du type de transport		
Bit	**Significatio n**	**Description**
Bit D	Délai	Cherche une connexion comportant peu de latence.
Bit T	Débit	Cherche une haute vitesse de transfert.
Bit R	Fiabilité	Cherche un niveau de sécurité élevé. Les rejets (*discards*) sont rares.
Bit C	Coût	Cherche une route de faible coût.

Tableau 5 : Type de transport

Le dernier bit est inutilisé et a pour valeur 0.

Longueur du paquet – total length (16 bits) : il s'agit de la longueur totale du paquet. La différence entre la longueur de l'en-tête *IP* et la longueur du paquet fournit la quantité de données utiles. Une valeur de 16 bits correspond à la longueur maximale, soit 65 535 octets. La plupart des types de réseaux tels *Ethernet* ne gèrent pas cette taille de paquets et ces derniers seront fragmentés. Le champ ne contient donc pas la taille totale des données mais plutôt de celles contenues dans une trame.

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

Dans le cas d'un réseau interne, pour assigner des adresses réseau, il suffit de reprendre l'un des espaces d'adressage préconisés, le plus souvent un réseau de la classe C, et de déclarer le masque standard correspondant. Le problème est différent dans le cas d'une adresse *IP* officielle. Celle-ci ne correspondant pas à des ressources infinies et un réseau de classe A étant rarement nécessaire, même dans le cas d'un très grand organisme, et, enfin, une subdivision se justifiant probablement par la structure physique de réseau, l'espace d'adressage doit être partagé.

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 :

16 bits	10 bits	6 bits
ID de réseau	ID de sous-réseau	ID d'hôte

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 :

1111111111111111	1111111111	000000
ID de réseau	ID de sous-réseau	ID d'hôte

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.

16 bits	8 bits	2 bits	6 bits
ID de réseau	ID de sous-réseau		ID d'hôte

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.

8 bits	2 bits	6 bits
ID de sous-réseau		ID d'hôte

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.

Les protocoles Internet IPv6

Afin de satisfaire les exigences de l'Internet, un nouveau protocole a été conçu (mais non encore déployé) : le protocole *IP version 6* ou IPv6 . Cependant, *IPng* existe toujours, c'est-à-dire *IP nouvelle génération*. Actuellement, la version en cours d'utilisation est la version 4. Selon les experts de la question « *IP* », il faut s'attendre cependant à ce que l'Internet tel que nous le connaissons se heurte à de sérieuses difficultés, qui découlent de la croissance trop rapide du réseau des réseaux et des limites inhérentes à la conception de ses protocoles de communication, car il a été constaté que la taille de l'Internet double chaque année.

L'Internet, dans sa croissance exponentielle, entraîne dans son sillage celle des tables de routage des équipements qui sont sensés connaître toutes les routes mondiales (*full routing*). Or, en devenant gigantesque, ces tables posent de sérieux problèmes aux opérateurs de services *IP*. Fatalement, il arrivera un moment où il n'y aura plus d'adresses *IP* disponibles pour les utilisateurs futurs, vers 2010 selon certaines statistiques. Lorsqu'on arrivera à ce stade, plus aucun serveur web supplémentaire ne pourra être configuré, plus aucun utilisateur ne pourra créer de nouveaux comptes auprès d'un fournisseur d'accès Internet, et plus aucune machine ne pourra être configurée pour accéder au Web. Pour participer aux jeux en ligne, certaines personnes pourront trouver ce point comme étant un problème très grave!…

De nombreuses solutions ont été imaginées pour résoudre ce problème. Ainsi, une solution préconisait que plutôt que d'attribuer à chaque machine une adresse *IP* propre, il n'y avait qu'à cacher plusieurs postes de travail derrière une machine qui, elle, posséderait une adresse *IP* officielle reconnue. Cette solution est connue sous le nom de *NAT – Network Address Translation*, traduction d'adresse réseau, ou encore, *IP masquerading* – mascarade d'adresse *IP*. Mais cette solution avait malheureusement ses limites. En effet, la conséquence immédiate faisait qu'il était impossible d'adresser les machines cachées derrière l'adresse globale. Une autre conséquence était l'impossibilité de réaliser des connexions point-à-point. Or, ce type de connexion est essentiel pour les machines dédiées aux jeux en ligne. Cette solution fut abandonnée. Pour avoir plus de détails sur les *NAT*, nous vous suggérons de lire l'article *RFC* 3027.

Une autre approche du problème consistait à abandonner l'ancien protocole de l'Internet (l'actuelle suite *TCP/IP*) avec toutes ses limites d'adressage, et de le remplacer par un autre qui ne soit pas sujet à ces limites. La version 6 du protocole Internet pourra remplir cette fonction. IPv6 est en mesure d'adresser une plus grande plage d'adresses. Il est en outre doté d'une plus grande richesse fonctionnelle telle que la gestion et la sécurité des domaines privés, le cryptage des données ou encore un meilleur support pour les ordinateurs « nomades », le support du temps réel et du multipoint, etc.

IPv6 doit permettre d'adresser un espace beaucoup plus grand,10^9 réseaux au moins et fournir des techniques de routage plus efficaces en lien avec un adressage hiérarchique. Il faudra noter au passage que l'adressage actuel de IPv4 se fait sur 32 bits tandis que IPv6 est conçu selon une technologie d'adressage sur 128 bits. Grâce à cet adressage, il n'est plus besoin d'utiliser *NAT*, ce qui procure une très grande souplesse au niveau de la connectivité pour les machines actuelles architecturées autour de *IP*, de même que pour les futurs portables tels que les *PDA – Personal Digital Assistant*, assistant numérique personnel, mis en œuvre par Apple Computer en 1993 – qui sont des dispositifs tenant dans une seule main et qui englobent en particulier des caractéristiques d'ordinateur, de téléphone, de télécopieur et des caractéristiques réseau en général. Un *PDA* typique peut fonctionner comme un téléphone cellulaire ou un télécopieur. Mais contrairement aux micro-ordinateurs portables, à la place du clavier, ils utilisent des stylos spéciaux pour la saisie, puisqu'ils sont dotés de logiciels de reconnaissance de caractères. Certains *PDA* sont pourvus de la reconnaissance vocale. Il existe également des *PDA* à clavier.

Comparaison de IPv6 par rapport à IPv4

Le format des adresses de IPv6 s'étalera donc sur 128 bits plutôt que sur 32 bits pour IPv4. Une partie de cette adresse pourra être constituée de l'adresse *MAC* de l'équipement 48 bits. Enfin, l'adressage sera hiérarchique, c'est-à-dire qu'il sera organisé par zone géographique et (ou) par prestataire de services. Cette organisation de l'espace d'adressage permettra de réduire considérablement la taille des tables de routage actuelles.

Par ailleurs, une adresse IPv4 est normalement représentée par 4 chiffres décimaux compris entre 0 et 255 séparés par des points. Par exemple, l'adresse de l'hôte local (*localhost*) est représentée par : 127.0.0.1.

En théorie, cette représentation permet de connecter 2^{32} systèmes hôtes sur Internet. Cependant, la pratique a démontré que le groupage en sous-réseaux fait que toutes les adresses ne peuvent pas être disponibles.

IPv6, quant à lui, permet d'adresser théoriquement 2^{128} hôtes sur Internet. Ce qui, de toute évidence, est un très grand nombre de machines adressables, et largement suffisant pour couvrir tous les besoins actuels (PAD, téléphones cellulaires, même la téléphonie par Internet!) sans difficultés. Pour représenter un adressage IPv6, on utilise un groupe de huit chiffres hexadécimaux séparés par un deux-points (:). Pour reprendre l'exemple de l'adresse de boucle d'un hôte, en IPv6, sa représentation pourra être des plus simples. En effet :

« 127.0.0.1 » en IPv4 est équivalente à « ::1 » en IPv6.

En fait, la particularité de cette représentation est que les nombres non significatifs (nuls) sont tout simplement remplacés par « :: ». Mais attention, cette séquence ne peut apparaître qu'une seule fois dans une adresse IP!Ainsi, la représentation étendue de cette adresse est :

0:0:0:0:0:0:0:1

Vous avez pu noter au passage que le « 127 » de l'hôte local a disparu. Il n'a plus de raison d'être. C'est la nouvelle convention.

Voyons un autre exemple :

fe80::2a0:d2ff:fea5:e9f5
est aussi équivalent à
fe80:0000:0000:0000:02a0:d2ff:fea5:e9f5

Pour rendre les adresses *IP* gérables, on les divise en deux parties : les bits de la section réseau (*netbits*) définissent le réseau sur lequel est connectée une machine et les bits de la section hôte (*hostbits*) définissent l'appartenance d'une machine à un réseau ou à un sous-réseau. Pour IPv4 et IPv6, ces bits sont ainsi définis : les *netbits* sont les bits qui se trouvent à gauche (les bits les plus significatifs) tandis que les *hostbits* sont ceux qui se trouvent à droite (les bits les moins significatifs).

```
0<- - - - - - - - - - - - - - - - - - - - ->128
 +--------------------+--------------------+
 |     n netbits      |(128 – n) hostbits  |
 +--------------------+--------------------+
```

L'en-tête du paquet IPv6 est fortement simplifié (7 champs au lieu de 14 dans IPv4). Il inclut un champ d'extension pour les fonctionnalités optionnelles (sécurité, *source routing*, etc.).
Les options de IPv6 sont placées dans des en-têtes séparés, intercalés entre l'en-tête IPv6 et l'en-tête de la couche transport.

En-tête de IPv4 (sur 20 octets)

Version	Taille de l'en-tête	Type de service	Taille du datagramme	
Identification			Fanion	Déplacement du fragment
Durée de vie		Protocole	Somme de contrôle de l'en-tête	
Adresse *IP* source				
Adresse *IP* de destination				
Options et bourrage				

Figure 7 : Format du datagramme IPv4

Version (4 bits) (ici, IPv4)

Taille de l'en-tête (4 bits)
Longueur de l'en-tête en mots de 32 bits. La valeur minimale est 5 (20 octets sans option) et le maximum 15 (donc les options sont limitées à 40 octets).

Type de service (*ToS*) (8 bits)
Ce champ permet en théorie de distinguer différentes qualités de service et suppose que tous les paquets ne sont pas traités de la même façon. Ce champ se décompose en 3 bits de priorité (donc 8 niveaux) et trois indicateurs permettant de privilégier le débit, le délai ou la fiabilité. Ces champs sont rarement utilisés dans l'Internet actuel, mais pourraient être utilisés pour les mécanismes de qualité de service actuellement en cours de définition (*Diffserv*).

Longueur totale ou taille du datagramme (16 bits)
C'est la longueur totale du nombre d'octets complet du datagramme, en-tête *IP* compris. Il y a donc une limite de 2^{16} - 1 octets.

Identification (16 bits)
Ce numéro est affecté par l'émetteur et sert en particulier à identifier les fragments d'un même paquet.

Indicateurs ou drapeaux ou fanions (3 bits)
Le premier bit est actuellement inutilisé.
Deux indicateurs sont utilisés dans le cadre de la fragmentation :
Don't frag : le paquet ne doit pas être fragmenté si l'indicateur vaut 1.
More frag : le paquet est fragmenté et ce n'est pas le dernier fragment si l'indicateur vaut 1.

Déplacement du fragment (*Fragment Offset* ou *Offset*) (13 bits)
Ce paquet est un fragment, et sa position par rapport au début du paquet initial est donnée en nombre de mots de 8 octets (tous les fragments sauf le dernier sont donc alignés sur des multiples de 8 octets).

Durée de vie (*TTL – Time To Live*) (8 bits)
Ce champ est initialisé par l'émetteur puis diminué dans chaque routeur traversé (généralement de 1). Quand la durée de vie arrive à 0, le paquet est abandonné (avec émission d'un message *ICMP*).
Un fragment arrivé à destination voit aussi sa durée de vie diminuer tant que le paquet n'est pas complet, et peut donc être abandonné (message *ICMP*).

Protocole (8 bits)
Donne le numéro du protocole de la couche au-dessus de *IP*.

Contrôle d'en-tête (*Header Checksum*) (16 bits)
Contrôle sur l'ensemble des octets de l'en-tête. Un paquet erroné est abandonné silencieusement. Il n'y a pas de contrôle sur les données du paquet.

Adresse source (32 bits)
Adresse *IP* de l'émetteur : adresse à destination unique (*unicast*), généralement celle de l'interface par laquelle le paquet est envoyé.

Adresse destination (32 bits)
Adresse *IP* du récepteur : peut être une adresse à destination unique, multidiffusion (*multicast*) ou à diffusion générale (*broadcast*).

Options
Ce champ facultatif peut contenir une ou plusieurs options, dont la longueur totale est un multiple de 4 octets avec un maximum de 40 octets. Il existe des options courtes comportant uniquement un champ type d'un octet, et des options longues comportant un champ type (1 octet), un champ longueur (1 octet) et des paramètres.

En-tête de IPv6 (sur 40 octets)

Version	Priorité	Étiquette d'identification de flux	
Charge utile		En-tête suivant	Nombre de sauts
Adresse *IP* Source			
Adresse *IP* de destination			

Figure 8 : Format du datagramme IPv6

Version : (4 bits) pour IPv6.

Priorité (Priority) : (4 bits) pour la priorité. Le champ version est à la même place que le champ version de la IPv4.

Étiquette d'identification de flux (Flow Label) : (24 bits) pour la qualité de service.

Charge utile (Payload Length) : (16 bits) (entier non signé) le nombre d'octets qui suit. C'est la longueur de la « charge utile », soit le reste du paquet qui suit l'en-tête de IPv6. Il faut noter que tous les en-têtes d'extension présents sont considérés comme faisant parti de la charge utile, c'est-à-dire qu'ils sont inclus dans le décompte de la longueur.

En-tête suivant (Next Header) : (8 bits)
Sélecteur sur 8 bits. Identifie le type de l'en-tête suivant immédiatement l'en-tête IPv6. Utilise les mêmes valeurs que le champ « protocole » d'IPv4.

Nombre de sauts (Hop Limit) : (8 bits) (entier non signé)
Décrémenté de 1 par chaque nœud que le paquet traverse. Le paquet est éliminé si le nombre de sauts maximum arrive à zéro.

Adresse IP source (Source Address) : (128 bits), l'adresse de l'envoyeur.

Adresse IP de destination (Destination Address) : (128 bits), l'interface de destination. Sauf si l'en-tête de route optionnelle est présent.

Nouvelles fonctionnalités du protocole IPv6

La sécurité, tant décriée ou tant vantée – selon le point de vue où l'on se place –, sera rendue par des fonctions d'authentification et d'intégrité des données (*SAID – Security Association Identifier*, MD5, etc.), utilisées entre les stations source et destination. La fonction de confidentialité est assurée par le chiffrement partiel (données seules) ou complet du datagramme. Pour plus de détails sur les mécanismes de sécurité retenus pour IPv6, nous vous recommandons de consulter les appels de commentaires (*RFC – Request for comments*) 1826, 1827, 1828 à 1829.

En plus de la mobilité des équipements sur les réseaux, la sécurité était un autre préalable au successeur de IPv4. Ainsi, IPv6 intègre *IPsec* qui permet l'authentification, le cryptage et la compression du trafic *IP*. À l'exception des protocoles tels que *SSL – Secure Sockets Layer* ou *Ssh – Secure Shell* de la couche application, tout le trafic *IP* internœud est pris en compte sans aucun paramétrage des applications. L'avantage est que toutes les applications d'une machine peuvent bénéficier du cryptage et de l'authentification et, de plus, ces gestions peuvent être paramétrées par machine ou par réseau et non pas par service. On peut trouver une documentation sur *IPsec* en consultant le RFC 2411, la description propre du protocole pouvant être trouvée à RFC 2401.

Le routage en fonction de l'adresse de la source (*Source Routing*) (IPv4 ne route qu'en fonction de l'adresse de destination) sera implanté grâce au *SDRP – Source Demand Routing Protocol*. Il permettra le routage différencié (ou « politique »).

La configuration automatique des équipements sera rendue possible grâce à un protocole antérieur à la spécification de IPv6 mais qui sera adapté en conséquence : le *DHCP – Dynamic Host Configuration Protocol* (*RFC* 1541). Il s'agit d'une fonctionnalité qui vise à simplifier la phase de connexion d'un équipement au réseau. Cette caractéristique est communément appelée le « plug and play ». Elle permettra également de gérer la mobilité des équipements en rendant aisée la numérotation ou renumérotation en cas de besoin.

La multidiffusion devient une propriété intrinsèque de IPv6 pour les routeurs comme pour les postes de travail. Ce qui implique que dans le monde de IPv6, on pourra se passer de *Mrouted* sur les stations, et que le réseau *Mbone* n'aura plus de raison d'être : le trafic multipoint deviendra alors complètement banalisé.

Dans l'univers de *IP*, il existe trois manières pour communiquer avec un hôte : la diffusion à destination unique (*unicast*), à diffusion générale (*broadcast*) et la multidiffusion (*multicast*), la plus simple étant la diffusion individuelle.

Pour IPv4, une adresse de diffusion individuelle est l'adresse *IP* normale attribuée à un hôte. Il en va de même pour les bits d'adresse. En ce qui concerne l'adressage de diffusion utilisée pour l'ensemble des hôtes appartenant à un même sous-réseau, les *netbits* ne sont identiques que pour le réseau tandis que les *hostbits* sont à « 1 ». Les adresses de multidiffusion sont quant à elles employées pour joindre un certain nombre de machines appartenant à un même groupe de multidiffusion, étant entendu que ces machines peuvent se trouver n'importe où sur Internet. Il faut reconnaître que IPv4 n'utilise pas tellement la multidiffusion. Il ne s'en sert que pour très peu d'applications telles que la diffusion vidéo.

En ce qui concerne IPv6, les adresses de diffusion individuelle sont identiques à celles de IPv4. Par contre, l'adressage pour les diffusions de IPv4 n'est plus valable pour IPv6 et c'est à ce niveau que la multidiffusion entre en jeu. Toutes les adresses réseau comprises dans ff::/8 sont réservées aux applications de multidiffusion. De plus, deux adresses spéciales de multidiffusion annulent la diffusion en provenance de IPv4. L'une d'elles est l'adresse de multidiffusion *all routers*, et l'autre *all hosts*. Ces adresses sont spécifiques aux sous-réseaux. Ainsi, un routeur connecté à deux sous-réseaux peut adresser tous hôtes ou tous les routeurs de n'importe quel sous-réseau auquel il est relié. Dans ce cas, ces adresses seront du genre :

ff0X::1 pour tous les hôtes et
ff0X::2 pour tous les routeurs,

Et, toujours dans l'exemple ci-dessus, « X » identifie le réseau.

Enfin, n'étant pas en mesure de pouvoir énumérer toutes les fonctionnalités de IPv6 sans renvoyer aux documents de référence (*RFC* 1752 et I-D[1][note 1]), on évoquera pour terminer les fonctionnalités de gestion des applications en temps réel. Elle sera rendue possible par l'utilisation du champ d'en-tête *Flow Label*, qui permet de différencier certains flux de données par rapport aux autres. Cela nécessitera la mise en œuvre d'un mécanisme de contrôle, notamment sur les équipements de routage tels que *RSVP – Ressource reSerVation Protocol*, par exemple I-D[3].

La transition de IPv4 vers IPv6 – dont l'une des données majeures est la vitesse d'épuisement des adresses IPv4 – peut se découper en trois phases :

- Phase où seuls des équipements IPv4 existent.

On arrive aujourd'hui à la fin de cette phase, puisque de nombreux constructeurs sont sur le point de proposer les premières versions de IPv6 pour les postes de travail et les routeurs (sans parler des plates-formes de tests déjà en place).

- Phase de coexistence d'équipements IPv4 et IPv6.

Cette phase sera probablement très longue et caractérisera l'Internet du siècle prochain.

- Enfin, phase où seuls subsisteront des équipements IPv6.

Trois techniques, décrites sommairement dans les paragraphes qui précèdent, ont été spécifiées à ce jour, ce qui ne préjuge pas de l'émergence d'autres possibilités :

- la « double pile *IP* », où chaque équipement implante complètement les deux protocoles *IP* (v4 et v6);
- l'encapsulation (*tunneling*) des paquets IPv6 dans des en-têtes IPv4 pour les acheminer à travers une infrastructure IPv4;
- la traduction des en-têtes IPv6 en en-têtes IPv4 (voire l'inverse...).

Néanmoins de nombreuses implantations sont en tests chez les constructeurs tant pour les postes de travail que pour les équipements de routage.

Il s'agit maintenant de déterminer ce qui doit migrer vers IPv6 et à quel moment. La réponse pourra être liée aux priorités de chacun, mais aussi aux possibilités nouvelles offertes par IPv6 et qui, pour certaines d'entre elles au moins, vont probablement devenir essentielles aux applications courantes de l'avenir : applications de vidéoconférence ou en temps réel, nécessitant une qualité de service garantie, la configuration automatique des équipements, la sécurité des données transportées.

Pour plus d'informations sur les développements de IPv6, on peut toujours consulter les pages suivantes :

http://www.fr.ipv6tf.org/
http://www.g6.asso.fr/tff/index.php/Main_Page
http://playground.sun.com/pub/ipng/html/ipng-main.html

Voici quelques sources d'informations sur le développement de IPv6 :
RFC 1550- IP: Next Generation (IPng) White Paper Solicitation. Dec 1993
RFC 1726- Technical Criteria for Choosing IP the Next Generation. Dec 1994
RFC 1752- The Recommendation for IP the Next Generation. Jan 1995
RFC 1826- IP Authentication Header. Aug 1995
RFC 1827- IP Encapsulating Security Payload. Aug 1995
RFC 1828- IP Authentication using Keyed MD5. Aug 1995
RFC 1829- The ESP DES-CBC Transform. Aug 1995
RFC 1541- Dynamic Host Configuration Protocol. Oct 1993
I-D[1]- Internet Draft: draft-ietf-ipngwg-ipv6-spec-02.txt. Jun 1995
I-D[2]- Internet Draft: draft-ietf-ngtrans-trans-mech-01.txt. May 1995
I-D[3]- Internet Draft: draft-ietf-rsvp-spec-07.txt. Jul 1995

Le protocole *IPX/SPX*

Ces deux sigles représentent *Internetwork Packet Exchange Protocol* et *Sequenced Packet Exchange Protocol*. Contrairement à ce qu'on pourrait penser, ce protocole n'a aucun rapport avec le protocole *IP* d'Internet.

IPX/SPX est un protocole spécifique à une entreprise. Il sera principalement exploité au sein de réseaux *Novell Netware*. Il s'agit d'un système de serveurs qui, à l'instar de Windows NT/2000 Server, propose des serveurs d'impression et de fichiers ainsi que des services d'annuaire. Le système repose sur un *IPX/SPX* spécialement adapté à cette tâche. *IPX* est le protocole de transmission en mode déconnecté, qui s'intègre dans la couche OSI 3 et qui assure les fonctions d'adressage et de routage des paquets. *SPX* a également été développé par Novell et représente une extension d'*IPX*. Il autorise la transmission orientée connexion des paquets entre les stations et étend la fiabilité du protocole *IPX* par sa seule fonction de transport.

Les autres types de protocole (*NetBEUI*, *AppleTalk*)

NetBIOS et *NetBEUI*

Nous n'élaborerons pas sur des protocoles désuets tels que *DECNET* ou *Ethertalk* qui ne se retrouvent plus dans les réseaux modernes.
Le *NetBIOS* constitue une interface de programmation d'accès aux réseaux locaux sous les systèmes d'exploitation MS-DOS, OS/2 et certains UNIX exploités sur des ordinateurs compatibles IBM. L'interface est exploitée par le *NetBEUI*.
NetBIOS et *NetBEUI*, tous deux développés par IBM, constituent une famille de protocoles quoique, en toute rigueur, seul *NetBEUI* corresponde réellement à un protocole. Il s'agit en effet d'un petit protocole rapide et efficace qui s'intègre au niveau de la couche OSI 4 (la couche transport) et qui autorise une application à ouvrir une session d'un programme sur un ordinateur distant.
L'avantage de *NetBEUI* est sa petite pile de protocoles, ce qui explique qu'il trouve son emploi principalement sur les ordinateurs fonctionnant sous MS-DOS. Ceux-ci exigent un protocole léger et rapide, peu gourmand en ressources.
Le grand désavantage de *NetBEUI* est qu'il n'est pas routable et que son application se limite, par conséquent, à un segment de réseau. Certain routeurs transmettent bien à la totalité des segments qu'ils gèrent les paquets *NetBEUI* qu'ils reçoivent, et de ce fait, s'effacent devant *NetBEUI*, mais le résultat final est une augmentation de la charge du réseau alors que le routage est censé la réduire! *NetBEUI* est donc inadapté aux grands réseaux.

AppleTalk

Le protocole *AppleTalk* a été conçu par la compagnie Apple pour relier les ordinateurs de type Macintosh. Il utilise la méthode d'accès au support *CSMA/CA*. Trois événements se produisent lorsqu'un dispositif s'attache :

- le dispositif s'assigne lui-même une adresse aléatoire,
- il fait ensuite une diffusion de l'adresse pour voir s'il y a un autre dispositif avec la même adresse,
- sinon, le dispositif l'utilisera la prochaine fois qu'il se connectera.

Le protocole *AppleTalk* s'installe sur une topologie en arbre ou en bus en fonction du type de câble utilisé et peut prendre en charge jusqu'à 32 périphériques.

Conclusion

En guise de conclusion, nous devons retenir les points suivants :

TCP/IP est une suite de protocoles qu'on peut qualifier de « Super Star »! Cela est dû au fait qu'il possède des atouts non négligeables. C'est avant tout une suite de protocoles ouverts. Les sources en langage C sont disponibles gratuitement sur Internet et ces protocoles sont développés indépendamment des architectures réseaux existantes. Elles ne sont donc tributaires d'aucune architecture ni d'aucune structure commerciale. Et tous ceux qui se sentent perfectionnistes dans l'âme sont invités à leur apporter des améliorations! La majeure partie des informations relatives à ces protocoles sont publiées dans les RFC (*Requests for comments*). Il est donc conseillé de s'y référer afin de prendre connaissance des dernières améliorations apportées aux protocoles *TCP/IP*, ainsi que de la description de nouveaux protocoles, des commentaires sur la gestion des réseaux… et la liste n'est pas exhaustive.

En ce qui concerne l'adressage *IP*, quelques précisions supplémentaires s'imposent. Nous avons, en effet, désigné un hôte par son adresse *IP*. Or, cette démarche n'est pas tout à fait exacte. Si nous devons considérer une passerelle, elle est physiquement connectée à au moins deux réseaux différents! La passerelle possède ainsi une adresse *IP* distincte dans chacun des réseaux. Le mieux serait de dire qu'une adresse *IP* est une interface plutôt qu'un hôte, car l'hôte en question peut se retrouver avec plusieurs adresses *IP*! On dit d'une machine ayant au moins deux adresses *IP* qu'elle est du type « *multi-homed* ».

IPv6 est le successeur annoncé du protocole IPv4; il est adressable sur 128 bits plutôt que sur les 32 bits de son prédécesseur IPv4. Certains défenseurs d'IPv6 ont même été jusqu'à conclure (après calculs) que l'adressage d'IPv6 permettrait de disposer de « plusieurs milliers d'adresses pour chaque mètre carré de la surface terrestre »! Ainsi, grâce à ce stock énorme d'adresses, le problème de la pénurie d'adresses *IP* annoncée semble momentanément écarté. On pourrait penser que les longues adresses d'IPv6 sont lourdes à manipuler puisqu'elles demandent plus d'espace mémoire. Pourtant, de par la conception de IPv6, beaucoup plus ouverte que celle de IPv4, la compilation des tables de routage est plus rapide, et la compression des données plus efficace.