RÉSEAUX INFORMATIQUES, MODÈLE OSI ET PROTOCOLE TCP/IP

TODO :

- 2.6.4.3 à 2.6.4.8 : schémas
- 4.3.2.3 : nouvel exemple n'ayant pas besoin d'utiliser ni le premier ni le dernier sous-réseau

v1.1.5.5 – 20/05/2010
peignotc(at)arqendra(dot)net / peignotc(at)gmail(dot)com

TABLE DES MATIÈRES

TABLE DES ANNEXES

TABLE DES ILLUSTRATIONS

1 INTRODUCTION AUX RÉSEAUX

1.1 PRINCIPES GÉNÉRAUX DES RÉSEAUX

Chaque jour, consciemment ou pas, chacun d'entre nous utilise divers réseaux « généraux » : les routes, le téléphone, le courrier, l'eau, l'électricité, etc. Le but de chacun de ces réseaux est de fournir les mêmes services à chaque élément du réseau, appelé **point** ou **nœud** du réseau ; pour cela les nœuds du réseau sont reliés par le biais d'une solution unique.

Ex. : Le réseau autoroutier français : chaque nœud du réseau est une grande ville, les nœuds sont reliés entre eux par des autoroutes ; ce réseau offre comme service, en partance d'un nœud du réseau, de permettre de rejoindre les autres nœuds le plus rapidement possible par voie terrestre.

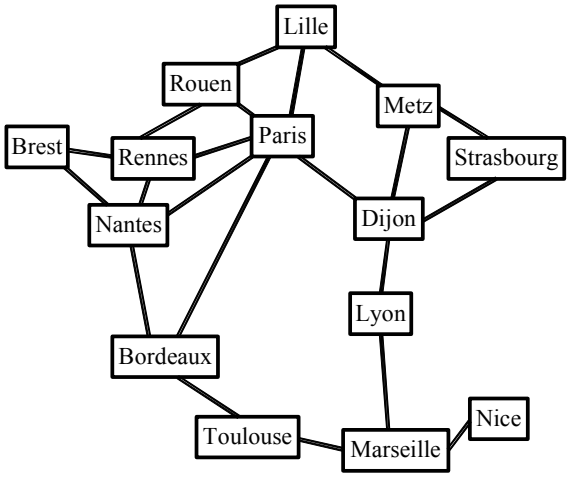

Figure 1.1 : exemple du réseau autoroutier français

1.2 NOTIONS GÉNÉRALES ET VOCABULAIRE

Lorsque l'on utilise le réseau autoroutier, on circule d'une ville de départ vers une ville d'arrivée en progressant au fur et à mesure via un certain nombre de villes intermédiaires ; on ne passe donc pas par tous les nœuds du réseau. Le réseau autoroutier est un réseau **point-à-point**.

Le réseau électrique fonctionne différemment : la centrale électrique fournit de l'électricité pour chaque logement qui y est raccordé ; le service est donc diffusé par le nœud-source à tous les nœuds du réseau, qu'on l'utilise ou pas [1]. Il s'agit là d'un réseau **multi-points**, ou réseau **à diffusion**.

L'usage du réseau autoroutier impose un certain nombre de règles, inscrites dans le code de la route. L'envoi d'un courrier via le réseau postal oblige à spécifier les coordonnées du destinataire, selon des règles précises. L'utilisation d'un réseau implique donc le respect de diverses règles et/ou mode opératoire pour son bon fonctionnement local ou global. Ces règles sont regroupées sous le terme de **protocole**.

Lors d'une communication par téléphone, en fonction du volume des informations à transmettre, on peut prévoir la durée approximative de la conversation téléphonique ; de plus, à la fin de la communication, étant donné la vitesse de

[1] D'où la nécessité d'un système de mesure de l'utilisation du réseau par chaque nœud (dans cet exemple, un compteur électrique).

transmission des données voix, on est sûr que les données ont bien été transmises [1]. La durée de la transmission des informations est donc prévisible. Dans ce cas-là, on parle de réseau **déterministe**.

À l'inverse, l'envoi d'un courrier postal est bien souvent aléatoire, et on ne saurait dire, au moment où le courrier est pris en charge par le réseau postal, dans combien de jours le destinataire recevra les informations [2]. La transmission des informations via le réseau n'apporte aucune garantie de sa transmission complète en un laps de temps prévisible. Il s'agit là d'un réseau **probabiliste**.

Lorsque l'on désire communiquer des informations à quelqu'un par téléphone, on ne parle que si le correspondant décroche ; lors de la communication, on s'assure généralement que le destinataire a bien reçu les informations. L'émetteur a donc en retour des garanties que l'information a été correctement transmise. De la même manière, l'envoi d'un courrier avec accusé de réception permet à l'envoyeur de lui signifier la livraison effective de ses informations au destinataire. On parle de communication **avec connexion**, ou en **mode connecté**.

Inversement, un courrier simple n'offre aucun service de ce type à l'envoyeur. Il s'agit d'une communication **sans connexion**, en **mode non connecté**.

Le réseau postal utilise les réseaux autoroutier, ferroviaire et aérien pour la réalisation des services attendus ; tous les courriers pris en charge, sont triés avant d'être acheminés ; enfin, il faut encore les distribuer. Les services offerts sont donc architecturés selon plusieurs niveaux (tri, acheminement, distribution, etc.), et un réseau peut utiliser les services offerts par un autre réseau. On parle de découpage du réseau **en couches**.

Au niveau du tri, les courriers vers une même destination (région, ville, etc.) sont regroupés dans des sacs étiquetés ou des containers, et sont alors pris en charge par les moyens de locomotion du réseau d'acheminement ; le conditionnement des informations devant transiter est donc modifié avant que celles-ci soient prises en charge par une autre couche du réseau. Il s'agit là du principe de **formatage** et d'**encapsulation des données** pour passage sur une autre couche.

Toutes les différentes notions mises en exergue dans les réseaux « généraux » sont également mises en œuvre dans les réseaux informatiques.

[1] La transmission est une chose, la compréhension et le traitement adéquat en sont d'autres…

[2] Le réseau lui-même se sait non-fiable : « engagement sur 48h ou remboursé »… (nda : no comment).

2 NOTIONS DE BASE

2.1 DÉFINITION

Un **réseau informatique** est un réseau dont chaque nœud est un système informatique autonome, reliés par un support matériel et logiciel, et qui ont ainsi la possibilité de communiquer entre eux directement ou indirectement.
En pratique, 2 ordinateurs suffisent pour constituer un réseau informatique [1].

Les services offerts par un réseau informatique peuvent être de nature très diverses et multiples, mais généralement prennent l'une des formes suivantes :

- échanger des informations ;
- partager et centraliser des ressources matérielles et/ou logicielles.

Nb : En informatique, le terme *réseau* véhicule différentes notions en fonction du contexte : il peut désigner l'ensemble des nœuds, l'ensemble des ordinateurs, la topologie, le protocole, la catégorie, etc.

Là encore, chacun d'entre nous utilise très souvent, consciemment ou pas, les réseaux informatiques : le réseau internet (et son usage au quotidien, le WWW [2]), le réseau Transpac (réseau français interbancaire), etc.
Les entreprises, et depuis bien longtemps avant les particuliers, font aussi un grand usage des réseaux : réseau de capteurs/actionneurs, réseau d'automates, réseau administratif, etc.

2.2 CATÉGORIES DE RÉSEAUX

Les différents types de réseaux informatiques peuvent être classifiés selon leur étendue (éloignement maximal entre systèmes informatiques connectés) :

- bus de terrain / réseau local industriel (RLI) : réseau orienté matériel, pour des systèmes informatiques reliés à des capteurs/actionneurs (IEEE488, CAN, I2C, AS-i, Profibus, Modbus, VME, etc.) ;
- LAN (Local Area Network) [3] : réseau local, pour des systèmes informatiques appartenant à une même entreprise (Ethernet, Token Ring, Appletalk, Telway, etc.) ;
- MAN (Metropolitan Area Network) : réseau de ville (BLR, WiMax, etc.) ;
- WAN (Wide Area Network) [4] : large réseau national ou international (Transpac, Telenet, etc.) ;
- internet : interconnexion de divers WAN (auxquels peut être raccordé tout MAN ou LAN).

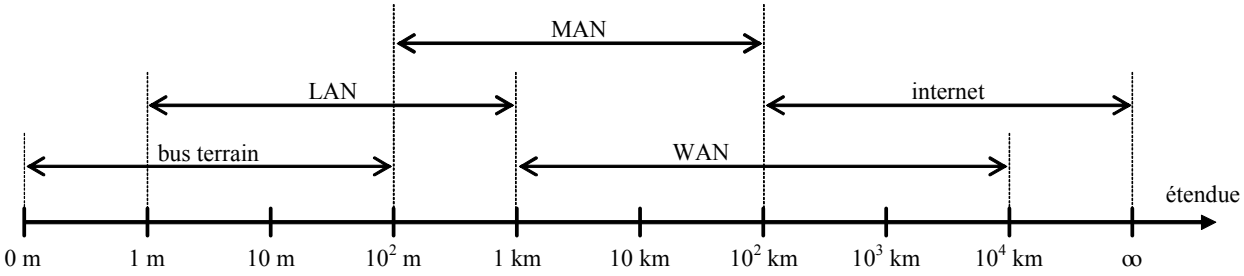

Figure 2.1 : catégories de réseaux

[1] Et les difficultés qui vont avec… qui n'a jamais eu de problème en voulant constituer une LAN-party occasionnelle entre amis ??!
[2] WWW : World-Wide Web (eng) ≡ toile mondiale (fr).
[3] Ou RLE (Réseau Local d'Entreprise).
[4] Ou RLD (Réseau Longue Distance).

2.3 SUPPORTS DE TRANSMISSION

Quelque soit la catégorie de réseaux, seules 3 technologies différentes sont mises en œuvre :

technologie	type	débit ou bande passante	remarques
fils métalliques	paire torsadée	1 Gbits/s pour 100m < 1 Mbits/s pour 1 km	sensible au bruit [1]
	câble coaxial	10 Mbits/s pour 1 km	très employé
fibre optique	multimode à saut d'indice	40 MHz pour 1 km	faible atténuation 1 répéteur / 10 km
	multimode à gradient d'indice	500 MHz pour 1 km	
	monomode	1 GHz pour 1 km	
ondes radioélectriques ou électromagnétiques	infrarouge	1 Mbits/s pour 1 m	petites distances (10 m)
	satellite géostationnaire	3-10 GHz	latence de 260 ms
	faisceau terrestre	2-40 GHz	

2.4 TOPOLOGIES DE RÉSEAUX

Le terme **topologie** désigne l'architecture physique des interconnexions entre les différents nœuds d'un réseau.

2.4.1 Réseaux point-à-point

Un **réseau point-à-point** est un réseau dans lequel les interconnexions entre les nœuds sont réalisées 2 par 2.

Après une réflexion triviale, il paraît facile de construire un réseau point-à-point fonctionnel en établissant une liaison directe entre chaque couple de nœuds. Cependant pour des raisons évidentes de coût et de logistique, cela est difficilement envisageable [2]. Il faut donc concevoir un réseau point-à-point viable avec des liaisons indirectes, et des nœuds intermédiaires, le tout de manière adaptée aux besoins du réseau.

Quatre structures différentes de réseaux point-à-point existent : en étoile, en arbre, en boucle, ou maillé.

2.4.1.1 Réseau point-à-point en étoile

Dans un réseau point-à-point en étoile, tous les nœuds sont reliés à un nœud central (serveur, machine puissante ou un appareillage dédié) ; le nombre de liaisons est ainsi minimal (N–1 liaisons pour N nœuds). Mais le réseau est limité par les possibilités du nœud central (bloqué en cas de défaillance, taille limitée aux possibilités de connectivité).

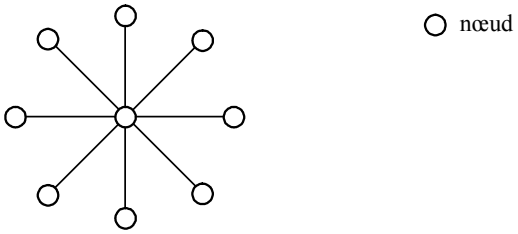

Figure 2.2 : topologie d'un réseau point-à-point en étoile

2.4.1.2 Réseau point-à-point en arbre

Un réseau point-à-point en arbre est un assemblage de réseaux point-à-point en étoile.

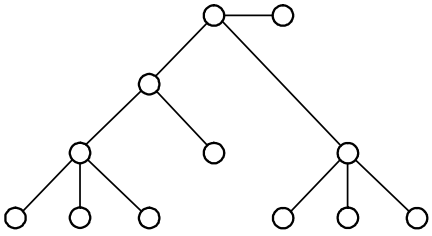

Figure 2.3 : topologie d'un réseau point-à-point en arbre

[1] L'appellation « bruit » fait référence aux perturbations électromagnétiques.

[2] Avec N nœuds, il faut établir $C_2^N = N \times (N-1)/2$ liaisons (nombre de combinaisons de 2 parmi N) ; ex. : 45 liaisons pour 10 nœuds.

2.4.1.3 Réseau point-à-point en boucle

Dans un réseau point-à-point en boucle (dit aussi en anneau), chaque nœud n'est relié qu'à 2 autres nœuds ; le nombre de liaisons est faible (N liaisons pour N nœuds). Généralement un nœud défaillant est court-circuité et ainsi ne bloque pas le réseau.

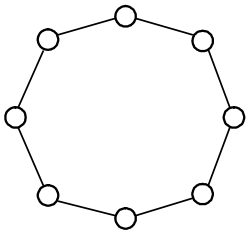

Figure 2.4 : topologie d'un réseau point-à-point en boucle

2.4.1.4 Réseau point-à-point maillé

Dans un réseau point-à-point maillé, chaque nœud est relié à un ou plusieurs autres nœuds sans logique particulière [1], proposant généralement ainsi plusieurs chemins différents entre deux nœuds quelconques, ce qui renforce la fiabilité du réseau.

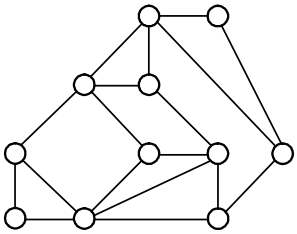

Figure 2.5 : topologie d'un réseau point-à-point maillé

2.4.2 Réseaux multi-points

Un **réseau multi-points** est un réseau dans lequel tous les nœuds sont reliés entre eux via une liaison unique ; chaque nœud y étant connecté, il s'agit donc d'un réseau à diffusion.

Les réseaux multi-points représentent une solution très économique et évolutive, la liaison unique étant le plus souvent constituée d'une paire de fils torsadée.

Trois structures différentes de réseaux multi-points existent : en bus, en anneau, ou en boucle.

2.4.2.1 Réseau multi-points en bus

Dans un réseau multi-points en bus, tous les nœuds sont reliés directement à la liaison centrale, laquelle est tirée entre les deux nœuds les plus éloignés.

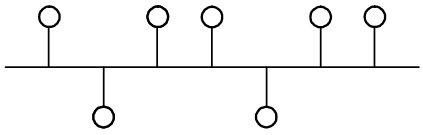

Figure 2.6 : topologie d'un réseau multi-points en bus

2.4.2.2 Réseau multi-points en anneau

Dans un réseau multi-points en anneau, tous les nœuds sont reliés directement à la liaison centrale, laquelle reboucle sur elle-même, les communications sont monodirectionnelles.

[1] C'est généralement le coût ou le besoin en ressources qui dirige l'établissement des liaisons.

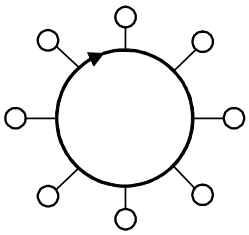

Figure 2.7 : topologie d'un réseau multi-points en anneau

2.4.2.3 Réseau multi-points en boucle

Un réseau multi-points en boucle est similaire à un réseau multi-points en anneau, à la différence que les communications sont bidirectionnelles.

2.4.3 Exemples

Internet : point-à-point maillé ;
Ethernet, Ethernet avec utilisation d'un concentrateur (hub) : multi-points en bus ;
Ethernet avec utilisation d'un commutateur (switch) : point-à-point en étoile ;
Bus de terrain : multi-points en bus ;
Token Ring : multi-points en anneau ;
FDDI : multi-points en boucle.

2.5 MODES DE TRANSMISSION

Considérons un réseau constitué uniquement de 2 nœuds, en l'occurrence 2 ordinateurs équipé chacun d'un élément de connexion physique au média.

Il s'agit donc d'un réseau point-à-point. Du point de vue strict du réseau, le nœud est l'élément de connexion, et pas l'ordinateur ni l'ensemble ordinateur + élément de connexion [1].

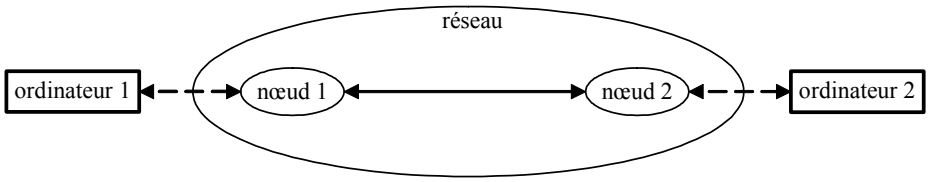

Figure 2.8 : exemple d'un réseau point-à-point à 2 nœuds

La constitution d'un réseau requiert 2 niveaux, qui pourraient être assimilés à des couches :

- un niveau est dédié à l'accès au média de communication et à la transmission des données via ce média ; ces opérations sont réalisées par l'Équipement Terminal de Circuit de Données (ETCD [2]) ;
- l'autre niveau s'occupe du traitement des données transmises du / vers le premier niveau ; ces opérations sont réalisées par l'Équipement Terminal de Traitement de Données (ETTD [3]).

L'ETCD est un élément matériel en général [4] (carte réseau, carte WiFi, modem, …), alors que l'ETTD est un élément logiciel (logiciels pilotes de l'interface).

Quelque soit le type de réseau, les données à transférer peuvent être formatées de 2 manières :

- en parallèle : les données sont transmises sur plusieurs bits à la fois en même temps (8, 16 ou 32 bits [5] en général), soit donc en utilisant un média constitué de plusieurs lignes différentes ;
- en série : les données sont transmises bit par bit [6], ne nécessitant basiquement qu'un média constitué de 2 fils.

Une transmission parallèle est potentiellement plus rapide, mais reste une solution inadaptée aux transferts sur une longue distance, car plus coûteuse, logistiquement moins facile à mettre en oeuvre, sujette à baisse de la qualité du

[1] Par ailleurs, un même ordinateur peut disposer de plusieurs éléments de connexion et être ainsi relié à plusieurs nœuds distincts.
[2] ETCD (fr) ≡ DTE : Data Terminal Equipment (eng).
[3] ETTD (fr) ≡ DCE : Data Communication Equipment (eng).
[4] Contre-exemple : l'interface loopback (127.0.0.1 ou localhost) est un exemple d'interface logicielle.
[5] Nombres puissance de 2.
[6] Avec lecture de chaque bit de données grâce à un registre à décalage.

signal de par sa conception [1], et plus sensible à la qualité de l'environnement électrique. Le plus souvent, on met donc en œuvre une transmission série, qui est plus lente, mais moins coûteuse et plus fiable.

Cependant, dans le cas de la transmission série, avec une seule ligne pour plusieurs informations et types d'informations, se pose le problème de la synchronisation entre l'émetteur et le récepteur [2]. La transmission peut alors être de 2 types :

- synchrone : l'émetteur et le récepteur ont une horloge de même fréquence, tous les accès en écriture sont donc synchronisés avec un accès en lecture (émission ou silence) ;
- asynchrone : l'émetteur et le récepteur ont des horloges de fréquences différentes, chaque début et chaque fin de transmission sont signifiés par l'envoi d'un bit particulier.

Au-delà des difficultés voire des impossibilités de synchroniser les fréquences de l'émetteur et du récepteur, une transmission synchrone est cependant plus lente afin de déterminer le début et la fin d'une transmission. En revanche, dans une transmission asynchrone, le début et la fin d'une transmission sont déjà marqués, et la vitesse globale de transmission est donc plus rapide ; c'est donc la solution généralement mise en œuvre.

L'accès au média partagé peut s'effectuer suivant 3 modes différents :

- unidirectionnel (simplex) : transmission possible dans un seul sens, l'un des nœuds est émetteur et l'autre est récepteur ;
- bidirectionnel alterné (half-duplex) : transmission possible dans les deux sens mais à tour de rôle, les nœuds sont alternativement émetteur puis récepteur ; il y a donc un risque de collision et pertes des données si les deux nœuds émettent en même temps, ce qui impose de mettre en place des mécanismes spécifiques (synchronisation, détection et récupération d'erreurs, etc.) ;
- bidirectionnel simultané (full-duplex) : transmission possible dans les deux sens en même temps, les nœuds sont à la fois émetteur et récepteur.

2.6 ACCÈS AU RÉSEAU

L'accès au réseau sur un réseau multi-points est un élément complexe des réseaux téléinformatiques, et regroupe un ensemble de problématiques de différents niveaux : *comment faire transiter physiquement le signal sur le support ?*, *comment faire cohabiter plusieurs transmissions simultanément ?*, *comment optimiser la durée d'occupation du réseau ?*, etc.

2.6.1 Transmission du signal sur le support

Au plus bas niveau de la transmission des informations sur un réseau, on trouve le support de transmission. Et quelque soit le type utilisé, la contrainte reste toujours la même : il faut faire transiter des informations numériques sur un média analogique, de la manière la plus optimale qui soit. Cette opération est à la charge de l'élément de connexion au réseau (ETCD).

2.6.1.1 Bande de base

Le terme **bande de base** [3] [4] désigne le cas où le signal est transmis sans modifier la bande de fréquences utilisée.

Les principes de cette méthode d'accès au réseau sont simples. Un signal binaire doit transiter ; on fait correspondre 2 niveaux de tension fixes distincts correspondant chacun à un niveau logique.

La première idée est donc de travailler de manière identique à la logique TTL en électronique [5], appelé RZ (Return to Zero [6]) : 0V pour le niveau 1, et $+V_1$ pour le niveau 0 (avec V_1 à 5V par exemple).

Ex. :

[1] Perturbations des lignes entre elles sur la nappe.
[2] Quand commence la transmission ? Quand finit-elle ?
[3] Comprendre *bande de fréquence de base*.
[4] Bande de base (fr) ≡ baseband (eng).
[5] Logique généralement inversée (afin qu'un problème de transmission puisse être distingué d'une absence de transmission).
[6] Return to Zero (eng) ≡ Retour à Zéro (fr) : le signal est susceptible d'être à 0V lors d'une transmission.

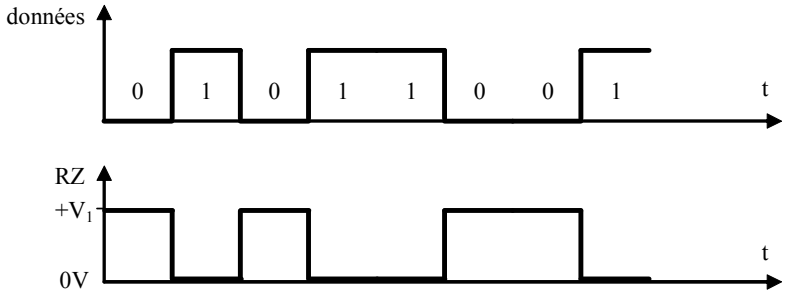

Figure 2.9 : exemple d'une transmission à méthode d'accès bande de base avec un codage RZ

Cependant un tel codage possède un inconvénient, qui est la composante continue [1], signal de fréquence 0, alors que les supports de transmission possèdent généralement une bande passante limitée en basses fréquences [2] ; le signal risque donc de souffrir de distorsion. De plus, aucune distinction ne peut être faite entre le niveau 1 et l'absence de signal, donc la synchronisation est délicate.

L'idée est alors d'utiliser un codage à 3 niveaux, appelé NRZ (Non Return To Zero [3]) : $-V_1$ pour le niveau 1, $+V_1$ pour le niveau 0 (avec V_1 à 5V par exemple).

Ex. :

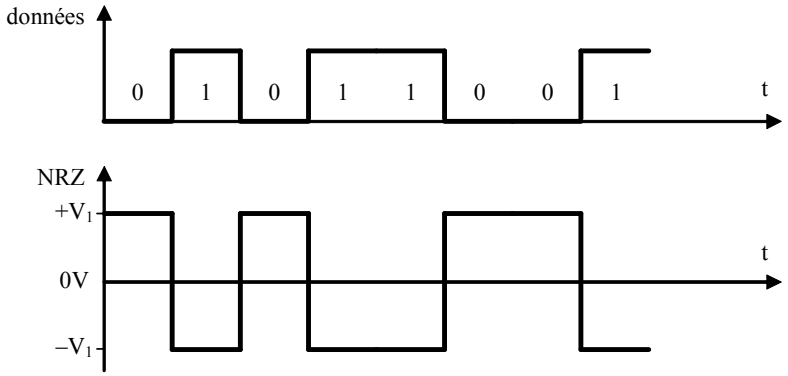

Figure 2.10 : exemple d'une transmission à méthode d'accès bande de base avec un codage NRZ

Néanmoins, le codage NRZ conserve toujours une composante continue, même si plus faible. De plus, toute inversion de polarité induit une inversion de l'information.

Le codage NRZI (Non Return to Zero Inverted on space [4]) pallie le problème de l'inversion éventuelle de polarité en détectant les changements de valeur de bit : un 0 logique inverse le signal, et un 1 logique le stabilise dans son état [5].

Ex. : On suppose que l'état au repos est l'état « haut » (1 logique).

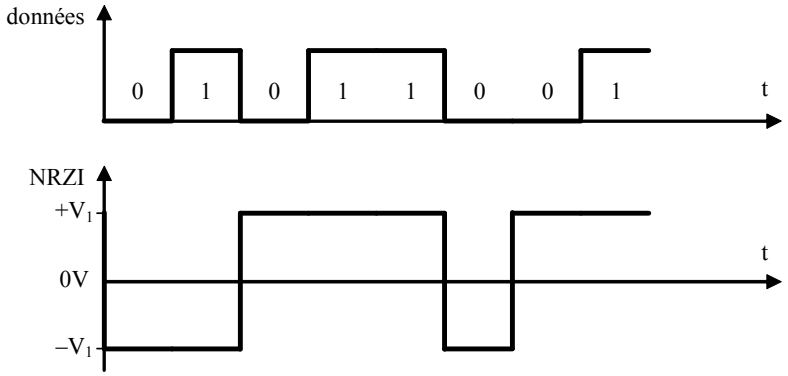

Figure 2.11 : exemple d'une transmission à méthode d'accès bande de base avec un codage NRZI

[1] Avec 2 niveaux en 0V et +5V, la valeur moyenne ne peut jamais être nulle.
[2] Ex. : bande passante de 300 – 3400 Hz pour la ligne téléphonique.
[3] Non Return to Zero (eng) ≡ Non Retour à Zéro (fr) : le signal n'est jamais à 0V lors d'une transmission.
[4] Non Return to Zero Inverted on space (eng) ≡ Non Retour à Zéro Inversé sur espace (fr), noté parfois NRZI-S (nda : espace = 0 logique).
[5] Ce type de codage, à inversion/stabilisation, est dit « différentiel ».

Malgré tout, les codages NRZ et NRZI comportent toujours un risque de désynchronisation lors d'une transmission continue (0 ou de 1 pour NRZ, 1 uniquement pour NRZI), ainsi qu'une synchronisation qui peut être problématique.

Le codage Manchester [1] propose comme solution de diviser la période du signal par 2, afin de symboliser un 1 logique par un front montant, et un 0 logique par un front descendant [2].

Ex. :

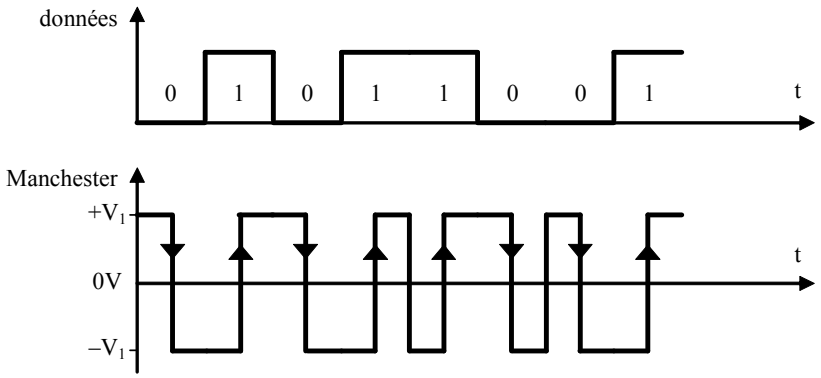

Figure 2.12 : exemple d'une transmission à méthode d'accès bande de base avec un codage Manchester

Avec un tel codage, il n'y a aucune composante continue et la valeur moyenne du signal est donc nulle ; le signal est donc peu sensible au filtrage en basses fréquences, et peut transiter plus facilement via tout type de média.

Tout comme le codage NRZ, le codage Manchester est sensible aux inversions de polarité ; à la place on peut donc utiliser le codage Manchester différentiel [3], qui détecte les changements de valeurs de bit : un 0 logique inverse le sens du front précédent, alors qu'un 1 logique conserve son sens.

Ex. :

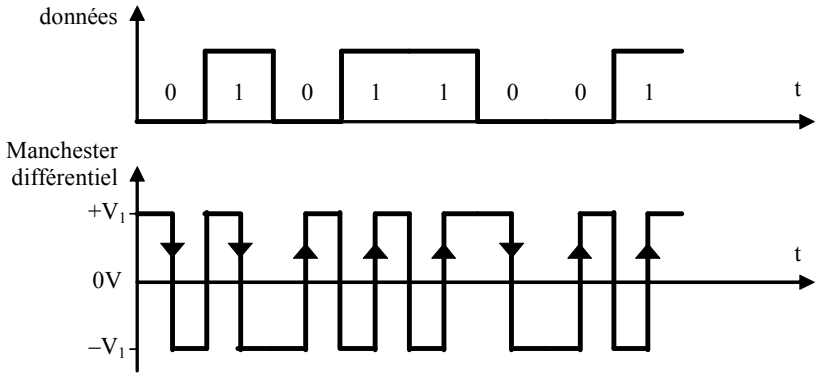

Figure 2.13 : exemple d'une transmission à méthode d'accès bande de base avec un codage Manchester différentiel

Avec un tel codage, tous les avantages sont réunis : détection de la présence/absence de signal, insensibilité aux inversions de polarité, meilleure synchronisation même en cas de série de 0 ou 1, peu de spectre en basses fréquences.

2.6.1.2 Large bande

La transmission en bande de base ne peut pas être mise en œuvre dans tous les cas, notamment lorsque le signal doit transiter sur des supports de transmission prévus pour les signaux analogiques à haute vitesse et/ou longue distance (lignes téléphoniques, faisceaux hertziens, etc.).

Par conséquent, il faut transformer le signal numérique en signal analogique type alternatif par modulation avec une porteuse analogique ; on obtient ainsi un seul signal analogique mais porteur d'informations de par modification de ses caractéristiques dans le temps (fréquence, amplitude, phase).

Ce type de transmission est dite alors **large bande** [4] [5].

[1] Appelé aussi codage bi-phase.
[2] Ce qui revient à faire un OU exclusif entre l'horloge et le codage NRZ.
[3] Aussi appelé codage delay mode ou codage Miller.
[4] Comprendre *large bande de fréquences*.
[5] Large bande (fr) ≡ broadband (eng).

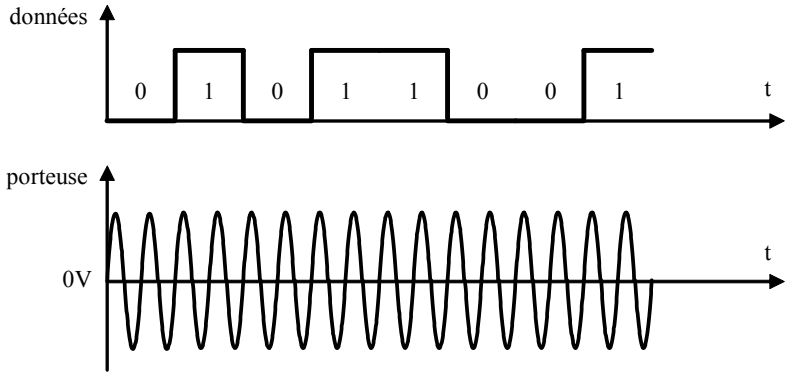

Figure 2.14 : exemple d'une porteuse analogique de modulation d'un signal numérique transmis en méthode large bande

On peut effectuer une modulation en fréquence [1], c'est-à-dire modifier la fréquence de la porteuse en fonction du signal numérique : le signal résultant aura une fréquence différente selon que le signal doit représenter un 0 logique ou un 1 logique.

Ex. :

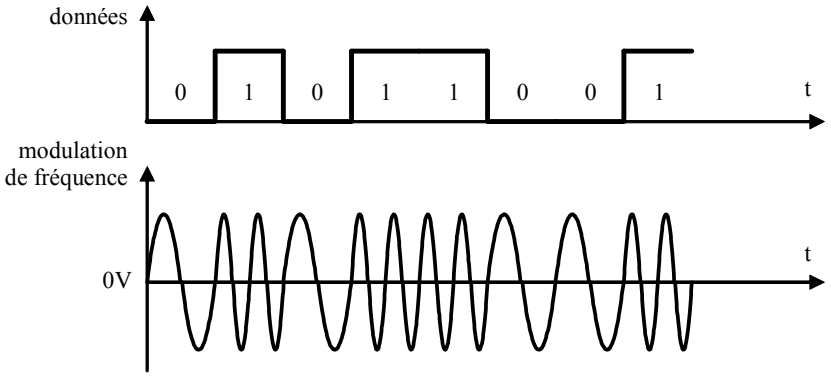

Figure 2.15 : exemple d'une transmission à méthode d'accès large bande avec modulation de fréquence

On peut aussi effectuer une modulation en amplitude [2], c'est-à-dire modifier l'amplitude de la porteuse en fonction du signal numérique : le signal résultant aura une amplitude différente selon que le signal doit représenter un 0 logique ou un 1 logique.

Ex. :

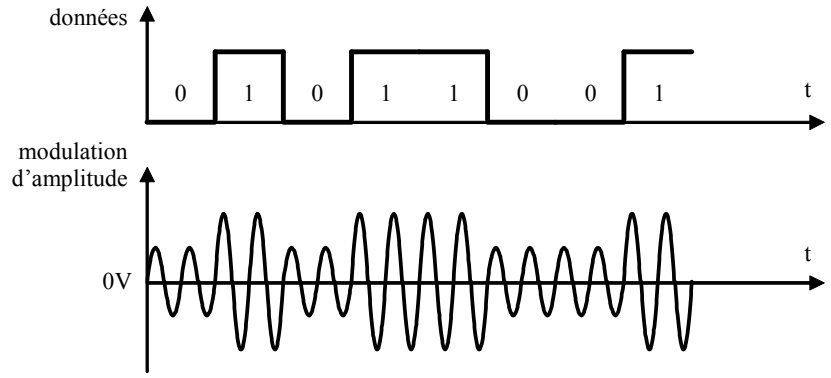

Figure 2.16 : exemple d'une transmission à méthode d'accès large bande avec modulation d'amplitude

La modulation d'amplitude est plus facile à mettre en œuvre, mais la modulation de fréquence, moins sensible aux perturbations électromagnétiques et au bruit [3], possède un meilleur rapport signal/bruit.

[1] Modulation de Fréquence (fr) ≡ Frequency Modulation (eng) : FM ; dans le cas d'un signal de départ numérique, la modulation est appelée FSK (Frequency Shift Keying).

[2] Modulation d'Amplitude (fr) ≡ Amplification Modulation (eng) : AM ; dans le cas d'un signal de départ numérique, la modulation est appelée ASK (Amplitude Shift Keying).

[3] Ayant une incidence sur l'amplitude, mais pas sur la fréquence.

Les appareils capables de moduler un signal pour le transmettre, et de le démoduler pour le recevoir sont communément appelés des modems [1].

Nb : On notera aussi la modulation de phase [2], mais qui n'est pas utilisée dans le cadre des réseaux informatiques.

La modulation de fréquences, qui introduit un décalage fréquentiel du signal de communication, produit un phénomène de dédoublement du spectre autour de la fréquence de la porteuse, de manière strictement symétrique. Généralement on filtre la partie basse du spectre.

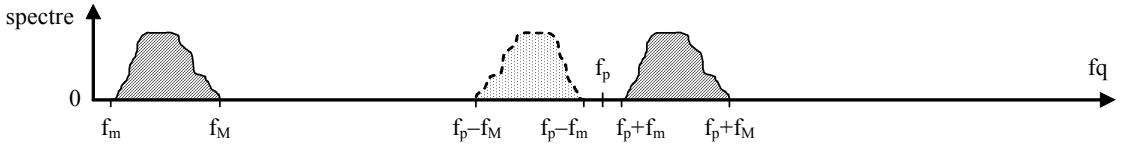

Figure 2.17 : modification de la bande spectrale lors de la modulation

2.6.1.3 Exemples

Liaison série RS-232 : NRZ ;
USB : NRZI ;
Ethernet : Manchester ;
Token Ring : Manchester différentiel ;
Fibre optique : modulation d'amplitude ;
Téléphonie, ADSL : modulation de fréquence.

2.6.2 Multiplexage

Le multiplexage est une opération qui consiste à rassembler plusieurs communications différentes et à les faire transiter sur le même support de transmission. Pour que ces transmissions puissent s'effectuer sans perturbation mutuelle, il faut modifier les caractéristiques des différents signaux de communication.
Deux techniques sont employées : le multiplexage fréquentiel, et le multiplexage temporel.

2.6.2.1 Multiplexage fréquentiel

Le **multiplexage fréquentiel** [3] permet de faire transiter différentes communications sur un même support en modulant chacun d'entre eux indépendamment avec une porteuse de fréquence différente. Ainsi chaque signal occupe une bande de fréquences propre, ne venant pas perturber les autres signaux transitant via le support.
De par son principe, le multiplexage fréquentiel ne peut être utilisé que dans le cas des transmissions large bande.

Ex. :

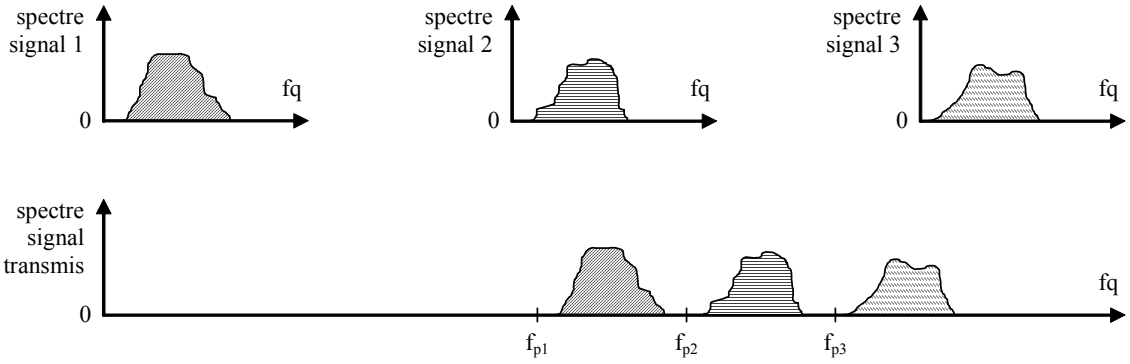

Figure 2.18 : exemple de multiplexage fréquentiel

À l'extrémité de la transmission, chacun des différents signaux de communication est récupéré par filtrage passe-bande et démodulation.

[1] MODulation / DEModulation.
[2] Modulation de Phase (fr) ≡ Phase Modulation (eng) : PM ; dans le cas d'un signal de départ numérique, la modulation est appelée PSK (Phase Shift Keying).
[3] Aussi appelé MRF : Multiplexage par Répartition de Fréquence (fr) ≡ FDM : Frequency Division Multiplexing (eng).

2.6.2.2 Multiplexage temporel

Le **multiplexage temporel** [1] permet de faire transiter différentes communications sur un même support en allouant successivement et cycliquement le support de transmission aux différents signaux devant transiter.

Ce principe rend cette technique impossible à mettre en œuvre pour les signaux analogiques, mais est en revanche parfaitement adaptée à la transmission de signaux numériques. Elle est généralement utilisée pour les transmissions en bande de base.

Ex. :

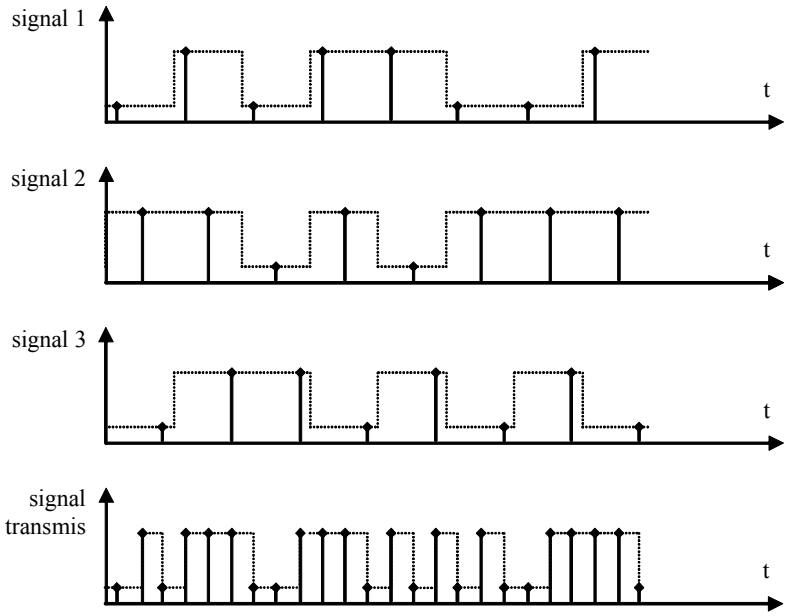

Figure 2.19 : exemple de multiplexage temporel

Pour récupérer chaque signal indépendamment, il faut relever les échantillons de bits au bon moment de manière cyclique, imputant ainsi un léger temps de retard [2].

2.6.3 Gestion des erreurs de transmission

Les sources d'erreurs de transmission [3] sont nombreuses (support de transmission physique, perturbations électromagnétiques, etc.). Pour assurer la fiabilité de la transmission, il faut donc mettre en place un mécanisme de détection d'erreur, lequel peut éventuellement être capable de corriger aussi l'erreur.

2.6.3.1 Contrôles de parité

Le **contrôle de parité** a pour principe de rajouter en supplément un bit de vérification à plusieurs bits de données afin de comptabiliser le nombre total de bits à 1 ; du caractère pair / impair de ce nombre total de bits dépend la valeur du bit de vérification.

Deux contrôles de parité existent et se complètent : la parité verticale et la parité longitudinale.

La parité verticale consiste à rajouter un bit de parité [4] à un groupe de bits de données – en général 7, formant ainsi un octet. On parle de *parité paire* lorsque le nombre total de bits (données + bit parité) est pair ; de même pour la *parité impaire*.

Ex. : 0101100, il y a trois « 1 » ; le bit de parité sera 1 en cas de parité paire, 0 en cas de parité impaire ; l'octet transmis sera alors respectivement 01011001 ou 01011000.

En cas d'erreur de transmission, le bit de parité ne correspond pas ; on détecte donc une erreur de transmission sur un groupe de bits. Cependant, il est impossible de localiser la source d'erreur, et de fait de la corriger. De plus, si l'erreur de transmission a affecté 2 bits différents, elle n'est pas détectée.

---

[1] Aussi appelé MRT : Multiplexage par Répartition dans le Temps (fr) ≡ TDM : Time Division Multiplexing (eng).
[2] Néanmoins inférieur à la durée d'émission d'un bit.
[3] Taux d'erreur moyen entre 10^{-5} et 10^{-7}.
[4] Appelé VRC : Vertical Redundancy Check (eng) ≡ Contrôle de Redondance Verticale (fr).

La parité longitudinale consiste à rajouter un octet de parité [1] à un bloc d'octets dont chacun s'est vu appliqué un contrôle de parité verticale. Là aussi, la parité peut être paire ou impaire.

Ex. : Un bloc de 3 octets transmis selon des parités verticale et longitudinale impaires.
0101100 0
1101011 0
1100101 1
1011101 0 : LRC

L'utilisation conjuguée de ces deux contrôles de parité permet de détecter l'erreur et de la situer selon un principe ligne/colonne. En revanche, on reste limité à une détection/localisation/correction de 1 erreur par ligne et par colonne.

2.6.3.2 Contrôle de redondance cyclique

Le **Contrôle de Redondance Cyclique** [2], ou CRC, consiste à rajouter à un bloc d'octets un code de contrôle contenant des éléments redondants du bloc. L'usage d'un CRC permet de manière bien plus optimale de détecter et corriger les erreurs de transmission.

Pour mettre en œuvre un CRC, il faut considérer les données, disponibles sous forme de blocs de bits, comme des polynôme binaires. Le code de contrôle est alors obtenu par division polynomiale des données à envoyer, constituées de (m) bits, par un polynôme prédéfini et choisi judicieusement pour ses propriétés, appelé polynôme générateur.

Ex. : Soit le bloc de bits 1101, le polynôme binaire correspondant est $1*X^3 + 1*X^2 + 0*X^1 + 1*X^0 = X^3 + X^2 + 1$.

Le polynôme générateur, constitué de (n) bits, est lui aussi assimilé à un polynôme binaire et permet de déterminer le nombre de bits du code de contrôle obtenu au final, qui correspond au degré du polynôme générateur (puissance la plus élevée de X), soit donc (n) bits.

Ex. : Soit le polynôme générateur 11, le polynôme binaire correspondant est X + 1, degré 1, et le CRC sera de 1 bit.

Le message final alors transmis sera le bloc d'octets suivi du CRC obtenu, soit donc une trame de (m+n) bits. C'est ce message final avec un CRC par défaut initialisé à 0 qui sert de base pour calculer le CRC final.
La division polynomiale se réalise comme une division euclidienne, ce qui est similaire en binaire à un OU exclusif.

Ex. : Soit le bloc d'octets 1101, à diviser par le polynôme générateur 11, pour obtenir un CRC de 1 bit, initialisé à 0.
11010
11
00010
11
1 : CRC obtenu
Le bloc d'octets finalement transmis sera donc : 1101 1.

Le récepteur vérifie alors l'exactitude des données en divisant le bloc de bits reçu par le même polynôme générateur que celui utilisé par l'émetteur. Une erreur est détectée si le reste est non-nul ; à l'inverse, un reste nul n'indique pas obligatoirement une absence d'erreur, mais une forte probabilité.

Ex. : 11011
11
00011
11
0

Un certain nombre de polynômes générateurs normalisés permettent des détections et corrections d'erreurs très efficaces grâce à leurs propriétés [3] :

- parité : $X + 1$;
- LRC sur 1 octet : $X^8 + 1$;
- HEC (ATM) : $X^8 + X^2 + X + 1$;
- CRC-16 (USB) : $X^{16} + X^{15} + X^2 + 1$;
- CRC CCITT v41 (Bluetooth, IrDA, HDLC) : $X^{16} + X^{12} + X^5 + 1$;
- CRC-32 (Ethernet) : $X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$.

[1] Appelé LRC : Longitudinal Redundancy Check (eng) ≡ Contrôle de Redondance Longitudinale (fr).
[2] Contrôle de Redondance Cyclique (fr) ≡ Cyclic Redundancy Check (eng).
[3] Démontrables mathématiquement mais qui ne sont pas l'objet du présent document.

2.6.4 Synchronisation

Pour la bonne transmission des signaux en bande de base, il est important de définir des règles sur l'accès au support de transmission [1] pour la lecture et l'écriture de données afin d'éviter les collisions et de synchroniser l'émetteur et le récepteur.

2.6.4.1 Protocole Xon / Xoff

Le **protocole Xon / Xoff** se base sur le principe que chaque nœud possède un tampon [2] d'émission et un tampon de réception. Lorsque le buffer de réception du récepteur est plein ou s'en approche, celui-ci envoie le caractère Xoff (0x13) à l'émetteur afin de mettre en pause l'émission ; lorsque ce buffer est vide ou s'en approche, le récepteur envoie le caractère Xon (0x11) à l'émetteur afin de poursuivre l'émission.

On s'assure ainsi d'une parfaite synchronisation entre l'émetteur et le récepteur afin d'optimiser la durée de la transmission et d'éviter les pertes de données.

2.6.4.2 Protocole ETX / ACK

Le **protocole ETX / ACK** est proche du protocole Xon / Xoff. L'émetteur envoie les données par blocs ; chaque bloc de données se termine par l'envoi du caractère ETX (End of TeXt, 0x03). Lorsque le récepteur reçoit ce caractère, il peut alors vider son tampon de réception et traiter le bloc de données reçu. Lorsque ce tampon est vide, le récepteur envoie le caractère ACK (ACKnowledge, 0x06) à l'émetteur afin qu'il lui envoie le bloc suivant.

Nb : Ce protocole suppose que le tampon d'émission de l'émetteur ne soit pas d'une taille supérieure au tampon de réception du récepteur.

2.6.4.3 Protocole CSMA

Le protocole **CSMA** (Carrier Sense Multiple Access [3]), basé sur le protocole Aloha, utilise le principe d'*écouter avant d'émettre*.

Un nœud qui désire émettre se met en position d'écoute pour savoir si une transmission est en cours sur le support de transmission partagé (MA) en détectant la présence d'une porteuse (CS) ; si ce n'est pas le cas alors le nœud peut émettre, sinon il se met en attente pour une durée aléatoire.

Néanmoins, des collisions peuvent survenir lorsque deux nœuds décident d'émettre en même temps ou à cause des temps de propagation (un autre nœud peut avoir déjà commencé à émettre sans que le nœud ne soit en mesure de le détecter).

Le principe de ce protocole fait que la durée d'une transmission peut être très rapide ou très longue si beaucoup de collisions surviennent ; il est donc probabiliste.

Ce protocole est la base de toute une famille de protocoles qui proposent une solution au problème des collisions et s'adaptent à diverses situations pratiques : CSMA/CD, CSMA/CA, CSMA/BA, CSMA/CR, CSMA/CP, CSMA/CARP.

2.6.4.4 Protocole CSMA/CD

Le protocole **CSMA/CD** (Carrier Sense Multiple Access w/ Collision Detection [4]) utilise le principe d'*écouter avant d'émettre / écouter pendant l'émission*.

Basé sur le protocole CSMA, un nœud désirant émettre écoute le réseau pour savoir si aucune transmission n'est en cours ; auquel cas le nœud peut émettre, sinon il attend.

Pour pallier le problème des collisions, chaque nœud est aussi à l'écoute pendant sa propre émission afin de détecter une éventuelle collision (CD) ; auquel cas, le nœud se met alors en position d'attente durant une durée variable et aléatoire et refait ensuite une tentative d'émission.

Nb : Pour que le nœud émetteur détecte une collision, la durée d'émission du message doit être supérieure au temps maximal d'aller/retour entre deux nœuds [5].

L'utilisation du protocole CSMA/CD suppose que les éléments matériels mis en œuvre soient capables de recevoir et d'écouter en même temps, et que l'hypothèse suivante peut être vérifiée : *si l'émetteur ne détecte aucune diffusion de transmission, alors c'est aussi le cas pour le récepteur*.

---

[1] On parle du contrôle sur l'accès au média (MAC : Medium Access Control).
[2] Zone de mémoire temporaire / buffer.
[3] Carrier Sense Multiple Access (eng) ≡ Accès Multiples par Écoute de Porteuse (fr).
[4] Carrier Sense Multiple Access w/ Collision Detection (eng) ≡ Accès Multiples par Écoute de Porteuse avec Détection des Collisions (fr).
[5] Ce qui explique la taille minimale imposée de 64 octets d'une trame Ethernet (fonction du temps de propagation et du débit).

Néanmoins, il est très apprécié pour sa facilité de déploiement, son coût, et a l'avantage de laisser l'initiative d'émission à chaque nœud, sans contraintes extérieures (comme c'est le cas des principes maître/esclaves, jeton, etc.).

2.6.4.5 Protocole CSMA/CA

Le protocole **CSMA/CA** (Carrier Sense Multiple Access w/ Collision Avoidance [1]) utilise le principe d'*écouter avant d'émettre / émettre pendant un temps prédéfini*.

Basé sur le protocole CSMA, un nœud désirant émettre écoute le réseau pour savoir si aucune transmission n'est en cours, mais il écoute aussi le réseau pour s'assurer qu'aucune demande d'émission n'est diffusée ; auquel cas le nœud peut émettre, sinon il attend.

Avant de débuter l'émission, le nœud émetteur diffuse un premier message annonçant qu'il va prendre la parole, incitant ainsi les autres nœuds à ne pas émettre (CA). Ce protocole peut être optimisé si l'émetteur diffuse un message RTS (Request To Send [2]) spécifiant aux autres nœuds la durée d'exploitation de la transmission sur le point d'être réalisée ; l'émission ne démarre qu'une fois le message CTS (Clear To Send [3]) reçu du récepteur.

Ce protocole, moins performant que CSMA/CD, est utilisé principalement lorsque ce dernier ne peut être appliqué (cas où il est impossible d'être à l'écoute durant une émission, limitation généralement technologique).

2.6.4.6 Protocole CSMA/BA

Le protocole **CSMA/BA** (Carrier Sense Multiple Access w/ Bitwise Arbitration [4]) utilise le principe d'*écouter avant d'émettre / émettre si prioritaire*.

Basé sur le protocole CSMA, un nœud désirant émettre écoute le réseau pour savoir si aucune transmission n'est en cours ; auquel cas le nœud peut émettre, sinon il attend.

Chaque nœud possède un niveau de priorité. Lors d'une collision, c'est le nœud de priorité la plus élevée, parmi ceux qui ont subi la collision, qui a le droit d'émettre (BA) ; les autres nœuds se mettent en attente pendant une durée aléatoire.

Ce protocole, plus performant que CSMA/CD (lors d'une collision, l'une des transmissions est poursuivie), nécessite la mise en place d'éléments matériels qui gèrent la priorité, et est donc plus coûteux.

2.6.4.7 Protocole CSMA/CR

Le protocole **CSMA/CR** (Carrier Sense Multiple Access w/ Collision Resolution [5]) utilise le principe d'*écouter avant d'émettre / ne pas émettre si non-prioritaire*.

Basé sur le protocole CSMA, un nœud désirant émettre écoute le réseau pour savoir si aucune transmission n'est en cours ; auquel cas le nœud peut émettre, sinon il attend.

Le support de transmission est tel que la transmission d'un 1 est prioritaire sur la transmission d'un 0 et masque celui-ci. Les nœuds étant aussi à l'écoute lorsqu'ils émettent, le ou les nœuds temporairement « prioritaires » ne détectent donc pas de collision et poursuivent leur transmission (CR) ; les autres nœuds détectent la collision et se mettent en attente pendant une durée aléatoire.

Ce protocole est peu utilisé car il nécessite un support de transmission numérique pour la gestion du 1 prioritaire sur le 0, donc assez coûteux.

2.6.4.8 Jeton

La technique d'accès par **jeton** consiste à réglementer les communications en ne permettant qu'au nœud possédant le « jeton » d'émettre. Le jeton, sorte de laissez-passer virtuel, circule sur le réseau en passant de nœud en nœud selon une cadence déterminée ; un nœud désirant émettre doit donc attendre d'obtenir le jeton afin d'y être autorisé. Auquel cas, les données sont associées au jeton, et celui-ci reprend son chemin ; les données sont dissociées du jeton par le destinataire lorsqu'il les reçoit ou bien par l'émetteur lorsqu'elles lui reviennent.

De par son principe, cette technique ne peut être mise en œuvre que sur un réseau en anneau ou en boucle, met offre en contrepartie un fonctionnement déterministe.

---

[1] Carrier Sense Multiple Access w/ Collision Avoidance (eng) ≡ Accès Multiples par Écoute de Porteuse avec Évitement des Collisions (fr).
[2] Demande d'émettre.
[3] Libre d'émettre.
[4] Carrier Sense Multiple Access w/ Bitwise Arbitration (eng) ≡ Accès Multiples par Écoute de Porteuse avec Arbitrage sur Valeur de bit (fr).
[5] Carrier Sense Multiple Access w/ Collision Resolution (eng) ≡ Accès Multiples par Écoute de Porteuse avec Résolution des Collisions (fr).

2.6.4.9 Exemples

Liaison série : Xon/Xoff, ETX/ACK ;
Ethernet : CSMA/CD ;
Transmission sans fil (WiFi, WiMAX, etc.) : CSMA/CA ;
CAN : CSMA/BA ;
Token Ring, FDDI : jeton ;
LocalTalk (Apple) : CSMA/CA ;
ALOHAnet (sans fil) : CSMA ;
RNIS (ISDN) : CSMA/CR ;
CPL (Courant Porteur en Ligne) : CSMA/CA.

2.6.5 Commutation de données

La **commutation de données** s'intéresse aux problématiques liées au chemin emprunté par une transmission sur les grands réseaux point-à-point, ainsi qu'à son cheminement. Le traitement de ces problématiques est communément appelé *routage*.

Il s'agit donc de déterminer quels nœuds intermédiaires seront utilisés pour la transmission entre le nœud source et le nœud destination, ainsi que la manière dont les données transiteront entre ces nœuds intermédiaires.

2.6.5.1 Commutation de circuits

La **commutation de circuits** consiste à déterminer, avant toute transmission, le chemin entre les 2 nœuds extrêmes. Ce chemin est fonction du taux instantané d'occupation de chaque nœud au moment de la transmission ; une fois fixé, ce chemin reste inchangé pour l'intégralité de la transmission, et celle-ci peut débuter. Les nœuds peuvent alors communiquer, comme s'ils étaient connectés directement l'un à l'autre.

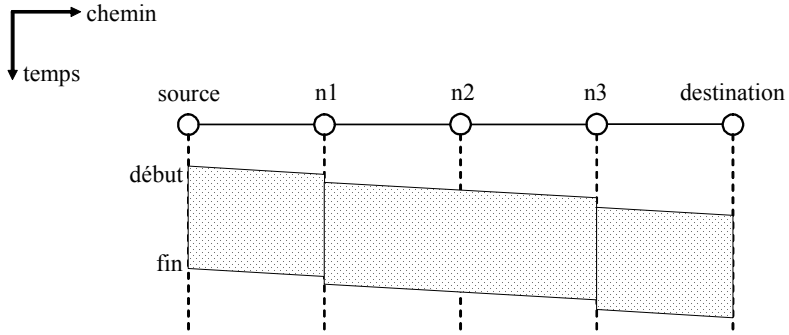

Figure 2.20 : commutation de circuits

De par le caractère figé du chemin emprunté par la transmission, cette technique de commutation est viable pour des transmissions ponctuelles et de courte durée, sinon le risque de dégradation des performances du réseau est important.

2.6.5.2 Commutation de messages

La **commutation de messages** consiste à transmettre, à partir du nœud source, l'intégralité des informations – id est le message – de nœud en nœud en direction du nœud destination. La référence du nœud destination étant incluse dans le message, chaque nœud du chemin a en charge, si le message ne lui est pas destiné, de le re-transmettre dans la direction du destinataire en fonction de ses tables de routage [1] et de la charge instantanée du réseau.

[1] Une table de routage est une liste qui associe un chemin à emprunter (quel nœud-relais ?) à une adresse ou un groupe d'adresses de nœuds à atteindre (cf. 4.5.1).

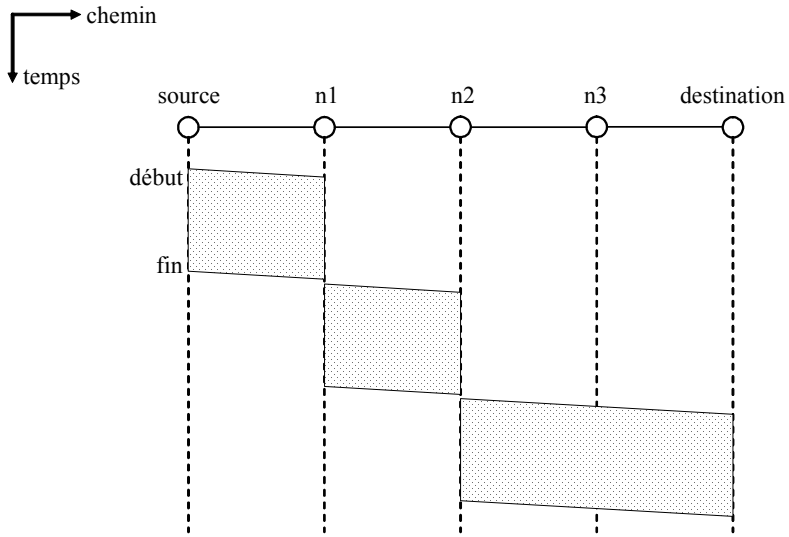

Figure 2.21 : commutation de messages

Un message devant être intégralement stocké par un nœud avant de pouvoir être réémis vers le nœud suivant, son envoi peut être différé si un nœud récepteur est temporairement indisponible. En revanche, les nœuds doivent disposer d'importants moyens de stockage ; de plus en cas d'erreur, l'intégralité du message doit être re-transmis.

Cette technique de commutation tend à disparaître au profit de la commutation de paquets.

2.6.5.3 Commutation de paquets

La **commutation de paquets** suit les mêmes principes que la commutation de messages, mais le message est découpé en morceaux de petite taille [1], appelés paquets, qui sont envoyés séparément sur le réseau, et transitent de manière totalement indépendante.

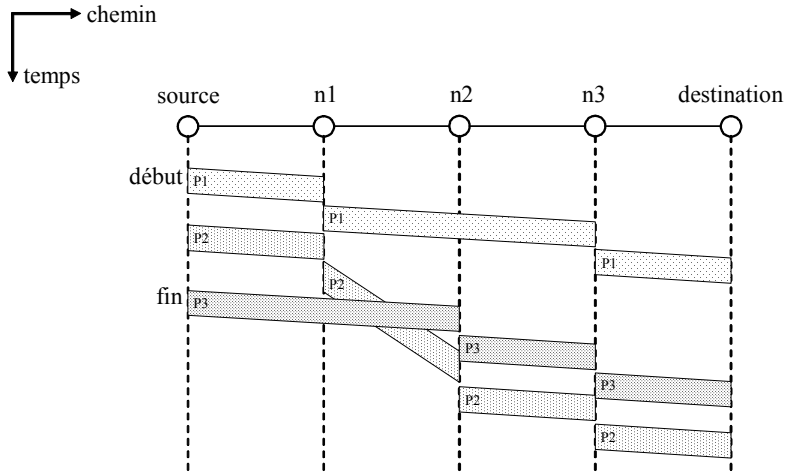

Figure 2.22 : commutation de paquets

Les paquets étant de très petite taille [2], leur stockage est très facile [3] ; de plus, en cas d'erreur, seul le paquet erroné doit être re-transmis. La charge instantanée du réseau pouvant varier rapidement, les paquets n'empruntent pas obligatoirement le même chemin et n'arrivent donc pas nécessairement dans l'ordre d'envoi ; par conséquent, en plus de la référence du nœud destinataire, chaque paquet doit être numéroté pour que le nœud destinataire puisse reconstitué le message originel.

---

[1] Principe de *fragmentation* (cf. 4.3.4).
[2] Exemple : 1500 octets au maximum dans le cas d'Ethernet.
[3] Généralement, la mémoire vive du nœud suffit.

3 LE MODÈLE OSI

Les problématiques liées aux réseaux téléinformatiques sont extrêmement nombreuses. Les moyens, solutions et protocoles permettant d'y répondre sont presque tout aussi nombreux.

Pour pouvoir réutiliser facilement des éléments ayant fait leurs preuves, tout comme conserver un fort potentiel d'évolutivité, ainsi que d'assurer de meilleures compatibilité et interopérabilité entre les diverses technologies, il est nécessaire d'organiser, structurer et hiérarchiser voire standardiser les moyens technologiques associés aux réseaux.

3.1 PRINCIPES

3.1.1 Architecture en couches

La plupart des systèmes de télécommunications sont construits selon une **architecture en couches**, c'est-à-dire une segmentation en plusieurs niveaux, empilés l'un sur l'autre, qui ont chacun des finalités différentes mais participent tous à la transmission de la communication entre plusieurs nœuds.

Chaque couche réalise donc une partie des opérations nécessaires pour y parvenir, et est liée aux 2 couches adjacentes [1] par la notion d'interface, qui est l'ensemble des opérations proposées à la couche supérieure, et utilisées de la couche inférieure. Deux couches de même niveau de deux nœuds différents peuvent ainsi communiquer, en utilisant un protocole spécifique.

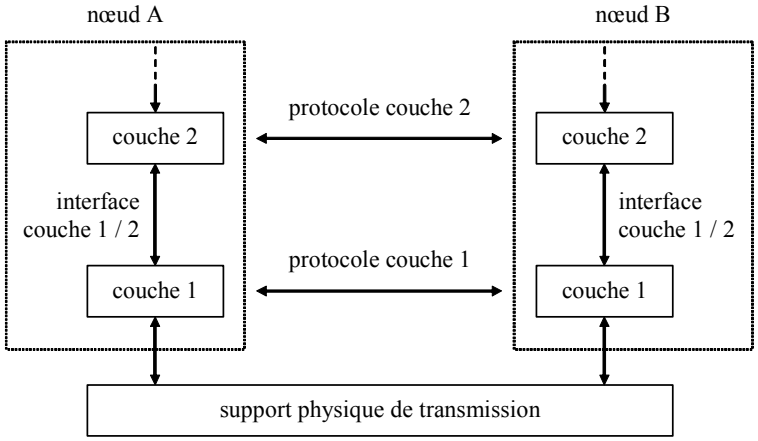

Figure 3.1 : architecture en couches d'un réseau

Le nombre de couches, leur nom, leurs fonctions, etc. peut varier d'un réseau à un autre.

3.1.2 Communication virtuelle

Les couches de même niveau ne communiquent pas directement entre elles, mais via un mécanisme qui est transparent ; on parle de **communication virtuelle**.

Au niveau du nœud émetteur, une donnée émise d'une couche (n) est prise en charge par la couche (n–1) qui réalise diverses opérations et transmet la donnée à la couche (n–2), et ainsi de suite, jusqu'à atteindre la couche la plus basse, laquelle a notamment pour fonction de gérer l'accès au support physique de transmission, et véhicule alors effectivement la donnée jusqu'à l'autre nœud.

[1] Supérieure et inférieure.

La donnée est alors prise en charge par la couche la plus basse du nœud destinataire, qui réalise diverses opérations et la transmet à la couche supérieure, et ainsi de suite, jusqu'à la couche (n) où l'on doit retrouver la donnée telle qu'elle a été émise à partir de cette même couche du nœud émetteur, comme si les couches avaient pu communiquer directement entre elles.

3.1.3 Interface entre deux couches

Le découpage en couches nécessite que chacune d'entre elles propose un ensemble d'opérations à la couche qui lui est supérieure. La manière dont les opérations d'une couche (n) mises à disposition de la couche (n+1) ont été réalisées doit restée méconnue par cette dernière, à la fois car elle n'en a pas besoin, et aussi parce que la manière dont elle réalise ses propres opérations doit en être indépendante ; on parle alors d'**interface**.

Les opérations ainsi mises à la disposition par une couche (n) sont appelées **services** ; la couche (n) est fournisseur de services, alors que la couche (n+1) est utilisateur de services ; la couche (n) est elle-même utilisateur des services de la couche (n–1).

Au niveau du nœud émetteur, lorsqu'une donnée est émise d'une couche (n), celle-ci fait appel à l'un des services de la couche (n–1), et lui transmet une trame constituée de la donnée à émettre et d'informations de contrôle, qui permettront à la couche (n) du destinataire d'exploiter les données reçues (numéro d'identification, CRC, etc.) : ce qui peut être résumé ainsi : *trame ≡ données + informations de contrôle*.

La donnée à transmettre, appelée SDU (Service Data Unit [1]), est donc associée aux informations de contrôle, appelées PCI (Protocol Control Information [2]), par **encapsulation** dans une trame de données, appelée PDU (Protocol Data Unit [3]) ; cette trame est alors considérée par la couche (n–1) comme la donnée qu'elle doit prendre en charge, c'est-à-dire le SDU. Et ainsi de suite : à chaque descente par une couche, la trame de données est donc encapsulée dans une autre trame.

Au niveau du nœud destinataire, le processus d'encapsulation est inversé : à chaque remontée par une couche, une trame de données est extraite de la trame.

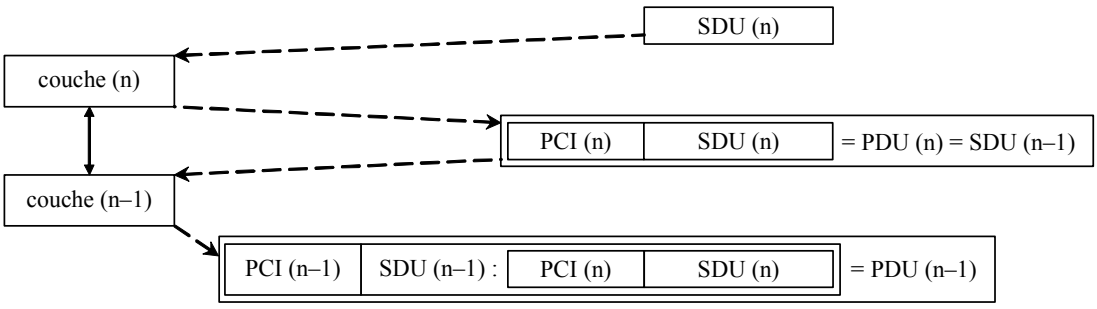

Figure 3.2 : encapsulation d'une donnée pour passage dans une autre couche

On peut synthétiser le mécanisme d'encapsulation réalisé aux interfaces de la façon suivante :

- PDU (n) ≡ SDU (n) + PCI (n) ;
- SDU (n–1) ≡ PDU (n).

3.2 DÉFINITIONS

Le **modèle OSI** (Open Systems Interconnection [4]) est un modèle générique et standard d'architecture d'un réseau en 7 couches, élaboré par l'organisme ISO [5] en 1984 [6]. La mise en évidence de ces différentes couches se base sur les caractéristiques suivantes qui étaient recherchées par l'ISO :

- création d'une couche lorsqu'un niveau d'abstraction est nécessaire ;
- définition précise des services et opérations de chaque couche ;
- définition des opérations de chaque couche en s'appuyant sur des protocoles normalisés ;
- choix des frontières entre couches de manière à minimiser le flux d'information aux interfaces ;
- définition d'une couche supplémentaire lorsque des opérations d'ordre différent doivent être réalisées.

[1] Service Data Unit (eng) ≡ Unité de Donnée du Service (fr).
[2] Protocol Control Information (eng) ≡ Informations de Contrôle du Protocole (fr).
[3] Protocol Data Unit (eng) ≡ Unité de Donnée du Protocole (fr).
[4] Interconnexion de Systèmes Ouverts ; un système ouvert est un système capable de s'interconnecter avec des systèmes de technologie différente.
[5] International Standardization Organisation (eng) ≡ Organisation Internationale de Standardisation (fr).
[6] Dans sa première version.

3.3 DÉCOUPAGE EN COUCHES

Dans le découpage en 7 couches, on distingue :

- les couches basses (1-4) : transfert de l'information par les différents services de transport ;
- les couches hautes (5-7) : traitement de l'information par les différents services applicatifs.

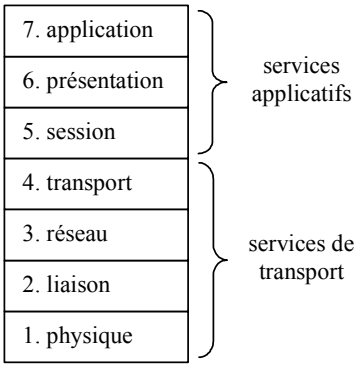

Figure 3.3 : représentation du modèle OSI

3.3.1 Physique

La couche **physique** (physical (eng)) gère la communication avec l'interface physique afin de faire transiter ou de récupérer les données sur le support de transmission, qui peut être électrique, mécanique, fonctionnel ou procédural.

Ce sont les contraintes matérielles du support utilisé qui décident des objectifs à atteindre pour cette couche : conversion en signaux électriques, taille et forme des connecteurs, dimensions et position des antennes, etc.

Ex. : Interconnexion avec le support physique de transmission (paire torsadée, fibre optique, etc.), choix du codage (NRZ, Manchester, modulation AM, FM, etc.), procédure de paramétrage.

Nb : Il est conceptuellement faux de considérer que le support physique de transmission lui-même appartient à cette couche.

3.3.2 Liaison

La couche **liaison** (liaison de données : datalink (eng)) s'occupe de la bonne transmission de l'information entre les nœuds via le support, en assurant la gestion des erreurs de transmission et la synchronisation des données.

Là aussi, le support de transmission conditionne les protocoles à mettre en œuvre.

Ex. : Gestion des erreurs (contrôles de parité, CRC, etc.), synchronisation (Xon/Xoff, CSMA/xx, etc.), multiplexage.

3.3.3 Réseau

La couche **réseau** (network (eng)) a en charge de déterminer le choix de la route entre les nœuds afin de transmettre de manière indépendante l'information ou les différents paquets la constituant en prenant en compte en temps réel le trafic. Cette couche assure aussi un certain nombre de contrôles de congestion qui ne sont pas gérés par la couche liaison.

Ex. : Techniques de commutation de données (circuits, paquets, etc.).

Nb : Dans les réseaux à diffusion, le routage est très simple, la couche réseau est donc minimaliste, voire inexistante.

3.3.4 Transport

La couche **transport** (transport (eng)) supervise le découpage et le réassemblage de l'information en paquets, contrôlant ainsi la cohérence de la transmission de l'information de l'émetteur vers le destinataire.

Ex. : Techniques de commutation par paquets, fragmentation.

3.3.5 Session

La couche **session** (session (eng)) gère une communication complète entre plusieurs nœuds, permettant ainsi d'établir et de maintenir un réel dialogue suivi (/ une session), pouvant être constitué de temps morts pendant lesquels aucune donnée n'est physiquement transmise.

Ex. : Une connexion HTTP avec suivi de navigation sur un même site web (usage des cookies), une connexion FTP.

3.3.6 Présentation

La couche **présentation** (presentation (eng)) a en charge la représentation des données, c'est-à-dire de structurer et convertir les données échangées ainsi que leur syntaxe afin d'assurer la communication entre des nœuds disparates (différences hardware et/ou software [1]).

Ex. : Codage des données (ASCII, Unicode, little-endian, big-endian, etc.), cryptage, compression.

3.3.7 Application

La couche **application** (application (eng)) est le point d'accès des applications aux services réseaux. On y retrouve toutes les applications de communication via le réseau communément utilisées sur un LAN ou sur internet : applications de transfert de fichiers, courrier électronique, etc.

Ex. : Navigateurs (HTTP), transfert de fichiers (FTP), clients email (SMTP).

3.4 TRANSMISSION DE DONNÉES À TRAVERS LE MODÈLE OSI

La transmission de données à travers le modèle OSI utilise le principe de communication virtuelle en usant des interfaces inter-couches. Il y a donc encapsulation successive des données à chaque interface (H : Header, T : Trailer).

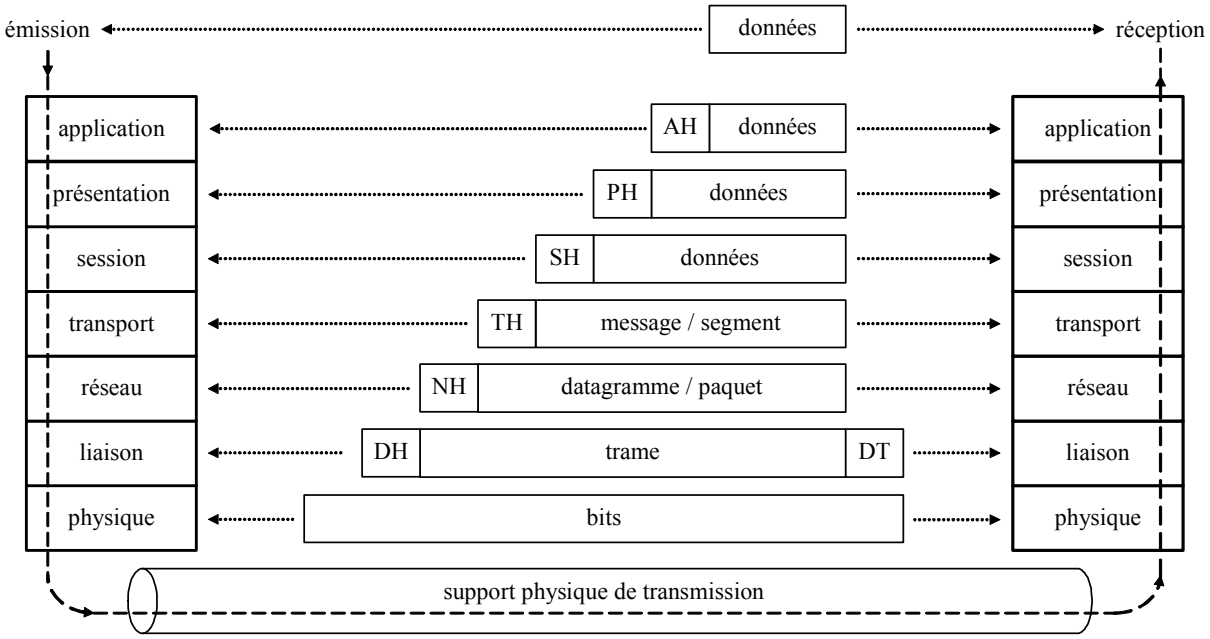

Figure 3.4 : transmission de données à travers le modèle OSI

Il est important de noter que le modèle OSI, reste comme son nom l'indique, un **modèle**, qui n'est pas scrupuleusement respecté, mais vers lequel on tente généralement de se rapprocher dans une optique de standardisation. De plus, ce modèle a été historiquement établi après la mise en place de technologies ayant fait leurs preuves, et il ne peut, de fait, pas toujours être rigoureusement suivi ; c'est le cas du protocole TCP/IP.

[1] Microprocesseurs, systèmes d'exploitation différents, etc.

4 LE PROTOCOLE TCP/IP

4.1 DÉFINITIONS

Le **protocole TCP/IP**, développé originellement par le ministère de la défense américain en 1981, propose l'évolution de concepts déjà utilisés en partie pour le réseau historique ARPAnet (1972), et est employé en très forte proportion sur le réseau internet. Au-delà de son aspect historique, TCP/IP doit aussi son succès à son indépendance vis-à-vis de tout constructeur informatique.

En réalité, TCP/IP définit une suite de divers protocoles probabilistes, appelé aussi *modèle DOD* (Department of Defense), pour la communication sur un réseau informatique, notamment le protocole TCP et le protocole IP qui sont parmi les principaux protocoles de ce modèle.

4.2 TCP/IP ET LE MODÈLE OSI

4.2.1 Découpage en couches

Le protocole TCP/IP étant antérieur au modèle OSI, il ne respecte pas réellement celui-ci. Cependant, on peut faire grossièrement correspondre les différents services utilisés et proposés par TCP/IP avec le modèle OSI, et obtenir ainsi un modèle en 4 couches.

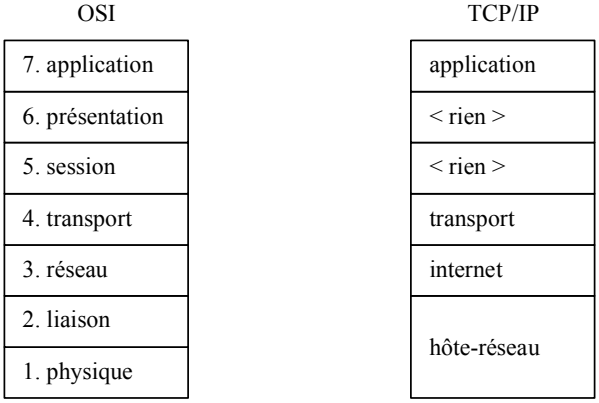

Figure 4.1 : représentation du modèle TCP/IP

Les services des couches 1 et 2 (physique et liaison) du modèle OSI sont intégrés dans une seule couche (hôte-réseau) ; les couches 5 et 6 (session et présentation) n'existent pas réellement dans le modèle TCP/IP et leurs services sont réalisés par la couche application si besoin est.

4.2.1.1 Hôte-réseau

La couche **hôte-réseau**, intégrant les services des couches physique et liaison du modèle OSI, a en charge la communication avec l'interface physique afin de transmettre ou de récupérer les paquets de données qui lui sont transmis de la couche supérieure. Le protocole utilisé pour assurer cet interfaçage n'est pas explicitement défini puisqu'il dépend du réseau utilisé ainsi que du nœud (Ethernet en LAN, X25 en WAN, …).

4.2.1.2 Internet

La couche **internet**, correspondant à la couche réseau du modèle OSI, s'occupe de l'acheminement, à bonne destination, des paquets de données indépendamment les uns des autres, soit donc de leur routage à travers les différents nœuds par rapport au trafic et à la congestion du réseau. Il n'est en revanche pas du ressort de cette couche de vérifier le bon acheminement.

Le protocole IP (Internet Protocol) assure intégralement les services de cette couche, et constitue donc l'un des points-clefs du modèle TCP/IP. Le format et la structure des paquets IP sont précisément définis.

4.2.1.3 Transport

La couche **transport**, pendant de la couche homonyme du modèle OSI, gère le fractionnement et le réassemblage en paquets du flux de données à transmettre. Le routage ayant pour conséquence un arrivage des paquets dans un ordre incertain, cette couche s'occupe aussi du réagencement ordonné de tous les paquets d'un même message.

Les deux principaux protocoles pouvant assurer les services de cette couche sont les suivants :

- TCP (Transmission Control Protocol) : protocole fiable, assurant une communication sans erreur par un mécanisme question/réponse/confirmation/synchronisation (orienté connexion) ;
- UDP (User Datagram Protocol) : protocole non-fiable, assurant une communication rapide mais pouvant contenir des erreurs en utilisant un mécanisme question/réponse (sans connexion).

4.2.1.4 Application

La couche **application**, similaire à la couche homonyme du modèle OSI, correspond aux différentes applications utilisant les services réseaux pour communiquer à travers un réseau.

Un grand nombre de protocoles divers de haut niveau permettent d'assurer les services de cette couche :

- Telnet : ouverture de session à distance ;
- FTP (File Transfer Protocol) : protocole de transfert de fichiers ;
- HTTP (HyperText Transfer Protocol) : protocole de transfert de l'hypertexte ;
- SMTP (Simple Mail Transfer Protocol) : protocole simple de transfert de courrier ;
- DNS (Domain Name System) : système de nom de domaine ;
- etc.

4.2.2 Suite de protocoles

Le modèle TCP/IP correspond donc à une **suite de protocoles** de différents niveaux participant à la réalisation d'une communication via un réseau informatique. Beaucoup de ces protocoles sont régulièrement utilisés par tous du fait de l'essor d'internet.

Telnet FTP HTTP SMTP DNS	application
TCP UDP	transport
IP ICMP ARP RARP	internet
PPP Ethernet ATM FDDI Token Ring *	hôte-réseau

* protocoles associés à des technologies d'architecture matérielle

Figure 4.2 : suite de protocoles du modèle TCP/IP

On parle aussi de *pile de protocoles* afin de rappeler qu'il s'agit bien d'une architecture en couches, et que les données issues d'un protocole d'une couche sont encapsulées dans un protocole de la couche inférieure. Ainsi, une requête HTTP est transportée dans un segment TCP, lui-même encapsulé dans un datagramme IP, etc.

4.3 LE PROTOCOLE IP

4.3.1 Définitions

Le **protocole IP** (Internet Protocol [1]), assure le service attendu de la couche réseau du modèle TCP/IP. Son rôle est donc de gérer l'acheminement des paquets (issus de la couche transport) entre les nœuds de manière totalement indépendante, même dans le cas où les paquets ont mêmes nœuds source et destination.

[1] Internet Protocol (eng) ≡ Protocole Internet (fr).

Le protocole IP offre un fonctionnement non fiable et sans connexion, à base d'envoi/réception de datagrammes (flux de bits structurés) :

- non fiable : absence de garantie que les datagrammes arrivent à destination ; les datagrammes peuvent être perdus, retardés, altérés ou dupliqués sans que ni la source ou la destination ne le sachent [1] ; on parle de « remise au mieux » (*best effort delivery*) ;
- sans connexion (/ mode non-connecté) : chaque datagramme est traité et donc acheminé de manière totalement indépendante des autres.

Le rôle d'IP étant de déterminer le chemin entre les nœuds source et destination, soit donc déterminer les nœuds intermédiaires, il faut disposer d'un mécanisme permettant d'identifier de manière unique chaque nœud sur le réseau.

Les nœuds intermédiaires sont appelés routeurs, et pour assurer cette communication, ce protocole se base sur ce que l'on appelle l'adresse IP que chaque nœud possède.

4.3.2 L'adressage IP

4.3.2.1 L'adresse IPv4

L'**adresse IP** d'un nœud est l'identifiant logiciel unique de ce nœud sur le réseau par lequel le nœud est directement joignable. Cette adresse, modifiable à volonté par simple configuration logicielle, est codée sur 32 bits regroupés en 4 octets (d'où le nom de IPv4), et est généralement notée *xxx.yyy.zzz.ttt* (dite « notation décimale pointée ») avec *xxx*, *yyy*, *zzz* et *ttt* compris entre 0 et 255 (0x00 et 0xFF). Elle est scindée en deux groupes de bits de taille variable se partageant les 32 bits :

- identifiant de réseau (Network ID, NetID) ;
- identifiant de l'hôte (le nœud) dans le réseau (HostID).

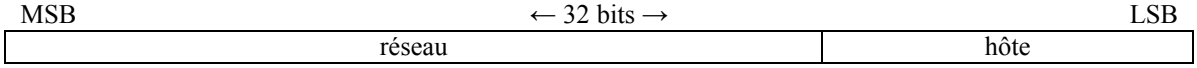

Figure 4.3 : position des identifiants de réseau et d'hôte dans une adresse IPv4

Ainsi, des hôtes possédant le même identifiant de réseau peuvent communiquer directement entre eux [2], car ils sont situés sur le même segment.

Des nœuds d'identifiant réseau différent doivent passer par une interface, qui réalise alors l'interconnexion physique et logique entre les 2 réseaux : une passerelle (généralement un routeur).

Comme la séparation réseau/hôte est variable d'un réseau à l'autre, il faut préciser sa position dans les 32 bits, en associant à l'adresse IP le **masque de réseau** (netmask), codé lui aussi sur 32 bits (séparés en 4 octets) de la manière suivante :

- bit de l'adresse IP définissant le réseau → bit correspondant du masque à 1 ;
- bit de l'adresse IP définissant l'hôte → bit correspondant du masque à 0.

L'identifiant de réseau s'obtient alors en réalisant un masquage bit-à-bit [3] de l'adresse IP avec le masque de réseau, et l'identifiant d'hôte en réalisant un masquage de l'adresse IP avec le complément à 1 du masque de réseau.

Ex. : Soit l'adresse IP 192.168.1.72, associée au masque de réseau 255.255.255.0, abrégée en 192.168.1.72 / 24 (les 24 premiers bits définissent l'identifiant réseau, et donc les 8 restants l'identifiant d'hôte).

	192 . 168 . 1 . 72	
Adresse IP	1100 0000 . 1010 1000 . 0000 0001 . 0100 1000	192.168.1.72
Masque	1111 1111 . 1111 1111 . 1111 1111 . 0000 0000	255.255.255.0
Identifiant réseau	1100 0000 . 1010 1000 . 0000 0001 . 0000 0000	192.168.1.0
Identifiant hôte	0000 0000 . 0000 0000 . 0000 0000 . 0100 1000	0.0.0.72

Même adresse IP, associée au masque de réseau 255.255.255.224, abrégée en 192.168.1.72 / 27.

Adresse IP	1100 0000 . 1010 1000 . 0000 0001 . 010 0 1000	192.168.1.72
Masque	1111 1111 . 1111 1111 . 1111 1111 . 111 0 0000	255.255.255.224
Identifiant réseau	1100 0000 . 1010 1000 . 0000 0001 . 010 0 0000	192.168.1.64
Identifiant hôte	0000 0000 . 0000 0000 . 0000 0001 . 000 0 1000	0.0.0.8

[1] Ce sont les protocoles de plus haut niveau qui doivent donc assurer la fiabilité si celle-ci est nécessitée.
[2] Cela suppose bien entendu que les hôtes sont interconnectés physiquement entre eux.
[3] ET logique.

Il faut noter que l'appellation *hôte* est abusive, et qu'une adresse IP est bien assignée à un nœud, soit donc l'interface de connexion au réseau. Par conséquent, un hôte peut posséder plusieurs interfaces afin d'être connecté à plusieurs réseaux différents et donc posséder autant d'adresses IP [1].

Nb : Le modèle IPv4 arrive à bout de souffle, en effet, pouvant coder « seulement » 2^{32} adresses différentes (≈ $4{,}3.10^{9}$), son champ d'action, prévu au départ pour les universités, les administrations, les gouvernements, les industries de pointe, etc. trouve ses limites avec le formidable essor d'internet [2].

IPv4 se voit donc progressivement remplacé par le modèle IPv6 [3], codé sur 16 octets, et ayant donc un potentiel de 2^{128} adresses différentes (≈ $3{,}4.10^{38}$), dont le but est aussi d'optimiser le protocole par rapport aux besoins actuels qui n'entraient que très peu en considération à la création de IPv4 (sécurisation, hiérarchisation, etc.).

4.3.2.2 Conventions d'adressage IPv4 : classes d'adresses et adresses réservées

Parmi les 2^{32} adresses disponibles avec IPv4, 5 plages d'adresses sont définies, distinguées par rapport aux 4 premiers bits de l'identifiant réseau :

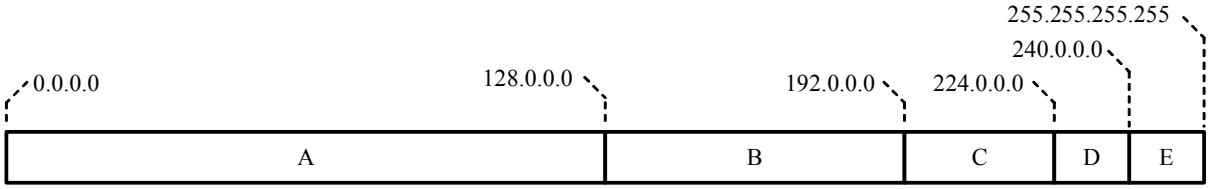

Figure 4.4 : classes d'adresses IPv4

classe	octet de poids fort	plage	usage	capacité
A	0xxx xxxx	0.0.0.0 – 127.255.255.255	unicast classe A	2^{31} adresses
B	10xx xxxx	128.0.0.0 – 191.255.255.255	unicast classe B	2^{30} adresses
C	110x xxxx	192.0.0.0 – 223.255.255.255	unicast classe C	2^{29} adresses
D	1110 xxxx	224.0.0.0 – 239.255.255.255	multicast	2^{28} adresses
E	1111 xxxx	240.0.0.0 – 255.255.255.255	réservé	2^{28} adresses

Les classes d'adresse A, B et C sont des adresses unicast, permettant d'établir une communication point-à-point entre les nœuds source et destination.

La classe D est une classe d'adresses multicast, permettant d'établir une communication multi-points entre les nœuds proposant la même adresse IP multicast [4].

Aux classes d'adresses A, B et C (unicast) on associe un masque différent, qui est généralement standard mais peut être modifié [5] en fonction des besoins en capacités réseaux/hôtes.

classe	plage	masque	adresse IP	capacité
A	0.0.0.0 – 127.255.255.255	255.0.0.0 (/8)	0.x.y.z – 127.x.y.z	2^{7} réseaux, 2^{24} hôtes
B	128.0.0.0 – 191.255.255.255	255.255.0.0 (/16)	128.0.x.y – 191.255.x.y	2^{14} réseaux, 2^{16} hôtes
C	192.0.0.0 – 223.255.255.255	255.255.255.0 (/24)	192.0.0.x – 223.255.255.x	2^{21} réseaux, 2^{8} hôtes

Nb : Le masque n'est pas un élément probant pour la détermination de la classe d'adresses d'une adresse IP (même s'il constitue souvent un bon indice) ; seule la valeur de l'octet de poids fort est fiable.

Chaque classe d'adresses A, B et C (unicast) contient une plage d'adresses privées. Les adresses privées sont des adresses non routables sur internet [6], et sont réservées pour l'adressage dans un cadre privé (particulier, LAN, entreprise, etc.).

classe	plage	réseau	adresse IP	capacité
A	10.0.0.0 – 10.255.255.255	10.0.0.0/8	10.x.y.z – 10.x.y.z	2^{0} réseaux, 2^{24} hôtes
B	172.16.0.0 – 172.31.255.255	172.16.0.0/12	172.16.x.y – 172.31.x.y	2^{4} réseaux, 2^{16} hôtes
C	192.168.0.0 – 192.168.255.255	192.168.0.0/16	192.168.0.x – 192.168.255.x	2^{8} réseaux, 2^{8} hôtes

D'autres conventions existent sur l'adressage IPv4 :

[1] C'est le cas des routeurs, par exemple, dont le but est justement de réaliser l'interconnexion entre plusieurs réseaux.
[2] Échéance de la pénurie d'adresses estimée à 2010.
[3] cf. 4.5.6.
[4] Cette adresse « commune » est donc une adresse de groupe de diffusion.
[5] Autorisé par le système CIDR (Classless Inter Domain Routing) apparu en 1993 afin de pallier la pénurie d'adresses.
[6] Codage hardware et software dans les routeurs connectés à internet.

- adresse IP 0.0.0.0 : adresse par défaut, aucune adresse IP n'est affectée au nœud ;
- adresse IP dont l'identifiant réseau est à 0 (tous ses bits à 0) : adresse IP de la machine dans le réseau courant (/identifiant d'hôte) ;
- adresse IP dont l'identifiant d'hôte est à 0 (tous ses bits à 0) : adresse IP du réseau (/identifiant réseau) [1] ;
- adresse IP dont tous les bits de l'identifiant d'hôte sont à 1 : adresse de diffusion dirigée (net-directed broadcast), s'adressant à tout nœud de même identifiant de réseau [1] (généralement bloquée par les routeurs , mais peut les passer si ceux-ci sont configurés en ce sens).
- adresse IP 127.0.0.0 / 8 (généralement 127.0.0.1) : correspond toujours à l'hôte local (localhost, lo), soit donc la machine locale ; les paquets émis vers cette adresse ne sortent pas physiquement du réseau [2] (loopback[3]) ;
- adresse IP 169.254.0.0 / 16 : adresse par défaut de configuration automatique, en cas d'absence de configuration manuelle et d'obtention d'une adresse par adressage automatique via serveur DHCP [4] ;
- adresse IP 255.255.255.255 : adresse de diffusion limitée (limited broadcast), s'adressant à tout nœud connecté au même segment [1] (bloquée par les routeurs).

Ex. : Soit l'adresse IP 192.168.1.72 / 24.

Adresse IP	1100 0000 . 1010 1000 . 0000 0001 . 0100 1000	192.168.1.72
Masque	1111 1111 . 1111 1111 . 1111 1111 . 0000 0000	255.255.255.0
Identifiant réseau	1100 0000 . 1010 1000 . 0000 0001 . 0000 0000	192.168.1.0
Identifiant hôte	0000 0000 . 0000 0000 . 0000 0000 . 0100 1000	0.0.0.72
Adresse broadcast	1100 0000 . 1010 1000 . 0000 0001 . 1111 1111	192.168.1.255

Soit l'adresse IP 192.168.1.72 / 27.

Adresse IP	1100 0000 . 1010 1000 . 0000 0001 . 010 0 1000	192.168.1.72
Masque	1111 1111 . 1111 1111 . 1111 1111 . 111 0 0000	255.255.255.224
Identifiant réseau	1100 0000 . 1010 1000 . 0000 0001 . 010 0 0000	192.168.1.64
Identifiant hôte	0000 0000 . 0000 0000 . 0000 0001 . 000 0 1000	0.0.0.8
Adresse broadcast	1100 0000 . 1010 1000 . 0000 0001 . 010 1 1111	192.168.7.95

4.3.2.3 Construction de sous-réseaux

La **construction de sous-réseaux** (subnetting) consiste à segmenter un même réseau en plusieurs sous-réseaux de taille non nécessairement identiques en utilisant des masques de sous-réseaux (subnet netmask) différents.

Les intérêts de construire plusieurs sous-réseaux au sein d'un réseau sont divers :
- organisation et gestion plus efficace des adresses IP par segment ;
- utilisation de technologies différentes sur chaque sous-réseau (Ethernet, Token Ring, FDDI, ATM, etc.) ;
- services et applications dédiés par segment ;
- réduction de la charge globale du réseau avec amélioration du trafic dans chaque segment ;
- sécurisation de segments à l'intérieur du réseau ;
- nécessités liées à l'infrastructure (sites géographiques distants).

Le découpage de l'ensemble des nœuds d'un réseau en segments distincts est local, c'est-à-dire que l'identifiant du sous-réseau est obtenu en utilisant des bits de l'identifiant d'hôte de départ ; le découpage en sous-réseau est donc invisible de l'extérieur du réseau considéré.

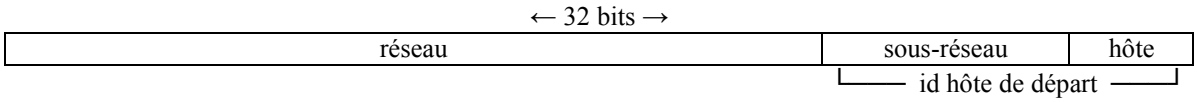

Figure 4.5 : position des identifiants de réseau, de sous-réseau et d'hôte dans une adresse IPv4

La construction logicielle d'un sous-réseau suit plusieurs étapes :
- détermination du nombre de segments nécessaires, et du nombre d'adresses IP pour chacun des segments ;
- détermination du masque nécessaire pour chacun des segments ;
- vérification du dimensionnement suffisant du réseau principal pour accueillir les différents segments ;
- construction des sous-réseaux dans le réseau principal (généralement ordonnés par ordre croissant ou décroissant de taille), calcul des adresses de sous-réseaux et des adresses de diffusion de chaque segment.

[1] Jamais utilisée pour désigner un hôte.
[2] Le localhost sert le plus souvent à tester la pile de protocoles TCP/IP afin de vérifier son bon fonctionnement ou bien à simuler un environnement réseau alors qu'on n'en dispose pas (application web, développement web, etc.).
[3] loopback (eng) ≡ boucle locale (fr).
[4] Spécifique aux machines fonctionnant sous le système d'exploitation Windows.

Ex. : Soit le réseau 192.168.1.0 / 24 pour lequel on doit construire 3 sous-réseaux :
- segment administratif (*Ad*) : 60 ordinateurs ;
- segment atelier (*At*) : 100 machines-outils ;
- segment services internet (*SI*) : 20 serveurs.

Analyse rapide :
- réseau 192.168.1.0 / 24, classe C → 256 adresses valides ;
- 180 adresses nécessaires pour les nœuds + 2 adresses supplémentaires pour chaque segment (adresse de sous-réseau et adresse de diffusion) → 186 adresses nécessaires → capacités suffisantes (en apparence).

Cependant, un sous-réseau a obligatoirement une taille puissance de 2 (séparation réseau/hôte) ; on vérifie :

Analyse approfondie :
- sous-réseau *Ad* de 60 ordinateurs → 62 adresses nécessaires → sous-réseau de 64 adresses (2^6) ;
- sous-réseau *At* de 100 machines-outils → 102 adresses nécessaires → sous-réseau de 128 adresses (2^7) ;
- sous-réseau *SI* de 20 serveurs → 22 adresses nécessaires → un sous-réseau de 32 adresses (2^5) ;
- 224 adresses réservées au total (< 256) → capacités suffisantes.

Construction des sous-réseaux (par ordre décroissant de taille) :

segment	plage	masque	adresse sous-réseau	broadcast	plage hôtes
At	192.168.1.0 192.168.1.127	255.255.255.128 (/25)	192.168.1.0 / 25	192.168.1.127	192.168.1.1 192.168.1.126
Ad	192.168.1.128 192.168.1.191	255.255.255.192 (/26)	192.168.1.128 / 26	192.168.1.191	192.168.1.129 192.168.1.190
SI	192.168.1.192 192.168.1.223	255.255.255.224 (/27)	192.168.1.192 / 27	192.168.1.223	192.168.1.193 192.168.1.222

Cette configuration de sous-réseaux est une proposition parmi plusieurs satisfaisantes ; autre exemple : SI (192.168.32 / 27), Ad (192.168.1.64 / 26), At (192.168.1.128 / 25).
Cependant, tout n'est pas faisable, ainsi le sous-réseau *At* ne peut être positionné qu'en 192.168.1.0 ou 192.168.1.128, et en aucun cas en 192.168.1.64 par exemple, car il serait alors impossible de définir 1 seul masque permettant d'isoler 128 adresses contiguës (192.168.1.64 / 25 renvoie à la plage 192.168.1.0 – 192.168.1.127).

Remarques :
- La segmentation en sous-réseaux possède un inconvénient : le « gâchis » d'adresses dans chaque sous-réseau, qui peut parfois être problématique.
  Ex. : Si le sous-réseau *Ad* comprenait 64 machines et non pas 60, il faudrait alors un sous-réseau de 128 adresses (2^7) ; il serait donc impossible de servir les 3 sous-réseaux distincts (128 + 128 + 24 > 256), même si le nombre d'adresses réellement utilisées est inférieur au potentiel du réseau principal (190 < 256) ;
- La segmentation d'un réseau principal en sous-réseaux impose l'attribution d'une adresse supplémentaire pour la passerelle d'interconnexion des sous-réseaux entre eux. Une passerelle possède donc autant d'interfaces réseaux (nœuds) qu'elle interconnecte de sous-réseaux et fait partie de chacun des sous-réseaux.
  Ex. : Si le sous-réseau *Ad* comprenait 62 machines et non pas 60, il faudrait un sous-réseau de 64 adresses (2^6) ; mais le sous-réseau serait alors complet et ne pourrait offrir d'adresse pour la passerelle, empêchant ce sous-réseau de communiquer avec les autres ;
- Le plus petit sous-réseau viable constructible dans un réseau quelconque est de 4 adresses (2^2) : l'adresse du sous-réseau, l'adresse de diffusion, deux adresses d'hôtes ; en pratique, l'une de ces adresses d'hôtes est donc réservée à la passerelle ;
- Selon les RFC, l'utilisation de sous-réseaux dont les bits de la partie sous-réseau sont tous à 0 ou tous à 1 est déconseillée afin d'éviter la confusion entre l'identifiant du réseau principal et l'identifiant de l'un de ses sous-réseaux (le premier), ainsi que la confusion entre l'adresse de broadcast du réseau principal et l'adresse de broadcast de l'un de ses sous-réseaux (le dernier).

Nb : Inversement, des réseaux peuvent être agrégés selon un système de sur-réseau à partir du moment où leurs adresses sont contiguës. Ceci permet de désigner un ensemble de réseaux différents en utilisant le même identifiant [1].

4.3.3 Le datagramme IPv4

Un **datagramme** IP, aussi appelé *paquet IP* lorsque fragmenté, correspond aux données émises de la couche supérieure (généralement issues du protocole TCP ou UDP) encapsulées dans une trame constituée de 2 champs :

[1] Propriété utilisée principalement dans les opérations de routage.

→ 1 datagramme IP (PDU) : 20 – 65535 octets ;
(taille minimale conseillée de 576 [1], et taille maximale de 65535 (64 ko) jamais atteinte [2])

- 1 champ en-tête (PCI) : 20 – 60 octets pour 13 informations ;
  - informations principales : 20 octets ;
  - informations optionnelles : 0 – 40 octets (assez peu souvent utilisées).
- 1 champ données (SDU) : 0 – 65515 octets.

Nb : Si le datagramme est d'une longueur trop importante pour transiter via la couche inférieure, il est fragmenté en plus petits paquets (aussi appelés fragments) afin de se conformer aux limitations de la technologie de réseau employé.

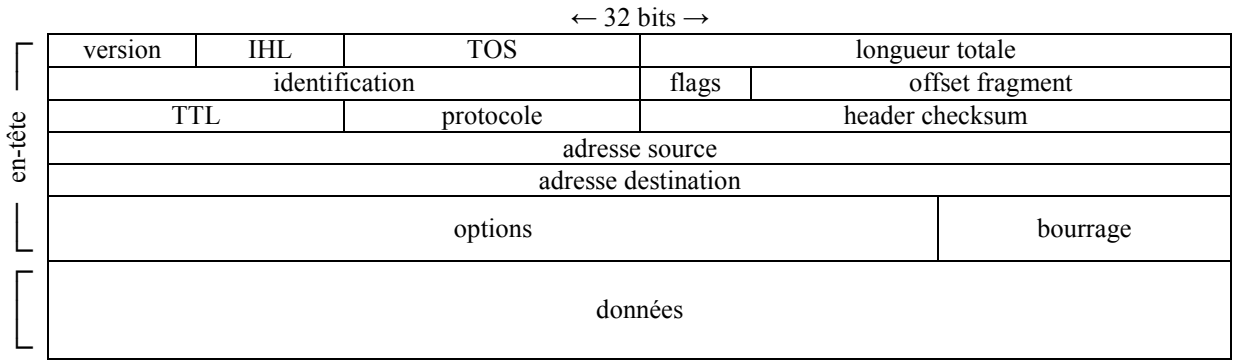

Figure 4.6 : datagramme IPv4

Les différentes informations principales du champ en-tête sont les suivantes :

- version (4) : numéro de version du protocole IP, 4 pour IPv4, 6 pour IPv6 ;
- IHL, Internet Header Length (4) : longueur de l'en-tête en mots de 32 bits (PCI), 5 – 15 ;
- TOS, Type Of Service (8) : type de service (utilisable par certains routeurs pour diriger éventuellement plus efficacement le datagramme afin de répondre à ses exigences, généralement ignoré cependant) ;
  - priorité (3) : de normale (défaut) à maximale, 0 – 7 ;
  - indicateurs de qualité (4) ;
    - D, Delay (1) : délai, 1 ;
    - T, Throughput (1) : débit, 1 ;
    - R, Reliability (1) : fiabilité, 1 ;
    - C, Cost (1) : coût de transmission, 1.
  - bit inutilisé (1) : 0.
- longueur totale (16) : longueur du datagramme en octets (SDU+PCI) ;
- identification (16) : numéro de fragment (utile si le datagramme a dû être fragmenté pendant le transit) ;
- flags (3) : indicateurs de fragmentation ;
  - bit inutilisé (1), 0 ;
  - DF, Don't Fragment (1) : ne doit pas être fragmenté, 1 ;
  - MF, More Fragments (1) : dernier fragment du datagramme, 0.
- offset fragment (13) : décalage du fragment dans le datagramme originel en mots de 64 bits, 0 pour le 1er, 0 si le datagramme n'a pas été fragmenté ;
- TTL, Time To Live (8) : durée de vie en secondes du datagramme lors du transit, décrémenté de 1 à chaque passage via un routeur, ou plus s'il stagne dans sa file d'attente, détruit si la durée de vie est atteinte (= 0), généralement initialisé à 64 ou 128 (le but est d'éviter qu'un datagramme circule indéfiniment sur le réseau) ;
- protocole (8) : nom du protocole encapsulé, 0 pour IP, 1 pour ICMP, 2 pour IGMP, 6 pour TCP, 17 pour UDP ;
- header checksum (16) : somme de contrôle de vérification de l'en-tête (pas de prise en compte du champ données), recalculée lors de chaque passage par un routeur (du fait de la modification du champ TTL) ;
- adresse source (32) : adresse IPv4 du nœud source sur 4 octets ;
- adresse destination (32) : adresse IPv4 du nœud destination sur 4 octets.

Les informations optionnelles du champ en-tête sont très peu utilisées (sécurité, gestion, route, datation, etc.) ; si une ou plusieurs options sont spécifiées, le champ en-tête est rempli de bits de bourrage (= 0), afin d'obtenir une longueur multiple de mots de 32 bits (4 octets).

[1] Préconisé par les RFC afin de s'associer à un grand nombre de technologies de réseaux (nda : 1280 octets pour IPv6).
[2] Adaptation du datagramme à la MTU de la technologie de réseaux employée afin d'éviter la fragmentation, coûteuse en ressources.

Le champ données peut être complété si nécessaire par des octets nuls pour atteindre une trame d'une longueur minimale attendue.

4.3.4 La fragmentation

La **fragmentation** consiste à découper un datagramme IP en fragments afin qu'il puisse être pris en charge par la couche hôte-réseau utilisée. En effet, le protocole IP peut être implanté sur diverses technologies de réseaux, et il doit donc proposer un mécanisme lui permettant de s'adapter aux limitations de chacune d'entre elles.

Sachant qu'un datagramme peut transiter via plusieurs réseaux différents, donc des technologies éventuellement différentes, il est impossible à priori de définir la taille maximale du fragment. Le datagramme est donc fragmenté, si besoin est, au niveau de chaque traversée de routeur conformément à la MTU du réseau allant être traversé ; la MTU [1] définit la taille maximale de la trame autorisée par la technologie de réseau employé [2]. La taille du fragment est bien évidemment choisie la plus grande possible, l'unité étant le mot de 64 bits (8 octets) [3].

Si la MTU d'un réseau est suffisamment grande pour accepter le datagramme ou le fragment, ce dernier sera encapsulé sans être fragmenté. Le réassemblage des fragments s'opère quoi qu'il arrive au niveau du nœud destination, et jamais au niveau des routeurs intermédiaires, même si les réseaux traversés autoriseraient des fragments plus grands.

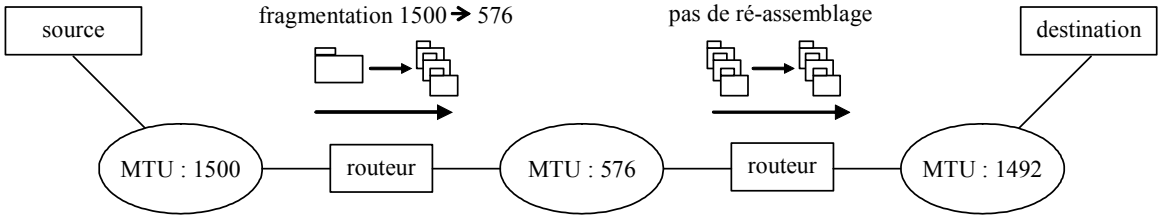

Figure 4.7 : fragmentation de datagrammes

À l'arrivée du premier fragment d'un datagramme, le nœud destinataire active un compte à rebours, utilisé conjointement avec le champ TTL de chaque fragment, qui détermine ainsi le délai laissé à tous les autres fragments pour arriver. Si ce délai arrive à expiration, les fragments reçus sont néanmoins détruits, et le datagramme n'est donc pas traité ; de plus un message ICMP [4] est alors envoyé à l'émetteur pour lui signifier l'erreur de transmission.

4.4 LES PROTOCOLES TCP/UDP

4.4.1 Généralités

4.4.1.1 Système client/serveur

La suite de protocoles TCP/IP est généralement mise en œuvre dans un système de communication où chaque machine tient un rôle précis. Pour établir la communication, l'une des machines doit débuter l'émission vers la seconde ; laquelle doit être en mesure de répondre à sa demande, et donc être en attente d'une réception.

Dans le cadre d'une communication ponctuelle, on distingue donc :

- la machine **serveur** : en attente d'une réception provenant de n'importe quelle machine, on dit *être à l'écoute* ;
- la machine **cliente** : réalise une émission en direction d'un serveur précis.

C'est toujours le client qui initie la communication en s'adressant à un serveur apte à répondre à sa demande, celui-ci proposant donc les services [5] recherchés en attendant passivement les requêtes des clients. Le serveur est alors un fournisseur de services, alors que le client est consommateur de services.

Un serveur est généralement capable de répondre à plusieurs clients en même temps, c'est-à-dire de traiter simultanément plusieurs communications avec différents clients, ainsi que d'être à l'écoute d'un nouveau client [6].

---

[1] Maximum Transfer Unit (eng) ≡ Unité de Transfert Maximale (fr).
[2] Exemples (en octets) : Ethernet : 1500, RTC : 576, PPPoE : 1492, ATM :9180, FDDI : 4470, Token Ring : ArpaNET : 1000, X25 : 128, SLIP : 256.
[3] Recherche de MTU optimale (cf. 4.5.2.3).
[4] cf. 4.5.2.2.
[5] Le mot « services » utilisé ici, ne correspond pas aux mêmes mécanismes mis en œuvre à l'interface inter-couches du modèle OSI (cf. 3.1.3).
[6] Les applications serveurs sont donc multi-threadées (thread : processus léger) afin de pouvoir répondre à toutes les sollicitations.

4.4.1.2 Port de connexion

Dans le protocole TCP/IP, la communication des données se réalise par messages découpés en paquets, arrivant dans un ordre incertain. Ces messages peuvent correspondre à plusieurs sessions différentes et donc être entremêlés ; ils peuvent aussi être issus d'applications diverses et utilisant donc des protocoles différents.

Le protocole TCP/IP doit donc disposer d'un mécanisme permettant de distinguer les messages appartenant à une session différente ainsi que les messages encapsulant des protocoles différents ; ce mécanisme est le principe de port de connexion.

Le **port de connexion**, aussi appelé *port de service*, est l'identifiant logiciel unique de l'application émettant ou recevant le message associé à une communication ponctuelle. Ce port est codé sur 16 bits.

Lors d'une communication entre deux hôtes, les ports de communication source et destination doivent donc être définis, et associés respectivement à l'adresse source et à l'adresse destination. L'ensemble (adresse + port), noté *adresse:port* et communément appelé **socket**, constitue un identifiant totalement unique, et le couple de sockets *adresse_src:port_src / adresse_dst:port_dst* les deux extrémités d'un support par lequel une communication point-à-point bidirectionnelle peut s'établir.

Il faut noter que dans un système client/serveur, pour contacter un service spécifique, seul le numéro de port peut réaliser la distinction entre les services proposés et les protocoles utilisés. Il en résulte que les ports source et destination d'une même communication sont généralement différents [1].

Parmi les 2^{16} numéros de port de connexion disponibles, 3 zones sont définies [2] :

- 0 – 1023 : ports reconnus [3], réservés aux services standard ;
  Exemples :
  - 20-21 : FTP
  - 23 : Telnet
  - 25 : SMTP
  - 53 : DNS
  - 80 : HTTP
  - 110 : POP3
  - 443 : HTTPS
  - 137-139 : NetBios
  - 445 : partage de fichiers et d'imprimante Windows / SaMBa
- 1024 – 49151 : ports enregistrés [4], utilisables temporairement par le système d'exploitation ou dans le domaine privé ;
- 49152 – 65535 : ports publics et dynamiques, libres d'utilisation.

Ainsi, un ordinateur émettant une requête HTTP vers un serveur web établit donc une connexion entre un port local déterminé aléatoirement par le système d'exploitation et le port 80 du serveur web, soit donc par exemple le couple de sockets 212.195.198.187:2256 / 78.109.84.60:80.

4.4.2 Le protocole TCP

4.4.2.1 Définitions

Le **protocole TCP** (Transmission Control Protocol [5]) assure les services attendus de la couche transport du modèle TCP/IP. Son rôle est donc de gérer le fractionnement et le réassemblage en paquets des segments de données [6] qui transitent via le protocole IP. Afin de fiabiliser la communication, TCP doit donc aussi réordonner les paquets avant de les assembler, et doit aussi gérer les paquets erronés ou perdus.

Pour cela, TCP fonctionne en mode *connecté* en usant de deux mécanismes mettant en œuvre un principe de synchronisation/question/réponse/confirmation :

[1] Il est en effet parfaitement imaginable qu'une machine soit à la fois cliente et serveur d'un même type de services.
[2] cf. annexe B.2.
[3] Ports reconnus (fr) ≡ well-known ports (eng).
[4] Ports enregistrés (fr) ≡ registered ports (eng).
[5] Transmission Control Protocol (eng) ≡ Protocole de Contrôle de Transmission (fr).
[6] On parle de protocole « bout en bout » (end to end) car les paquets TCP sont vus comme des parties d'une entité (un message, un fichier, etc.) et non comme des données « anonymes » et indépendantes.

- accusé de réception (/ acquittement : ACK [1]) : tout envoi de données de la machine A vers la machine B est acquitté par B en renvoyant un acquittement à A ; cet acquittement est transporté soit par un paquet dédié, soit par un paquet transportant aussi des données à transmettre de B vers A [2] ;
  - l'acquittement doit être reçu avant l'échéance d'une temporisation amorcée par A lors de l'envoi des données ;
  - un paquet-acquittement peut transporter un acquittement cumulatif pour plusieurs envois de données distincts ;
  - l'émetteur conserve une trace des paquets émis ;
  - le receveur garde une trace des paquets reçus.

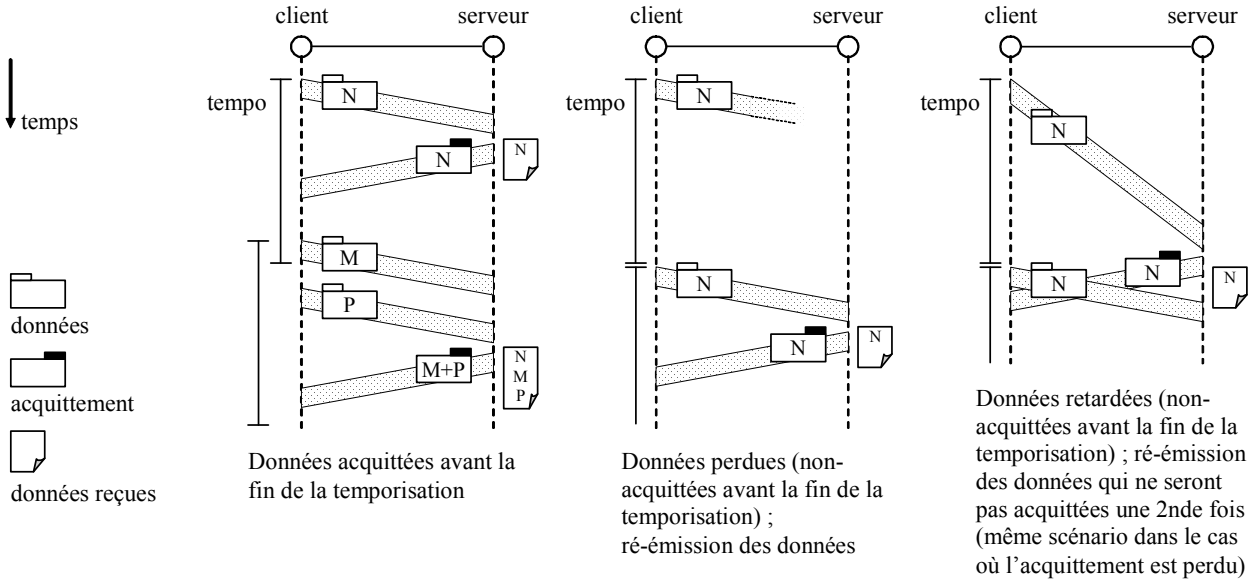

Figure 4.8 : principe et usage de l'acquittement

- mécanisme dit « poignée de main » : les deux parties se contactent et s'accordent afin de s'assurer d'être prêtes à communiquer avant de débuter la transmission ; un mécanisme similaire est mis en place en fin de transmission.

TCP, malgré son ancienneté (1981), demeure un protocole robuste et fiable [3], utilisé dans de nombreuses applications [4] car bien adapté aux réseaux maillés, et de ce fait utilisé dans plus de 90% du volume de données transitant via internet ; son principal inconvénient est le manque de sécurisation qu'il propose.

4.4.2.2 Le segment TCP

Un **segment TCP**, aussi appelé *paquet TCP*, correspond aux données émises de la couche supérieure encapsulées dans une trame constituée de 2 champs :

→ 1 segment TCP (PDU) : 20 – 65515 octets (doit pouvoir être encapsulé dans un datagramme IP) ;

- 1 champ en-tête (PCI) : 20 – 60 octets ;
  - informations principales : 20 octets ;
  - informations optionnelles : 0 – 40 octets.
- 1 champ données (SDU) : 0 – 65495 octets.

Le segment TCP est encapsulé dans un datagramme IP en fixant le champ protocole à 6.

[1] Acquittement (fr) ≡ ACKnowledgment (eng).
[2] Cette technique d'association de données et d'un acquittement au sein d'un même segment TCP est appelée *piggybacking*.
[3] Mais lent… au contrario du protocole UDP (cf. 4.4.3).
[4] HTTP, FTP, SMTP, POP3, …

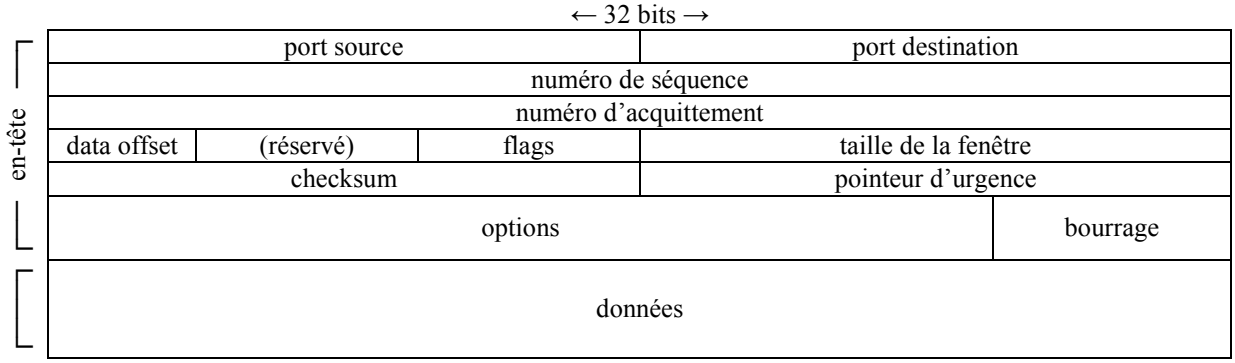

Figure 4.9 : segment TCP

Les différentes informations principales du champ en-tête sont les suivantes :

- port source (16) : port source de l'application sur la machine source ;
- port destination (16) : port destination de l'application sur la machine destination ;
- numéro de séquence (32) : numéro du premier octet du paquet dans l'ensemble du flux de données transmis, 0 pour le 1er paquet d'un message – cf. bit SYN ;
- numéro d'acquittement (32) : numéro de séquence attendu, soit donc le numéro du dernier octet reçu incrémenté de 1 (les paquets de numéros inférieurs ayant donc tous été reçus) – cf. bit ACK ;
- data offset (4) : position du champ données dans le paquet en mots de 32 bits ≡ longueur de l'en-tête (PCI) ;
- (réservé) (6) : positionné à 0 ;
- flags (6) : bits de contrôle ;
  - URG, URGent (1) : utilisation du champ *pointeur d'urgence*, 1 ;
  - ACK, ACKnowledgment (1) : validation du champ *numéro d'acquittement*, 1 ;
  - PSH, PuSH (1) : livraison instantanée des données à l'application sans mise en mémoire tampon ≡ demande d'acquittement, 1 ;
  - RST, ReSeT (1) : demande de réinitialisation de connexion, 1 ;
  - SYN, SYNchronisation (1) : synchronisation des numéros de séquence, 1 ;
  - FIN, FINalize (1) : fin de la transmission, 1.
- taille de la fenêtre (16) : nombre d'octets à transmettre sans nécessité d'accusé de réception ;
- checksum (16) : somme de contrôle de vérification (prise en compte du champ données, et d'un champ en-tête virtuel constitué des adresses IP source et destination extraites de l'en-tête IP) ;
- pointeur d'urgence (16) : position d'une donnée urgente par rapport au numéro de séquence, spécifiant une livraison instantanée à l'application dès que l'octet pointé est lu (à positionner juste après la donnée urgente) – cf. bit URG.

Les informations optionnelles du champ en-tête sont peu utilisées (indication de la MTU pour optimisation, etc.) ; si une ou plusieurs options sont spécifiées, elle sont codées en multiple de 8 bits, et le champ en-tête est rempli de bits de bourrage (= 0), afin d'obtenir une longueur multiple de mots de 32 bits.

4.4.2.3 Gestion d'une connexion TCP

Le protocole TCP est orienté connexion suivant un mécanisme de « poignée de main » en trois temps [1] qui vise à s'assurer que la source et la destination sont prêtes à communiquer en se synchronisant. Une fois la communication établie, les 2 parties sont *connectées* jusqu'à sa fermeture explicite.

Les différentes étapes de ce mécanisme pour l'établissement de la communication sont les suivantes :

- 0 : Le serveur attend passivement l'arrivée d'une demande de connexion (mise en écoute sur un port donné avec la primitive *listen* + mise en attente d'une connexion avec la primitive *accept*) ;
- 1 : Le client envoie une demande de connexion (*adresse_dst:port_dst*) sous forme d'un segment TCP avec les bits SYN à 1 et ACK à 0, ainsi que son numéro de séquence initial (N) (« ouverture active » avec la primitive *connect*), puis attend une réponse ;
- 2 : Le serveur répond en envoyant un segment avec les bits SYN et ACK à 1, qui acquitte le segment reçu du client avec le numéro d'acquittement (N+1) et indique son numéro de séquence initial (P) (« ouverture passive ») ;
- 3 : Le client acquitte ce segment reçu en renvoyant un segment avec le bit ACK à 1 et le numéro d'acquittement (P+1) ; la connexion est établie, les données peuvent être transmises.

[1] Poignée de main en trois temps (fr) ≡ triple ways handshake (eng).

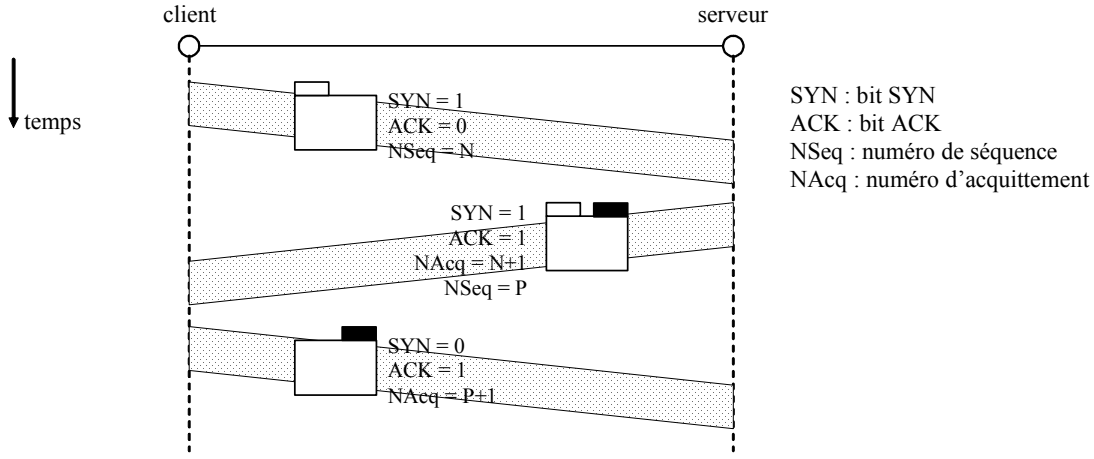

Figure 4.10 : établissement d'une connexion TCP

La synchronisation – la « poignée de main » – consiste donc à ce que chacune des deux parties fasse connaître à l'autre son numéro de séquence initial dans le cadre de cette communication ponctuelle, et que ce numéro de séquence soit acquitté.

Un principe similaire est mis en jeu à la fermeture de la connexion, sachant que la finalisation de la connexion peut être initiée aussi bien par le serveur que par le client :

- 0' : La partie désireuse de finaliser la connexion envoie un segment avec les bits FIN à 1 et ACK à 0, ainsi qu'un numéro de séquence (X) (fermeture avec la primitive *close*) ;
- 1' : L'autre partie acquitte ce segment reçu en renvoyant un segment avec le bit ACK à 1 et un numéro d'acquittement (X+1) ; la connexion est maintenant fermée dans le sens demandeur_finalisation → acquitteur_finalisation, il ne reste qu'une « demi »-connexion ; seuls des acquittements peuvent être envoyés par le demandeur_finalisation ;
- 2'+3' : Même chose pour l'autre partie lorsqu'elle désire fermer sa « demi »-connexion.

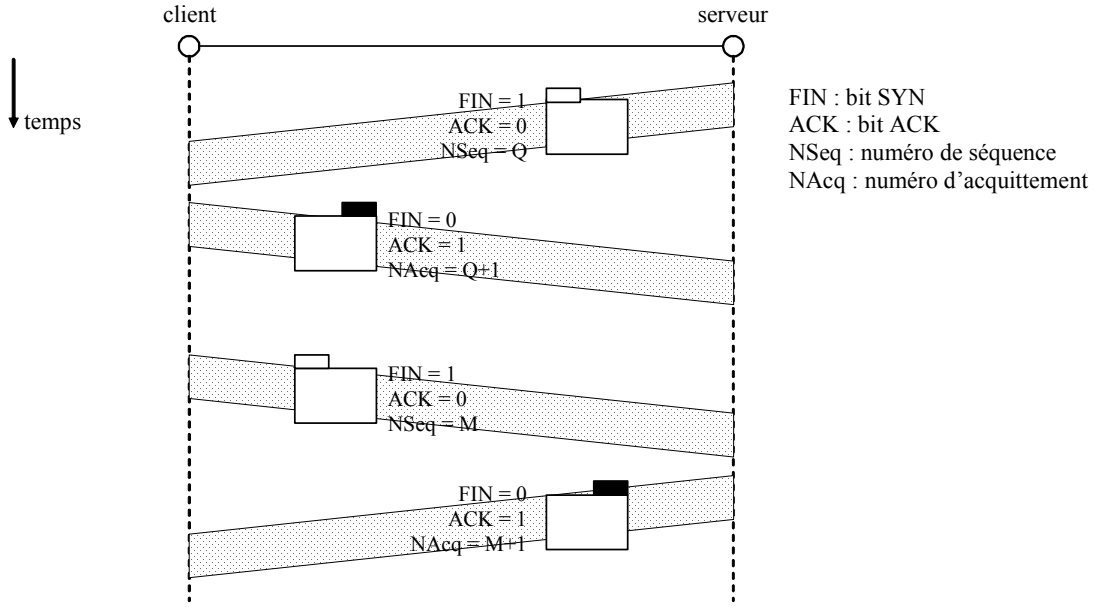

Figure 4.11 : fermeture d'une connexion TCP

Le numéro de séquence initial est déterminé selon plusieurs critères (temps en seconde depuis le démarrage du système d'exploitation, nombre de sockets ouverts depuis le démarrage, etc.), qui, associés au grand intervalle de valeurs possibles [1], assurent ainsi la quasi-unicité du numéro de séquence à un instant donné ; ceci permet de distinguer facilement plusieurs connexions TCP ouvertes en parallèle.

[1] $2^{32} - 1 \approx 4.3 \times 10^9$.

Nb : Une amélioration de TCP, nommée *acquittement sélectif* (SACK, Selective ACKnowledgment), permet d'acquitter cumulativement des paquets reçus dans le désordre.

4.4.2.4 Contrôle de flux

Comme les deux machines qui communiquent sont hétérogènes – avec donc des buffers d'émission et de réception de tailles différentes – il est nécessaire de disposer d'un mécanisme de synchronisation afin d'éviter les pertes de données, ainsi que d'optimiser le trafic.

Le **contrôle de flux** est mis en œuvre avec la méthode de la fenêtre glissante, basée sur la taille de la fenêtre du récepteur – son buffer de réception –, et qui est utilisée conjointement avec le principe d'acquittement de tout envoi de données et à la possibilité d'acquitter plusieurs envois de données en une seule fois.

Le récepteur communique la taille de la fenêtre à l'émetteur lors de la poignée de main d'établissement de la connexion ainsi que lors de chaque envoi d'acquittement, car la taille de cette fenêtre est susceptible de varier (« glissante »). L'émetteur se synchronise alors en temps réel sur cette valeur de fenêtre pour réaliser ses envois de données sans attendre de recevoir d'acquittement, et ce jusqu'à atteindre la taille de la fenêtre précisée ou qu'un acquittement lui parvienne. De son côté, le récepteur reçoit les données et envoie un acquittement cumulatif lorsque son buffer de réception arrive à saturation ou que la taille de la fenêtre évolue.

Nb : Le flux peut aussi être modifié tacitement par interprétation implicite de l'état du réseau entre l'émetteur et le destinataire. Ainsi, l'absence ou le duplicata d'acquittements implique une modification du flux par temporisation, réémission de certains paquets, etc.

4.4.3 Le protocole UDP

4.4.3.1 Définitions

Le **protocole UDP** (User Datagram Protocol [1]) assure les services attendus de la couche transport du modèle TCP/IP. Tout comme TCP, son rôle est donc de gérer le fractionnement et le réassemblage en paquets des segments de données qui transitent via IP. Cependant, UDP n'assure aucun autre service supplémentaire : pas de réordonnancement, pas de suivi de la communication à l'aide d'accusé de réception, pas de contrôle de flux.

UDP fonctionne en mode *non connecté*, c'est-à-dire qu'il ne fait que transporter les paquets de manière indépendante, sans assurer la moindre cohérence entre eux.

Cette implémentation simpliste de la couche transport assure une transmission résolument non fiable [2] mais extrêmement rapide, dédiant UDP à un usage spécifique aux transmissions de données de très faible volume [3], devant être transmises rapidement [4], ou étant peu sensibles à un faible taux d'erreur [5]. De plus, cette transmission sans gestion de connexion permet à UDP d'assurer des transmissions de type multicast, en diffusion vers plusieurs nœuds destinataire en même temps ; ce dont n'est pas capable TCP [6].

Aucune garantie n'est donc assurée par UDP, et c'est alors à l'application qui communique via UDP de gérer éventuellement les problèmes de pertes, duplications, retards, etc.

4.4.3.2 Le datagramme UDP

Un **datagramme UDP**, aussi appelé *paquet UDP*, correspond aux données émises de la couche supérieure encapsulées dans une trame constituée de 2 champs :

→ 1 datagramme UDP (PDU) : 8 – 65515 octets (doit pouvoir être encapsulé dans un datagramme IP) ;

- 1 champ en-tête (PCI) : 8 octets ;
- 1 champ données (SDU) : 0 – 65507 octets.

Le datagramme UDP est encapsulé dans un datagramme IP en fixant le champ protocole à 17.

[1] User Datagram Protocol (eng) ≡ Protocole de Datagramme Utilisateur (fr).
[2] Au contrario du protocole TCP (cf. 4.4.2).
[3] DHCP, DNS, NTP, …
[4] Streaming, …
[5] Données multimédia, …
[6] Puisqu'il lui faudrait gérer la connexion (synchronisation, acquittement, etc.) avec N nœuds destinataires, soit donc gérer effectivement N communications en même temps avec des nœuds pleinement connus – via leur adresse IP –, c'est-à-dire transmettre N fois les mêmes données.

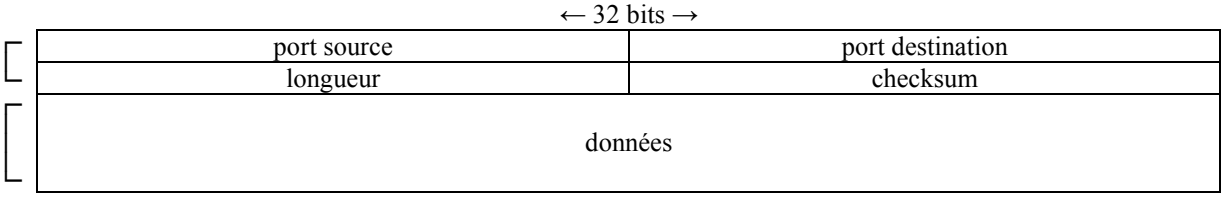

Figure 4.12 : datagramme UDP

Les différentes informations du champ en-tête sont les suivantes :
- port source (16) : port source de l'application sur la machine source ;
- port destination (16) : port destination de l'application sur la machine destination ;
- longueur (16) : longueur de l'en-tête et des données, 8 – 65515 ;
- checksum (16) : somme de contrôle de vérification de l'en-tête (prise en compte du champ données, et d'un champ en-tête virtuel constitué des adresses IP source et destination extraites de l'en-tête IP).

Le champ données peut être complété si nécessaire par un octet nul pour atteindre un nombre total d'octets pair.

4.5 COMPLÉMENTS

4.5.1 Routage IP

4.5.1.1 Définitions

Le **routage IP**, principale fonctionnalité assurée par le protocole IP, désigne le processus de détermination du chemin par lesquels les datagrammes IP transitent de la source à la destination.

La *route* est représentée par la liste ordonnée des différentes machines intermédiaires et successives par lesquelles la communication s'effectue, machines alors appelées *routeurs*. Ce choix de route s'appuie généralement sur le trafic et l'état de congestion du réseau au moment de l'envoi du message afin d'acheminer les paquets le plus rapidement possible, mais il peut aussi se baser sur des critères comme la fiabilité ou le coût de transmission [1].

Comme IP fonctionne selon un système de commutation par paquets, l'émetteur ne détermine donc pas l'intégralité de la route optimale et des routes optionnelles avant l'envoi, mais fonctionne suivant un système de routage par sauts successifs [2], en déterminant uniquement le premier routeur dans la direction de la destination. Lequel réalise la même opération, et ainsi de suite jusqu'à atteindre le réseau du destinataire. Ceci a aussi l'avantage d'éviter de maintenir une liste de routes exhaustive et complexe [3], et de laisser le choix du segment de route à l'appréciation de chaque routeur.

Chaque nœud du réseau, routeur ou non, ne connaît ainsi que quelques segments de route, constituant ce qu'on appelle la *table de routage*, et qui contient les informations permettant d'atteindre les différentes destinations possibles [4]. Un segment de route est ainsi défini par 4 informations principales :
- destination : adresse IP d'un réseau (éventuellement d'un hôte) ;
- masque : masque associé à l'adresse IP de destination ;
- passerelle : adresse IP du routeur devant prendre en charge les communications pour cette destination (aucune si la destination est sur un réseau accessible directement) ;
- interface : interface physique par laquelle la destination est accessible (directement ou indirectement via une passerelle).

Toute table de routage contient toujours une route par défaut, à suivre si le destinataire du datagramme ne correspond à aucune destination des segments de route spécifiés dans la table de routage.

En pratique, pour établir la table de routage, on procède donc à l'inverse : on détermine la route par défaut (dans le cas d'une machine pouvant accéder à internet, il s'agit généralement de la route qui y mène), puis on s'occupe des réseaux qui ne peuvent être atteints via cette passerelle.

Ex. :

[1] Ceux-ci étant disponibles dans le protocole IP sous forme d'indicateurs de qualité (cf. 4.3.3).
[2] Routage par sauts successifs (fr) ≡ next-hop routing (eng).
[3] Les gros routeurs peuvent avoir à gérer jusqu'à plusieurs milliers de routes distinctes.
[4] La commande « netstat –r » permet d'afficher la table de routage courante ; la commande « route » permet de gérer la table de routage.

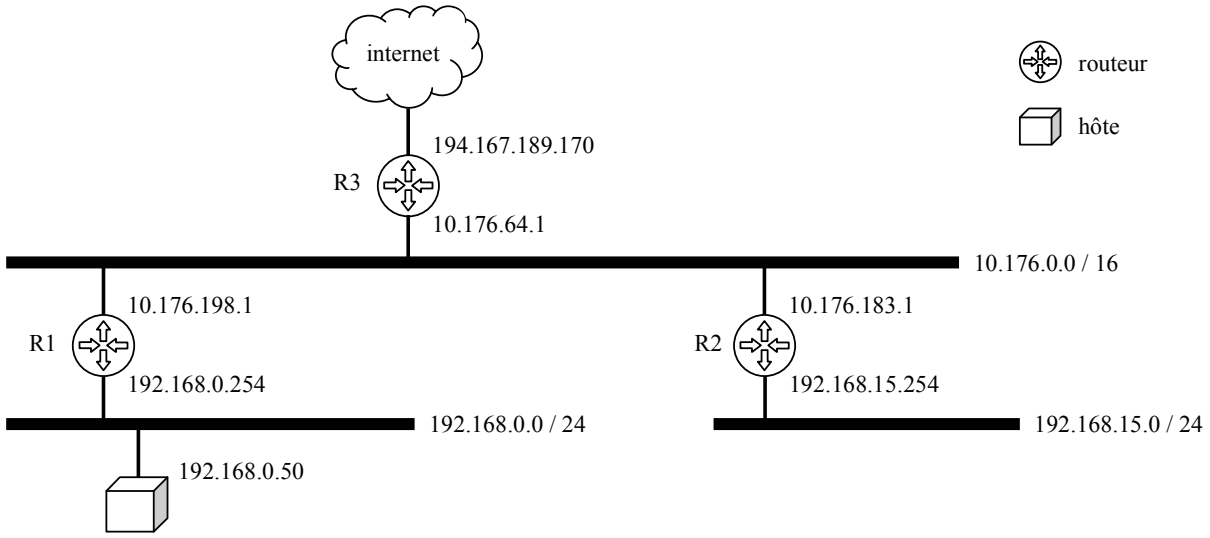

Figure 4.13 : exemple d'interconnexion de réseaux via des routeurs

La table de routage associée à l'hôte 192.168.0.50 est la suivante :

destination	*masque*	*passerelle*	*interface*
192.168.0.0	255.255.255.0 (/24)	0.0.0.0 / * / 192.168.0.50	192.168.0.50 / eth0
127.0.0.0	255.0.0.0 (/8)	0.0.0.0 / * / 127.0.0.1	127.0.0.1 / lo
0.0.0.0 / default	0.0.0.0 (/0)	192.168.0.254	192.168.0.50 / eth0

La table de routage associée au routeur R1 est la suivante :

destination	*masque*	*passerelle*	*interface*
192.168.0.0	255.255.255.0 (/24)	0.0.0.0 / * / 192.168.0.254	192.168.0.254 / eth1
192.168.15.0	255.255.255.0 (/24)	10.176.183.1	10.176.198.1 / eth0
10.176.0.0	255.255.0.0 (/16)	0.0.0.0 / * / 10.176.198.1	10.176.198.1 / eth0
127.0.0.0	255.0.0.0 (/8)	0.0.0.0 / * / 127.0.0.1	127.0.0.1 / lo
0.0.0.0 / default	0.0.0.0 (/0)	10.176.64.1	10.176.198.1 / eth0

Nb : L'indication 0.0.0.0 pour destination et masque correspond à n'importe quelle adresse IP. L'indication 0.0.0.0 pour la passerelle désigne un réseau accessible directement (la passerelle est alors l'hôte courant lui-même) ; on peut indiquer à la place *, ou bien l'adresse IP de l'interface sur laquelle le réseau est directement joignable. L'adresse IP de l'interface peut être remplacée par le nom de l'interface au sein du système d'exploitation : ethx pour les interfaces type ethernet, lo pour l'interface loopback, pppx pour les interfaces série, etc.

L'ordre des segments de route indiqués dans la table de routage est important. En effet, lors d'un choix de route, l'adresse de destination du datagramme à router est recherchée dans la table de haut en bas ; si l'adresse correspond à l'une des entrées (*destination* & *masque*), alors le datagramme est routé vers le routeur indiqué (*passerelle*) via le nœud associé (*interface*). C'est pourquoi la route « par défaut » est toujours indiquée en dernier.

4.5.1.2 Routage dynamique

Afin d'assurer la bonne transmission des datagrammes, tout comme pour optimiser le trafic, il est nécessaire de maintenir régulièrement à jour les tables de routage de l'ensemble des routeurs d'un réseau. Sur les gros réseaux, cette tâche devient rapidement irréalisable. Il faut alors mettre en place une solution de routage dynamique.

Le **routage dynamique** est un mécanisme par lequel les routeurs communiquent entre eux et peuvent alors se configurer les uns les autres de manière totalement automatique. Ainsi, on s'assure que les tables de routage sont à jour en reflétant l'état réel et physique des routes. Les avantages sont multiples : optimisation du trafic, baisse des erreurs de transmission, gestion des surcharges, etc.

Le routage dynamique est implémenté par l'utilisation d'un protocole de routage. Divers protocoles existent que l'on peut regrouper en 2 catégories :

- IGP (Interior Gateway Protocol [1]) : Protocole de routage dynamique entre routeurs d'un même système autonome, c'est-à-dire dont la gestion dépend d'une administration unique (routeur vers internet d'un petit

[1] Interior Gateway Protocol (eng) ≡ Protocole de Passerelle Interne (fr).

réseau local, ensemble des réseaux d'une multinationale, etc.) ; ce type de protocole privilégie la fiabilité et les plus courts chemins pour la rapidité de transmission ;
  - RIP (Routing Information Protocol [1]) : Routage basé sur le nombre de routeurs intermédiaires entre deux réseaux ;
  - OSPF (Open Shortest Path First) : Routage basé sur l'état des liens entre deux réseaux.
- EGP (Exterior Gateway Protocol [2]) : Protocole de routage dynamique entre routeurs de différents systèmes autonomes ; les règles de routage sont établies en fonction d'accords commerciaux, de considérations politiques ou de sécurité, etc.
  - BGP (Border Gateway Protocol [3]) : Protocole utilisé pour l'internet afin de mettre à jour les informations de routage des quelques 20000 systèmes autonomes déployés dans le monde.

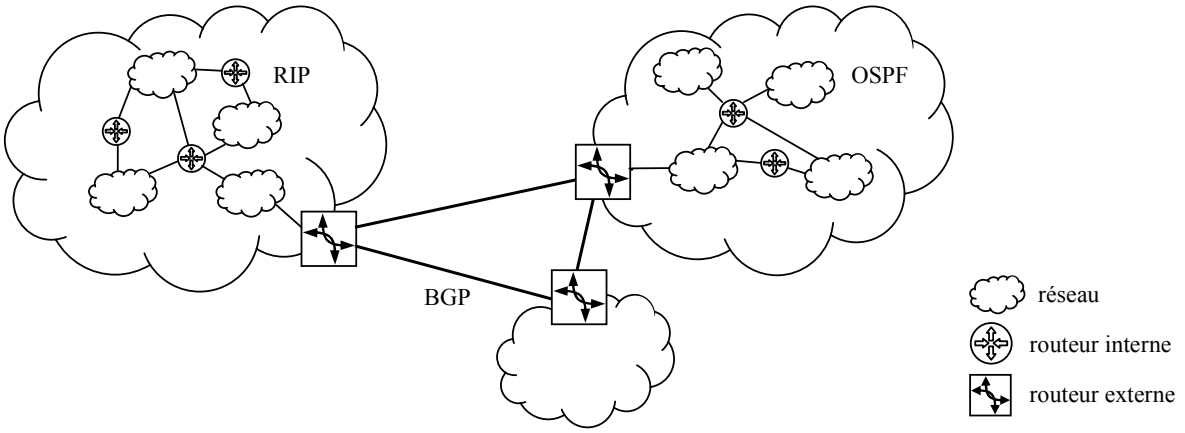

Figure 4.14 : internet : interconnexion de systèmes autonomes

Bien que les annonces de ces divers protocoles soient généralement encapsulées dans TCP (ex. : BGP) ou UDP (ex : RIP), les protocoles de routage font partie de la couche réseau (3 OSI, 2 TCP/IP).

4.5.1.3 Le protocole RIP

Le **protocole RIP** (Routing Information Protocol) est un protocole de routage dynamique type IGP destiné à être utilisé sur des réseaux peu étendus. Il fonctionne sur la base du nombre de routeurs intermédiaires sur la route pour joindre une destination donnée ; cette information est appelée *métrique*, et est codé sur 4 bits : 1 pour la route la plus courte (seul le routeur courant doit être traversé), 15 pour la route la plus longue, 16 étant réservé pour signaler une route infinie. Ce protocole fonctionne de manière à détecter la route optimale pour joindre une destination, soit donc celle proposant la métrique la plus petite.

Toutes les 30 secondes, chaque routeur diffuse à ses routeurs voisins une annonce relative aux règles de sa table de routage courante qui spécifie une ou plusieurs entrées *adresse destination* / *métrique* / *adresse passerelle* [4].

Les entrées reçues sont alors analysées par chaque routeur pour mettre à jour leur table locale :
- ajout de 1 à la métrique de chaque règle de routage ;
- si l'adresse de destination n'existait pas, la règle est ajoutée à la table de routage ;
- si l'adresse de destination existait, seule l'entrée ayant la meilleure métrique (la plus petite) est conservée ;
- si la métrique atteint la valeur infinie (16), l'entrée est ignorée ;
- les entrées ont une durée de vie de 180 secondes qui est réinitialisée en cas de modification de la règle.

Une fois les règles de routage ajoutées, modifiées ou enlevées, la table mise à jour est propagée par le routeur à l'émission de sa prochaine propre annonce RIP.

Une annonce RIP est constituée de 2 champs :
→ 1 annonce RIP (PDU) : 24 – 504 octets ;
- 1 champ en-tête (PCI) : 4 octets ;
- 1 champ données (SDU) : 20 – 500 octets.

L'annonce RIP est encapsulée dans un datagramme UDP.

[1] Routing Information Protocol (eng) ≡ Protocole d'Informations de Routage (fr).
[2] Exterior Gateway Protocol (eng) ≡ Protocole de Passerelle Externe (fr).
[3] Border Gateway Protocol (eng) ≡ Protocole de Passerelle de Bordure (fr).
[4] 25 routes maximales par message RIPv2.

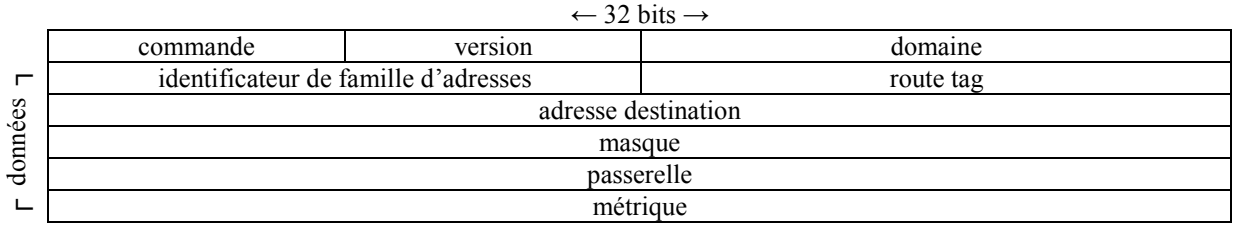

Figure 4.15 : annonce RIP

Les différentes informations du champ en-tête sont les suivantes :
- commande (8) : type de commande, 1 pour une requête de tout ou partie d'une table de routage, 2 pour une réponse ou pour une annonce diffusée ;
- version (8) : version du protocole RIP, 1 pour RIPv1, 2 pour RIPv2 ;
- domaine (16) : découpe du réseau en sous-réseaux logiques pour prise en compte du découpage en sous-réseaux (0 pour RIPv1 car inutilisé).

Les différentes informations du champ données sont les suivantes (pour 1 entrée de la table de routage) :
- identificateur de famille d'adresses (16) : 0 pour une requête de toute la table de routage, 2 pour une adresse IPv4 de destination, 65535 (0xFFFF) pour une authentification ;
- route tag (16) : marqueur afin de distinguer une route issue du protocole RIP ou d'un autre protocole de routage (EGP ou IGP) (0 pour RIPv1 car inutilisé) ;
- adresse destination (32) : adresse IPv4 de destination du réseau visé ;
- masque (32) : masque associé à l'adresse destination (0 pour RIPv1 car inutilisé) ;
- passerelle (32) : adresse IPv4 de la passerelle (0 pour RIPv1 car inutilisé) ;
- métrique (32) : nombre de routeurs intermédiaires pour atteindre la destination.

L'annonce RIP peut contenir des règles pour 25 routes au maximum. Éventuellement, un mécanisme d'authentification peut être introduit, qui occupe l'espace complet de la première entrée ; auquel cas, seules 24 routes peuvent être communiquées par l'annonce.

RIP est un ancien protocole de routage dynamique qui a comme faiblesse de ne prendre en compte que le nombre de routeurs et aucunement les débits maximaux assurés par les réseaux traversés en suivant ces routeurs. De plus, il est peu performant en cas de défaillance d'un réseau, et propose un trafic réseau supplémentaire non-négligeable.

De fait, RIP tend à être remplacé par le protocole OSPF qui se base sur la propagation de l'état des liens permettant à chaque routeur d'avoir un aperçu fidèle en temps réel de la topologie complète du réseau.

Néanmoins, RIP est encore utilisé sur les petits et moyens réseaux car l'algorithme de gestion est très simple par rapport à des solutions comme OSPF qui sont coûteuses en ressources même si plus efficaces.

4.5.2 Le protocole ICMP

4.5.2.1 Définitions

Le **protocole ICMP** (Internet Control Message Protocol [1]) est un protocole de la couche réseau offrant un ensemble d'outils et de signaux nécessaires au routage pour la gestion de l'acheminement des paquets. Bien qu'appartenant à la couche internet du modèle TCP/IP, ICMP est encapsulé dans IP, et permet ainsi non pas de fiabiliser une transmission mais de déterminer les causes éventuelles d'un problème en proposant un compte-rendu d'erreur.

Les messages ICMP, traitant principalement des erreurs issues du protocole IP, sont donc envoyés par les routeurs situés entre les nœuds émetteur et destinataire. Cependant, comme ICMP peut aussi se référer à TCP ou UDP, voire même directement aux applications, des messages ICMP peuvent aussi être envoyés par le destinataire.

Il faut noter que ICMP étant acheminé via IP, aucune garantie n'est donc assurée pour sa bonne transmission. Cependant, afin d'éviter l'envoi d'erreurs en avalanche, les problèmes de transmission des messages ICMP eux-mêmes ne sont jamais traités.

4.5.2.2 Le message ICMP

Un **message ICMP** correspond aux informations du message transmis :
→ 1 message ICMP (PDU) : 4 – 65515 octets (limite jamais atteinte) ;
- 1 champ en-tête (PCI) : 4 octets ;
- 1 champ données (SDU) : 0 – 65511 octets.

[1] Internet Control Message Protocol (eng) ≡ Protocole de Message de Contrôle de l'Internet (fr).

Le message ICMP est encapsulé dans un datagramme IP en fixant le champ protocole à 1.

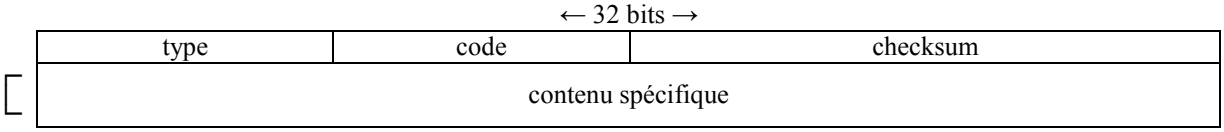

Figure 4.16 : message ICMP

Les différentes informations du champ en-tête sont les suivantes :
- type (8) : type de message transmis ;
- code (8) : code précisant le type de message ;
- checksum (16) : somme de contrôle de vérification du message.

Voici une liste non-exhaustive des 18 types de messages ICMP (type / code) :
- 8 / 0 : demande d'écho (*echo request*) ;
- 0 / 0 : réponse d'écho (*echo reply*) ;
- 3 / : compte-rendu d'erreur envoyé par un routeur à l'expéditeur lorsque celui-ci ne peut délivrer un datagramme IP :
  - / 0 : réseau inaccessible ;
  - / 1 : machine inaccessible ;
  - / 2 : protocole inaccessible ;
  - / 3 : port inaccessible ;
  - / 4 : fragmentation nécessaire mais bit DF [1] positionné à 1 (*destination unreachable / fragmentation required*) ;
  - / 6 : réseau destination inconnu ;
  - / 7 : destinataire inconnu.
- 4 / 0 : destinataire surchargé ;
- 5 / : demande de modification de route envoyée par un routeur à l'expéditeur lorsque celui-ci détecte que l'expéditeur utilise une route non optimale :
  - / 0 : redirection pour un réseau ;
  - / 1 : redirection pour une machine.
- 9 / 0 : annonce d'une route ;
- 10 / 0 : demande de route ;
- 11 / : détection d'un TTL [2] à 0 (*TTL exceeded*) :
  - / 0 : pendant le transit du datagramme IP ;
  - / 1 : lors du réassemblage du datagramme.
- 12 / 0 : en-tête IP erroné.

Le contenu du message contient toujours l'en-tête IP et les 8 premiers octets du datagramme qui est source de l'erreur, permettant ainsi d'identifier le protocole et l'adresse IP en cause.

4.5.2.3 Utilisation des messages ICMP

La commande `ping` permet de déterminer l'existence, la visibilité, l'accessibilité et le temps de réponse moyen d'une machine sur le réseau via son adresse IP.

Elle fait usage d'un couple de messages ICMP : *echo request* (8/0) qui réalise la demande d'écho, et *echo reply* (0/0) qui est la réponse à cette demande.

La commande `traceroute` [3] permet de suivre la route utilisée pour communiquer avec une machine précise sur le réseau, c'est-à-dire déterminer les adresses IP des routeurs par lesquels transite la communication.

Elle fait usage d'un même message quelconque [4] envoyé successivement plusieurs fois, avec une valeur du champ TTL initialisée à 1, et incrémentée de 1 à chaque fois. Ainsi, lorsque le premier message ICMP atteint le premier routeur sur la route, le champ TTL est décrémenté ; celui-ci devient donc 0, signifiant que le message ne doit pas être transmis, et le routeur renvoie alors le message ICMP *TTL exceeded* (11/0) à l'expéditeur. Le second message ICMP sera refusé par le second routeur sur la route et en informera l'expéditeur ; et ainsi de suite, jusqu'à joindre enfin le destinataire où aucun message *TTL exceeded* ne sera envoyé.

[1] Don't Fragment (eng) ≡ Ne pas Fragmenter (fr) (cf. 4.3.3).
[2] Time To Live (eng) ≡ Temps À Vivre (fr) (cf. 4.3.3).
[3] Commande appelée par « tracert » dans le monde Windows ; cf. « pathping ».
[4] Sous Windows, il s'agit du message *echo request* (8/0) ; sous Unix, on utilise un paquet UDP (de port inutilisé, généralement 33434) ou bien le message *port unreachable* (3/3).

En théorie, le fonctionnement d'IP ne garantit pas que 2 messages successifs suivent exactement la même route, rendant le résultat de *traceroute* à priori peu pertinent ; en pratique, les routes empruntées ne varient pas ou peu.

La recherche de MTU [1] optimale (*path MTU discovery*) permet de paramétrer sa connexion pour optimiser le débit.
On utilise pour cela différents messages ICMP *echo request* (8/0) / *echo reply* (0/0) de taille croissante (ou décroissante), associés au bit DF (*Don't Fragment*) positionné à 1, soit donc au message ICMP *fragmentation required* (3/4). La MTU optimale est la MTU maximale ne nécessitant pas de fragmenter le datagramme pour joindre un hôte via la commande `ping` [2]. Ainsi, en suivant au plus près, sans la dépasser, la valeur de MTU optimale pour sa connexion, on optimise le débit de volumes importants de données à transmettre ; en revanche, on détériore le temps de réponse – appelé *ping* – de faibles volumes de données (pouvant être inclus dans un seul datagramme).
À l'inverse, en réduisant fortement sa MTU, on améliore grandement ce temps de réponse ; en contrepartie on perd en débit, car pour un même volume important de données à transmettre il faudra plus de fragments différents, donc autant d'en-têtes supplémentaires, donc au total une plus grande quantité d'octets à transmettre.

Généralement, on recherche la MTU optimale en cherchant à joindre un hôte ressortant du réseau de son FAI ; mais on peut aussi chercher à joindre n'importe quel hôte, ce qui permet ainsi de détecter un éventuel routeur « trou noir » sur la route menant à l'hôte.

4.5.3 Les protocoles ARP et RARP

4.5.3.1 Introduction

L'adresse IP est un identifiant logique, modifiable à volonté, permettant aux hôtes de communiquer librement sans se soucier des interconnexions physiques et des exigences propres à chaque technologie. Néanmoins, lors d'une communication, les données transitent bien physiquement sur le réseau après avoir été encapsulées [3]. Cela sous-entend que l'émetteur a été capable de distinguer le destinataire parmi tous les hôtes connectés physiquement au réseau ; ce, grâce à un identifiant physique, communément appelé *adresse physique* [4] – en opposition à *adresse logique* pour l'adresse IP –, qui dépend directement du module qui interface physiquement l'hôte au réseau (carte réseau).
De fait, pour transmettre physiquement les données au destinataire, l'émetteur a besoin d'un mécanisme d'association entre l'adresse logique (couche 3 OSI, 2 TCP/IP) et l'adresse physique (couche 2 OSI, 1 TCP/IP).

4.5.3.2 Définitions

Le **protocole ARP** (Adresse Resolution Protocol [5]) est un protocole de la couche liaison permettant de déterminer l'adresse physique d'un hôte à partir de son adresse logique.
Chaque hôte maintient à jour une *table ARP* (/ cache ARP) associant une adresse logique à une adresse physique [6]. Pour pallier les changements de matériels ou d'adressage logique, cette table est dynamique et ses entrées ont une durée de vie limitée [7]. Si une communication est à destination d'un hôte non-référencé dans la table ARP, alors une requête ARP est diffusée sur le réseau [8] afin de déterminer son adresse physique. Seul l'hôte ayant reconnu son adresse logique y répond, l'émetteur premier peut ainsi mettre à jour sa table de correspondance.
Du fait des opérations réalisées par les routeurs, les informations ARP sont limitées au même réseau physique [9].

Le **protocole RARP** (Reverse Address Resolution Protocol [10]) est un protocole de la couche liaison permettant de déterminer l'adresse logique d'un hôte à partir de son adresse physique. Il est donc exactement l'inverse du protocole ARP et utilise le même type de messages. Ce protocole est généralement utilisé par un hôte au démarrage si celui-ci n'a pas d'adresse logique configurée afin de s'auto-configurer [11] (cas des hôtes sans disque dur par exemple, type station de travail), et nécessite la mise en place d'un serveur RARP, centralisant de manière statique les associations adresse logique / adresse physique.

---

[1] cf. 4.3.4.
[2] Sous Windows, la commande exacte à passer est « ping –f –l taille destination ». Ex. : « ping –f –l 1464 wikipedia.org » envoie à l'hôte wikipedia.org le message *echo request* qui ne doit pas être fragmenté avec un champ de données de 1464 octets ; si cette commande permet de joindre la destination alors que la même commande avec un champ de données de 1465 octets reçoit le message *fragmentation required*, la MTU optimale est donc 1464 + 8 (en-tête ICMP) + 20 (en-tête IP car ICMP encapsulé dans IP) = 1492 octets.
[3] Principe de communication virtuelle (cf. 3.1.2).
[4] Sur les réseaux type Ethernet, l'adresse physique est appelée adresse MAC.
[5] Address Resolution Protocol (eng) ≡ Protocole de Résolution d'Adresse (fr).
[6] La commande « arp –a » permet d'afficher la table ARP courante.
[7] De l'ordre de quelques minutes : 120 secondes sous Windows.
[8] Ce qui limite l'usage du protocole ARP aux réseaux supportant la diffusion générale.
[9] L'usage d'un proxy ARP permet éventuellement de pallier cette limitation et ainsi de constituer un réseau logique de même identifiant réseau, à partir de plusieurs réseaux physique séparés par un routeur.
[10] Reverse Address Resolution Protocol (eng) ≡ Protocole Inversé de Résolution d'Adresse (fr).
[11] RARP peut être avantageusement remplacé par un serveur DHCP (/BootP) sur les réseaux TCP/IP+Ethernet en proposant une gestion dynamique des adresses.

4.5.3.3 Le message ARP

Un **message ARP** correspond à une requête ou une réponse de résolution d'adresse :
→ 1 message ARP (PDU) : 12 – ? octets ;
- 1 champ en-tête (PCI) : 8 octets ;
- 1 champ données (SDU) : 4 – ? octets.

Le message ARP est encapsulé directement dans une trame de communication du réseau physique.

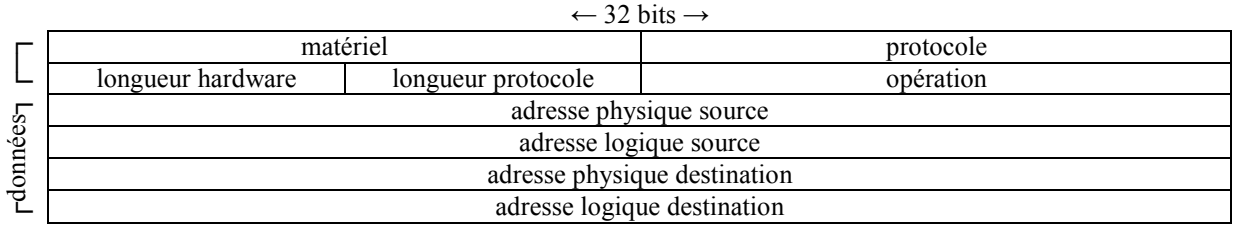

Figure 4.17 : message ARP

Les différentes informations du champ en-tête sont les suivantes :
- matériel (16) : type d'interface matérielle mis en œuvre, 1 pour Ethernet ;
- protocole (16) : type d'adressage logique mis en œuvre, 2048 (0x800) pour l'adressage IP ;
- longueur hardware (8) : longueur des adresses physiques utilisées, 6 pour Ethernet, 1 pour Token Ring ;
- longueur protocole (8) : longueur des adresses logiques utilisées, 4 pour IPv4, 16 pour IPv6 ;
- opération (16) : type d'opération réalisée, 1 pour une requête ARP, 2 pour une réponse ARP, 3 pour une demande RARP, 4 pour une réponse RARP.

Les différentes informations du champ données sont les suivantes :
- adresse physique source (?), adresse physique destination (?) : adresses physiques ; 0 pour l'adresse physique destination en cas de requête ARP (adresse de diffusion, car celle recherchée est encore indéterminée) ;
- adresse logique source (?), adresse logique destination (?) : adresses logiques.

Le protocole ARP pouvant être mis en œuvre sur diverses technologies de réseaux par divers protocoles de la couche réseau, la taille des adresses physiques et logiques n'est pas figée, et est connue grâce aux champs *longueur hardware* et *longueur protocole* (ex. : 48 bits = 6 octets dans le cas d'Ethernet).

4.5.4 Le système DNS

4.5.4.1 Définitions

Le **système DNS** (Domain Name System [1]) est un système d'annuaire associant un nom alphanumérique à une adresse IP.

Le but principal de ce système est de désigner un hôte avec une appellation beaucoup plus facilement mémorisable qu'une adresse IP ; par ailleurs, on dispose d'un système de nom masquant les spécificités d'une adresse IP [2], permettant un changement d'adresse IP transparent. Un nom DNS correspond donc généralement à 1 seule adresse IP [3], alors qu'une adresse IP peut cependant être associée à plusieurs noms DNS [4].

Historiquement, les associations IP / nom DNS étaient définies dans un fichier (fichier *hosts* [5]), stocké en local sur chaque machine du réseau ARPAnet. Ce fichier devant être identique pour toutes les machines, il est évident que cette méthode statique est inadaptée sur les gros réseaux.

Une solution est alors d'utiliser une machine serveur mettant à disposition les données du fichier global d'associations IPs / noms DNS [6]. Toute machine du réseau s'adresse à ce serveur pour déterminer l'adresse IP connaissant le nom DNS [7], fonctionnalité appelée *résolution de nom* qui est exécutée lors d'une requête DNS [8].

[1] Domain Name System (eng) ≡ Système de Nom de Domaine (fr).
[2] Classe d'adresse, identifiant de réseau, identifiant d'hôte, …
[3] Au-delà des systèmes de répartition de charge réseau, type DNS round-robin, utilisés pour des domaines générant un trafic important.
[4] Principe d'alias à un nom DNS.
[5] /etc/hosts sous Unix, ou <Windows>\system32\drivers\etc\hosts sous Windows, fichier qui peut encore être utilisé pour référencer les hôtes sur un LAN (ne serait-ce que 1 ou 2 serveurs).
[6] Principes du système NIS (Network Information Services) développé par la société Sun.
[7] Les machines du réseau doivent donc connaître la machine serveur DNS par son adresse IP et pas par son nom, sinon le problème reste entier.
[8] Pour exécuter une requête DNS (*quelle IP correspond à ce nom DNS ?*) ou une requête DNS inversée – on parle de *résolution inverse* – (*quel nom DNS correspond à cette IP ?*) ; on utilise la commande « nslookup » sous Windows et « host » et « dig » sous Unix.

Cependant, étant donné le nombre de domaines existant sur un réseau tel que l'internet, un tel système centralisé ne peut être maintenu à jour efficacement par une seule autorité [1]. Une solution dans laquelle plusieurs responsabilités se partagent le travail lui est donc préférable.

DNS est basé sur une architecture client/serveur, avec un principe de requête/réponse utilisant le **protocole DNS** (Domain Name Service [2]), protocole de la couche session (5 OSI), qui définit les messages échangés pour la résolution d'un nom. Chacun de ces messages est encapsulé dans un datagramme UDP [3], le serveur écoutant sur le port 53.

4.5.4.2 Organisation

DNS est un système distribué et hiérarchisé en arborescence. Plus précisément, chaque élément du niveau de l'arborescence est géré par une entité autonome [4], qui a en charge d'assurer le bon fonctionnement du service et l'unicité des noms pour tous les domaines qui sont de son ressort [5].

Cette organisation en niveaux se lit directement dans le nom DNS complet d'un hôte, appelé *nom FQDN* (Fully Qualified Domain Name [6]), qui indique le nom de chaque niveau séparé par un point ('.'), selon le modèle *hôte . sous-domaine . domaine . domaine-haut-niveau*, en commençant par le dernier niveau [7]. Celui-ci correspond donc toujours à l'hôte, et par extension le reste du nom est assimilé au domaine lui-même, aussi appelé *zone*.

Le nombre de niveaux maximal est 127 (découpage du domaine en sous-domaine, puis de chaque sous-domaine en sous-sous-domaine, etc.) ; chaque nom de niveau fait de 1 à 63 caractères, le nom FQDN faisant 255 caractères au maximum. Le domaine racine est nommé « point » (noté '.') [8], et est généralement omis dans les requêtes DNS [9].

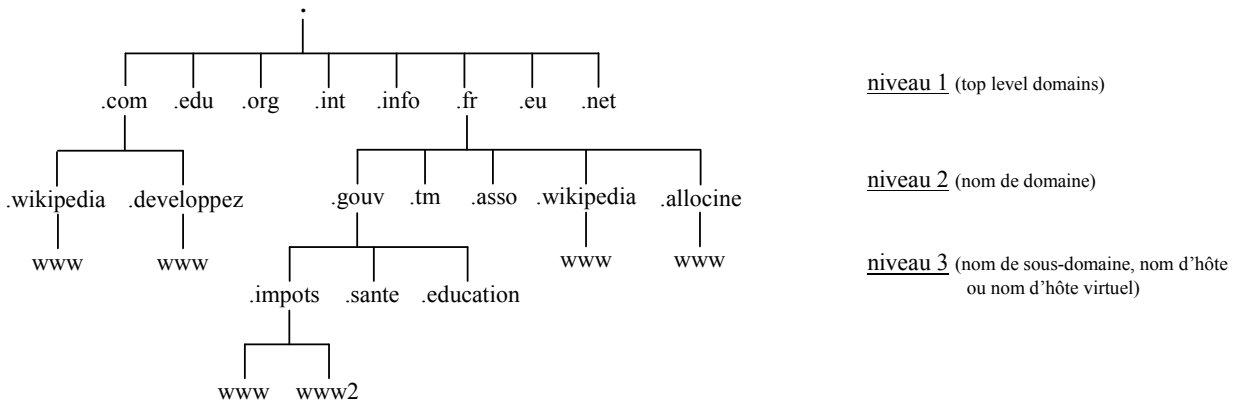

Figure 4.18 : aperçu de la hiérarchie DNS

Ex. : L'adresse internet *www.wikipedia.fr* désigne l'hôte (/ la machine) de nom *www* du domaine *wikipedia.fr* ; *www* et *www2* désignent 2 hôtes du domaine *impots.gouv.fr* (sous-domaine de *gouv.fr*).

4.5.4.3 Principes de résolution de nom DNS

Le principe d'un système DNS distribué et mis en œuvre suivant un mécanisme [10] client/serveur suppose un système de cache DNS, afin de stocker temporairement les résolutions d'adresses pour optimiser le trafic. Lorsque les données du cache ne permettent pas de déterminer l'adresse IP, la hiérarchie DNS est alors mise en jeu.

Un serveur DNS, lorsqu'il ne peut répondre directement à la requête (pas d'autorité sur le domaine visé et pas de mise en cache), peut fonctionner suivant 2 modes :

---

[1] Sans parler des questions politiques et économiques…

[2] Domain Name Service (eng) ≡ Service de Nom de Domaine (fr).

[3] Le protocole TPC est parfois utilisé lorsque la requête DNS dépasse 512 octets.

[4] Au niveau mondial, c'est l'ICANN (Internet Corporation for Assigned Names and Numbers) qui gère le système DNS, et qui délègue la gestion des domaines de haut niveau (top level domain) à divers organismes, comme l'AFNIC (Association Française pour le Nommage Internet en Coopération) qui a en charge le domaine *.fr*.

[5] Ces entités mettent ainsi à disposition 1 serveur principal dit *serveur primaire* afin de répondre aux requêtes DNS, et un ou plusieurs serveurs redondants dits *serveurs secondaires*.

[6] Fully Qualified Domain Name (eng) ≡ Nom de Domaine Pleinement Défini (fr).

[7] Le nom FQDN est insensible à la casse.

[8] En 2008, il existe 16 serveurs redondants gérant le domaine racine « point », nommés de *a.root-servers.net* à *p.root-servers.net* : 13 sont situés aux USA, 1 au Japon, 1 au Royaume-Uni, 1 aux Pays-Bas.

[9] On devrait ainsi écrire « *hôte . domaine . domaine-haut-niveau .* » et non « *hôte . domaine . domaine-haut-niveau* » (différence sur le « point » final représentant le domaine racine).

[10] Appelé *resolveur*.

- mode itératif : le serveur DNS (a) indique au client un autre serveur DNS (b) qu'il sait être plus approprié ;
- mode récursif : le serveur DNS (a) gère la requête DNS comme s'il était demandeur et va interroger le serveur DNS (b) qu'il sait être plus approprié ; lorsque la réponse lui parvient, il la transmet au client (le fait que le serveur DNS (b) fonctionne en mode itératif ou récursif est transparent).

Le mode itératif est le mode de fonctionnement par défaut d'un serveur DNS.

Voici les différentes étapes d'une résolution de nom utilisant un serveur DNS, lorsque l'hôte source *mhn.src.fr* veut joindre la destination *www.dst.com* :

① L'hôte *mhn* recherche *www.dst.com* dans son fichier hosts ;
② Sans succès, *mhn* interroge le serveur DNS *dns.src.fr* dont il dépend, en lui envoyant une requête de résolution concernant *www.dst.com* ;
③ Le serveur *dns.src.fr* ne gère pas le domaine *.dst.com* et n'a pas l'information en cache, il transmet alors la requête au serveur DNS du domaine racine « point » *dns.* ;
④ Le serveur *dns.* ne gère pas le domaine *.com* et n'a pas l'information en cache, mais a autorité sur le domaine « point » et il connaît donc le serveur DNS *dns.com* ;
⑤ Le serveur *dns.* communique au serveur *dns.src.fr* l'adresse du serveur *dns.com* ;
⑥ Le serveur *dns.src.fr* met en cache l'adresse du serveur *dns.com* et re-transmet sa requête en interrogeant alors ce serveur ;
⑦ Le serveur *dns.com* n'a pas l'information en cache, mais a autorité sur le domaine *.com* et il communique donc à *dns.src.fr* l'adresse du serveur DNS *dns.dst.com* ;
⑧ Le serveur *dns.src.fr* met en cache l'adresse du serveur *dns.dst.com* et re-transmet sa requête en interrogeant ce serveur ;
⑨ Le serveur *dns.dst.com* a autorité sur le domaine *.dst.com* et connaît donc l'adresse IP de l'hôte *www.dst.com,* il renvoie donc la réponse au serveur *dns.src.fr* ;
⑩ Le serveur *dns.src.fr* met la réponse en cache, et transmet la réponse à *mhn.src.fr*.

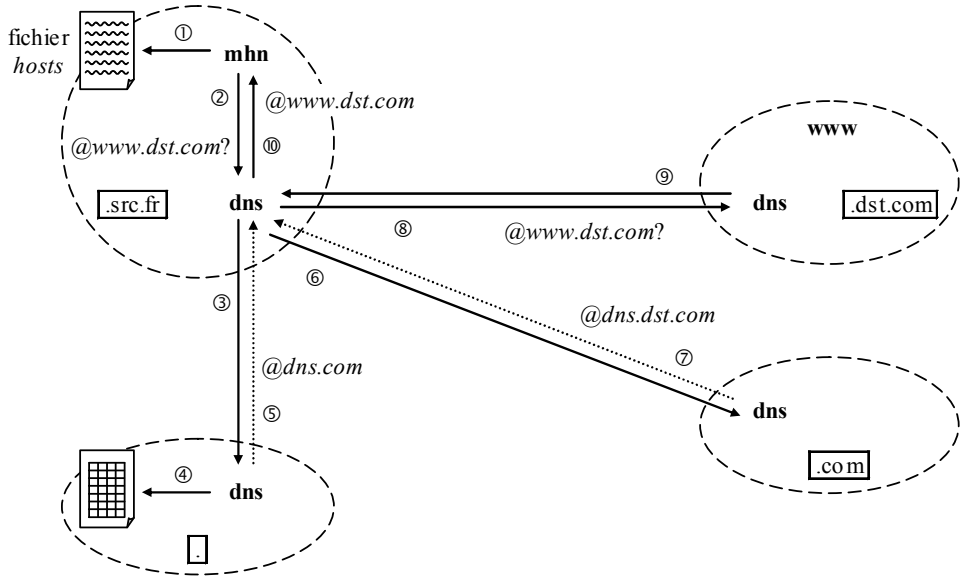

Figure 4.19 : résolution de nom via une requête DNS

Nb : Dans cet exemple, le serveur DNS *dns.src.fr* fonctionne en mode récursif, et les autres en mode itératif. Ce dernier est le mode de fonctionnement généralement adopté : le serveur local (serveur ayant une faible charge réseau) gère la requête DNS pour libérer le client, alors que les autres serveurs DNS délèguent la requête (les serveurs de premier et second niveaux sont extrêmement sollicités).

Le mode itératif est donc généralement dédié à des serveurs de domaine, ayant autorité sur 1 ou plusieurs domaines, alors que le mode récursif est utilisé par des serveurs cache (/ de relais), n'ayant autorité sur aucun domaine particulier [1] mais dont l'usage permet ainsi de décharger les serveurs de domaines.

[1] Caractéristique identifiable par le message *réponse ne faisant pas autorité* lors d'une requête DNS en usant de « nslookup ».

4.5.5 Le protocole DHCP

4.5.5.1 Définitions

Le **protocole DHCP** (Dynamic Host Configuration Protocol [1]) est un protocole de la couche réseau (3 OSI) de type client/serveur, utilisé par un hôte afin de configurer ses paramètres réseau conformément au réseau sur lequel il est connecté, lui permettant ainsi de s'intégrer à l'ensemble de ses hôtes, et de communiquer avec eux. Ces paramètres sont déterminés et centralisés par un serveur DHCP ; le client doit donc lui envoyer une requête afin de s'auto-configurer.

Les informations minimum transmises sont l'adresse IP et le masque, ce qui permet ainsi de se connecter au réseau ; mais on peut aussi préciser le(s) serveur(s) DNS afin de réaliser des translations de noms, la passerelle afin d'accéder à d'autres réseaux interconnectés avec le réseau courant, le nom de domaine, le serveur de mail (SMTP et POP3), le serveur de temps, etc.

Pour l'hôte, une configuration des paramètres réseau par DHCP est dynamique, et est donc préférée à une configuration statique pour simplifier les opérations de paramétrage.

De plus, cela permet un nombre d'hôtes connectables au réseau bien supérieur au nombre d'hôtes permis par le réseau (adresse IP & masque), tant que ces connexions ne sont pas simultanées. En effet, la gestion des adresses IP au niveau du serveur DHCP est dynamique, et une adresse IP est allouée pour un temps donné – appelé *bail* – ; à l'expiration du bail, l'adresse IP est à nouveau disponible pour n'importe quel hôte se connectant au réseau.

Enfin, en cas de changement d'un ou plusieurs des paramètres réseau, seule la configuration du serveur DHCP doit être modifiée.

Nb : Il est possible de réserver une adresse IP spécifique pour un hôte donné, assurant ainsi que l'hôte se voit toujours allouer la même adresse sur ce réseau. Cette réservation statique [2] se base sur l'adresse physique de l'hôte, rapprochant ainsi le protocole DHCP du protocole BootP, utilisé par un hôte pour se voir allouer une adresse IP au démarrage.

4.5.5.2 Principe d'une communication DHCP

Les messages échangés afin de configurer les paramètres réseaux sont encapsulés dans des datagrammes UDP en utilisant les ports 67 (serveur DHCP en écoute pour les requêtes) et 68 (client DHCP en écoute pour les réponses). La communication s'effectue en 4 temps, en usant de 4 messages DHCP distincts :

- DHCPDiscover : diffusion d'une requête DHCP par l'hôte demandant à se voir allouer une adresse IP (adresse IP source : 0.0.0.0, adresse IP destination : 255.255.255.255) en précisant son adresse physique ;
- DHCPOffer : tous les serveurs DHCP ayant reçu la requête répondent en diffusant un message qui propose un paramétrage adresse IP / masque ainsi que la durée du bail associé ;
- DHCPRequest : l'hôte accepte la première proposition reçue, s'auto-configure, et diffuse un message précisant ses différents paramètres réseau ;
- DHCPAck : le serveur concerné mémorise l'allocation de cette adresse, démarre le bail et envoie une confirmation au client.

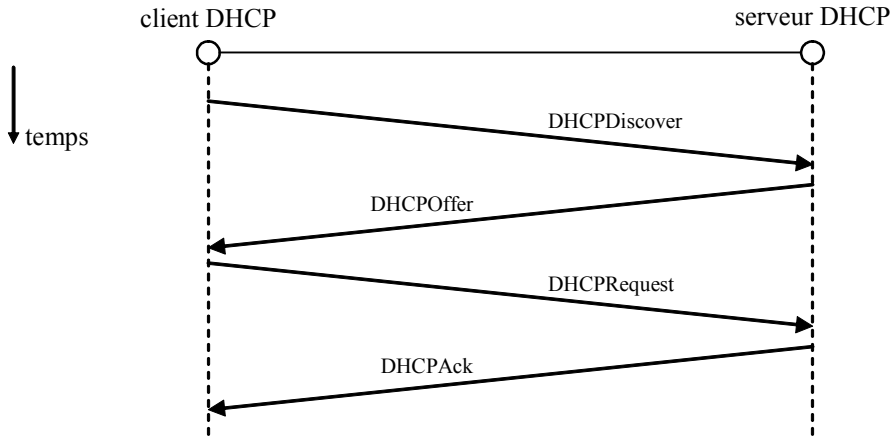

Figure 4.20 : communication DHCP

[1] Dynamic Host Configuration Protocol (eng) ≡ Protocole de Configuration Dynamique d'Hôte (fr).

[2] Cette configuration reste tout de même dynamique du point de vue du client ; elle est donc à distinguer d'une configuration manuelle statique directement sur l'hôte lui-même.

Les autres messages DHCP existants pouvant être envoyés sont :

- DHCPNAck : le serveur informe le client que son bail est arrivé à expiration ;
- DHCPDecline : le client refuse l'adresse IP proposée car celle-ci est déjà utilisée ;
- DHCPRelease : le client libère l'adresse IP et met fin au bail ;
- DHCPInform : le client demande les paramètres de configuration locaux (demande effectuée après avoir obtenu une adresse IP).

4.5.6 Le protocole IPv6

4.5.6.1 Présentation

Le **protocole IPv6** est l'évolution du protocole IPv4 et propose un certain nombre d'améliorations et d'ajouts par rapport au protocole IPv4 et vise ainsi à résoudre différents problèmes mis en évidence par l'utilisation à grande échelle d'IPv4 :

- utilisation d'un nouvel adressage IPv6 remplaçant l'adressage IPv4 afin de pallier les pénuries d'adresses ;
- mécanismes de configuration et renumérotation automatiques ;
- simplification des en-têtes de paquets ;
- implémentation native des protocoles IPsec (IP SECure : chiffrement des données et authentification), QoS (Quality of Service : gestion et optimisation du trafic en fonction du type de données) et multicast.

4.5.6.2 L'adresse IPv6

L'**adresse IPv6** est l'identifiant logiciel unique d'un nœud sur le réseau. Elle est codée sur 128 bits (16 octets) regroupés en 8 couples d'octets, et est généralement notée *aaaa:bbbb:cccc:dddd:eeee:ffff:gggg:hhhh* (dite « notation canonique ») avec *aaaa*, …, *hhhh* compris entre 0x0000 et 0xFFFF (0 et 65535).

Le potentiel est donc de 2^{128} adresses différentes soit ≈ $3{,}4.10^{38}$; à titre indicatif, ce potentiel d'adresses rapporté à la superficie totale de la planète terre permettrait de disposer de 667.10^{21} adresses par m², ou 227.10^{22} adresses par m² de surface terrestre.

L'adresse IPv6, tout comme l'adresse IPv4, contient 2 informations : l'identifiant de réseau, et l'identifiant d'hôte, chacun codé sur 64 bits.

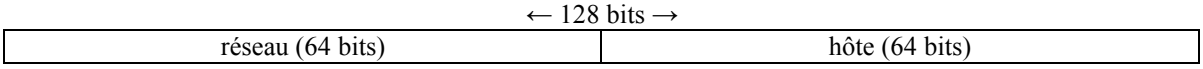

Figure 4.21 : position des identifiants de réseau et d'hôte dans une adresse IPv6

Lorsque l'un des couples d'octets vaut 0000, il peut être noté 0, voire complètement omis ; on a alors une notation :: (2 x deux-points), et l'adresse peut être notée *aaaa:bbbb::dddd:eeee:ffff:gggg:hhhh*.

Lorsque plusieurs couples d'octets consécutifs valent 0000, ils peuvent être omis ; on conserve alors une notation :: (2x deux-points), et l'adresse peut être notée *aaaa:bbbb::eeee:ffff:gggg:hhhh* (attention : cette écriture réduite ne peut être réalisée que pour un seul groupe d'octets consécutifs dans l'adresse IPv6).

Lorsque l'on désigne un hôte, l'adresse IPv6 doit être mise entre crochets, ceci afin de pouvoir spécifier le numéro de port sans risque de confusion : [*aaaa:bbbb:cccc:dddd:eeee:ffff:gggg:hhhh*]:port.

4.5.6.3 Conventions d'adressage IPv6

L'adressage IPv6 suit aussi des conventions :

- sous-réseau 2000:: / 3 : réservé à l'internet IPv6 (les autres adresses ne sont donc pas routables sur internet) ;
- sous-réseau 2002:: / 16 (6to4) : destiné à acheminer du trafic IPv6 via des réseaux IPv4 ;
- sous-réseaux 1fff:: / 16, 3ffe:: / 16 et 5f00:: / 16 (6bone) : utilisé pour l'expérimentation d'IPv6 (inutilisés depuis 2006) ;
- sous-réseau fe80:: / 64 : adresses locales (adressage privé) ;
- sous-réseau fe80:: / 96 : sous-groupe du sous-réseau fe80:: / 64 proposant 32 bits qui peuvent être considérés comme les 32 bits d'une adresse IPv4, dont l'adresse peut alors être notée *aaaa:bbbb:cccc:dddd:eeee:ffff:xxx.yyy.zzz.ttt* avec aaaa, …, ffff compris entre 0x0000 et 0xFFFF, et xxx, …, ttt compris entre 0 et 255 ; cette forme comprend donc une partie notée en hexadécimal (96 premiers bits) et une partie notée en décimal (32 derniers bits) ; cette correspondance de l'adresse IPv4 dans l'adresse IPv6, permet d'allouer automatiquement l'adresse IPv6 *aaaa:bbbb:cccc:dddd:eeee:ffff:xxx.yyy.zzz.ttt* à une interface ayant l'adresse IPv4 *xxx.yyy.zzz.ttt* ;
- adresse :: ou ::0.0.0.0 : adresse nulle ;
- adresse ::1 : adresse de bouclage (loopback) pour tester le localhost.

4.5.6.4 Le datagramme IPv6

Un **datagramme** IPv6 correspond aux données émises de la couche supérieure encapsulées dans une trame constituée de 2 ou 3 champs :

→ 1 datagramme IP (PDU) : 40 – 65535 octets ;

- 1 champ en-tête (PCI) : 40 octets pour 8 informations ;
- 1 ou plusieurs champs en-têtes d'extension (PCI) : ? octets ;
- 1 champ données (SDU) : 0 – 65495 octets.

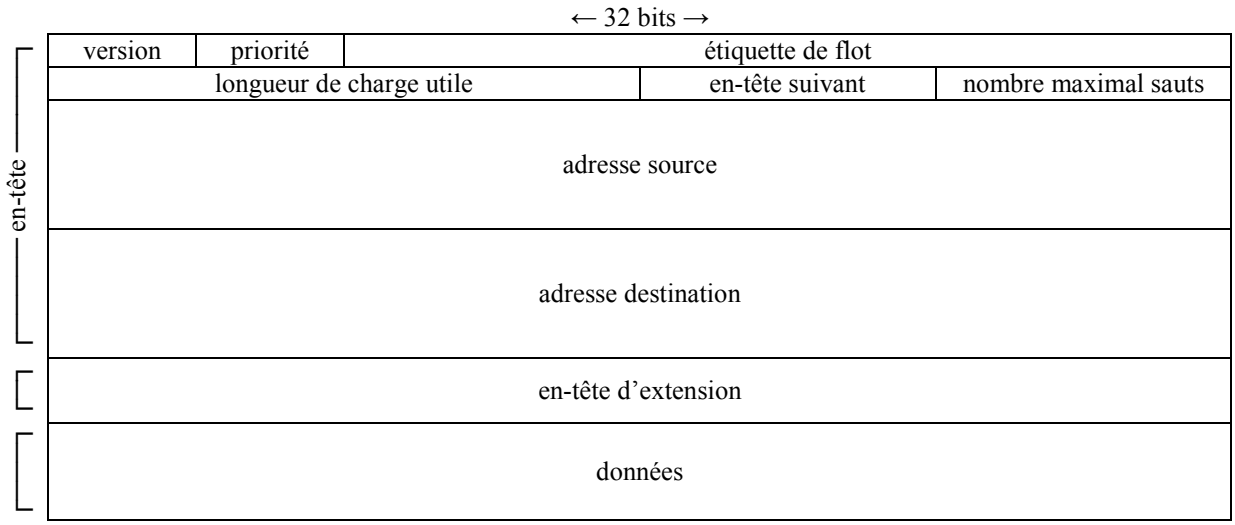

Figure 4.22 : datagramme IPv6

Les différentes informations du champ en-tête sont les suivantes :

- version (4) : numéro de version du protocole IP, 4 pour IPv4, 6 pour IPv6 ;
- priorité (4) : indicateur de contrôle de flux, 0 – 7 pour du trafic capable d'adapter le débit en cas de congestion, 8 – 15 pour du trafic temps réel ;
- étiquette de flot (24) : indicateur de transmission à condition particulière (champ expérimental) ;
- longueur de charge utile (16) : longueur du champ données (/ charge utile) en octets (SDU) ;
- en-tête suivant (8) : type de l'éventuel en-tête d'extension suivant ;
- nombre maximal de sauts (8) : durée de vie du datagramme lors du transit, décrémenté de 1 à chaque passage via un routeur, ou plus s'il stagne dans sa file d'attente, détruit si la durée de vie est atteinte (= 0) (champ identique au champ TTL d'IPv4) ;
- adresse source (128) : adresse IPv6 du nœud source sur 16 octets ;
- adresse destination (128) : adresse IPv6 du nœud destination sur 16 octets.

Le ou les éventuels champs en-têtes d'extension fournissent des informations complémentaires. Il en existe 6 types définis :

- pas-après-pas : informations destinées aux routeurs ;
- routage : informations sur la route partielle ou totale à suivre (ce qui se traduit par une liste de routeurs) ;
- fragmentation : informations de fragmentation (seul le nœud source peut fragmenter le datagramme) ;
- authentification : mécanisme d'authentification du nœud source ;
- charge utile chiffrée : informations de cryptage du champ données ;
- option de destination : informations spécifiques pour le nœud destination (actuellement inutilisé).

A HISTOIRE DES TÉLÉCOMMUNICATIONS

1464 : Institutionnalisation en France de la communication par voie écrite avec la mise en place de la Poste royale, sous le règne de Louis XI.
1794 : Invention du télégraphe optique par C. Chappe : tours placées en hauteur et visibles de loin constituées de grands bras articulés définissant les caractères par position relative.
1832 : Invention du télégraphe électrique par P. Shilling, surnommé *le fil qui chante*.
1837 : Création de l'Administration du Télégraphe par le ministère de l'intérieur américain.
Invention du code Morse par S. Morse : alphabet télégraphique constitué de points et traits, symbolisés par la transmission d'un signal visuel ou sonore court ou long.
1854 : Mise en place d'un premier projet de téléphone par C. Bourseuil.
1876 : Dépôt du brevet du téléphone par G. Bell.
1879 : Création du ministère des Postes et du Télégraphe (*P et T*).
1889 : Nationalisation de la Société Française de Téléphone.
1896 : Création de la première liaison TSF (Transmission Sans Fil) par G. Marconi.
1901 : Mise en place de la liaison transatlantique.
1917 : Mise au point du code Baudot utilisé sur le réseau télégraphique commuté (Télex) par E. Baudot : code alphanumérique à 5 bits.
1969 : Début de la mise en place de ARPAnet, ancêtre d'Internet, par l'agence de projets de recherche avancée (Advanced Research Projects Agency) du département de la défense des États-unis (DoD : Department of Defense) afin d'assurer les communications entre les centres névralgiques militaires face à une potentielle menace d'attaque nucléaire (période de guerre froide).
1983 : Scission de ARPAnet en MilNet (MILitary), dédié au gouvernement et à l'armée, et en NSFnet (National Science Foundation), utilisé par la communauté universitaire, pour des raisons évidentes de sécurité dues à l'accroissement du réseau historique.
Mise en place du système DNS.
1984 : Publication des documents de référence du modèle OSI (Interconnexion des Systèmes Ouverts) par l'ISO (Organisme de Normalisation International) : ISO 7498, (X.200 pour le CCITT), après une étude ayant débuté en 1977.
1995 : Validation des conventions d'adressage IPv6.

B RÉFÉRENCE

B.1 LISTE DES RFC

Les RFC (Request For Comments) étaient originellement des documents visant à résoudre les problèmes d'architecture liées aux réseaux téléinformatiques. À l'heure actuelle, ils constituent un ensemble de documents définissant et décrivant les règles à suivre pour respecter un protocole de communication, des règles de communication, des propositions de protocoles, ou bien des mises à jour de RFC déjà diffusées.

RFC 768 : UDP (User Datagram Protocol) ;
RFC 791 : IP (Internet Protocol) ;
RFC 792 : ICMP (Internet Control Message Protocol) ;
RFC 793 : TCP (Transfer Control Protocol) ;
RFC 826 : ARP (Address Resolution Protocol) ;
RFC 827 : EGP (Exterior Gateway Protocol) ;
RFC 854 : Telnet (terminal virtuel) ;
RFC 867 : protocole DayTime ;
RFC 868 : protocole Time ;
RFC 894 : IP over Ethernet (PPPoE) ;
RFC 903 : RARP (Reverse Address Resolution Protocol) ;
RFC 950 : procédure de construction de sous-réseaux (subnetting) ;
RFC 951 : BootP (BOOTstrap Protocol) ;
RFC 959 : FTP (File Transfer Protocol) ;
RFC 1001 et 1002 : NetBIOS ;
RFC 1034 et 1035 : DNS (Domain Name System) ;
RFC 1042 : IP over Token Ring ;
RFC 1157 : SNMP (Simple Network Management Protocol) ;
RFC 1166 : adressage IP (classes, etc.) ;
RFC 1180 : couche de protocoles TCP/IP ;
RFC 1188 : IP over FDDI ;
RFC 1247 : OSPF (Open Shortest Path First) ;
RFC 1267 : BGP (Border Gateway Protocol v3).
RFC 1305 : NTP (Network Time Protocol v3) ;
RFC 1350 : TFTP (Trivial File Transfer Protocol rev2) ;
RFC 1403 : interfaçage BGP / OSPF ;
RFC 1459 : IRC (Internet Relay Chat protocol) ;
RFC 1661 : PPP (Point-to-Point Protocol) ;
RFC 1855 : Netiquette (règles de respect et de savoir-vivre des communications via mail, forum, chat, news, etc.) ;
RFC 1918 : adressage IP pour réseaux privés ;
RFC 1932 : IP over ATM (PPPoA) ;
RFC 1939 : POP3 (Post Office Protocol v3) ;
RFC 1945 : HTTP/1.0 (HyperText Transfer Protocol v1.0) ;
RFC 2045 – 2047, 4289 et 2049 : MIME (Multipurpose Internet Mail Extensions) ;
RFC 2131 : DHCP (Dynamic Host Configuration Protocol) ;
RFC 2228 : FTPS (File Transfer Protocol over Secure Sockets Layer) ;
RFC 2328 : OSPF (Open Shortest Path First v2) ;
RFC 2440 : OpenPGP (Pretty Good Privacy) ;
RFC 2453 : RIP (Routing Information Protocol v2) ;
RFC 2460 : spécification IPv6 ;
RFC 2581 : contrôle de congestion de TCP ;

RFC 2595 : POP3S (Post Office Protocol v3 over Secure Sockets Layer) / IMAP4S (Interactive Mail Access Protocol v4 over Secure Sockets Layer) ;
RFC 2616 : HTTP/1.1 (HyperText Transfer Protocol v1.1) ;
RFC 2818 : HTTPS (HyperText Transfer Protocol over Secure Sockets Layer) ;
RFC 2821 : SMTP (Simple Mail Transfer Protocol) ;
RFC 3022 : NAT (Network Address Translation) ;
RFC 3501 : IMAP4 (Interactive Mail Access Protocol v4 rev1) ;
RFC 3977 : NNTP (Network News Transfer Protocol) ;
RFC 4250 – 4254 : SSH (Secure SHell) ;
RFC 4271 : BGP (Border Gateway Protocol v4) ;
RFC 4346 : TLS (Transport Layer Security v1.1) / SSL (Secure Sockets Layer v3.1) ;
RFC 4347 : DTLS (Datagram Transport Layer Security) ;
RFC 4632 : CIDR (Classless Inter-Domain Routing) ;
RFC 5424 : protocole Syslog (SYStem LOG).

Pour plus d'informations, voir la liste des RFC « actives » (non-rendues obsolètes par un document plus récent) : http://www.faqs.org/rfcs/rfc-activeT.html.

B.2 LISTE DES PORTS DE CONNEXION RECONNUS

Voici une liste non-exhaustive des *well-known ports* tels que définis par l'IANA (Internet Assign Numbers Authority), chargé de coordonner un certain nombre de standards sur l'internet (adressage IP, noms de domaines, etc.).

13 : DayTime ;
20-21 : FTP (File Transfer Protocol) ;
22 : SSH (Secure SHell) / SFTP (SSH File Transfer Protocol) ;
23 : Telnet (terminal virtuel) ;
25 : SMTP (Simple Mail Transfer Protocol) ;
37 : protocole Time ;
53 : DNS (Domain Name System) ;
67-68 : DHCP (Dynamic Host Configuration Protocol) / BootP (BOOTstrap Protocol) ;
69 : TFTP (Trivial File Transfer Protocol) ;
80 : HTTP (HyperText Transfer Protocol) ;
110 : POP3 (Post Office Protocol v3) ;
119 : NNTP (Network News Transfer Protocol) ;
123 : NTP (Network Time Protocol) ;
137-139 : NetBios ;
143 : IMAP4 (Interactive Mail Access Protocol v4) ;
161-162 : SNMP (Simple Network Management Protocol) ;
179 : BGP (Border Gateway Protocol) ;
194 : IRC (Internet Relay Chat) ;
443 : HTTPS (HyperText Transfer Protocol over Secure Sockets Layer) ;
445 : partage de fichiers et d'imprimante Windows / SaMBa ;
514 : SysLog (SYStem LOGging) ;
520 : RIP (Routing Information Protocol) ;
989-990 : FTPS (File Transfer Protocol over Secure Sockets Layer) ;
993 : IMAP4S (Interactive Mail Access Protocol v4 over Secure Sockets Layer) ;
995 : POP3S (Post Office Protocol v3 over Secure Sockets Layer).

Pour plus d'informations, voir la liste complète : http://www.iana.org/assignments/port-numbers.

B.3 LISTE DES DOMAINES DE HAUT NIVEAU

Il existe deux catégories de domaines de haut niveau, notés TLD (Top Level Domain).

B.3.1 Domaines génériques

Les domaines génériques, notés gTLD (Generic Top Level Domain), sont des domaines de haut niveau classés par secteur d'activité.

.aero : industrie aéronautique ;
.arpa : réseau historique ARPAnet ;
.biz : entreprises commerciales ;
.com : domaines très courants, pour tout style de sujet et d'intérêt ; à l'origine pour les entreprises commerciales ;
.coop : coopératives ;
.edu : organismes éducatifs ;
.gov : organismes gouvernementaux ;
.info : organismes liés à l'information ;
.int : organisations internationales ;
.mil : organismes militaires ;
.museum : musées ;
.name : noms de personnes ou de personnages imaginaires ; destiné au particulier ;
.net : domaines très courants, image moins commerciale que *.com* ; à l'origine pour les organismes liés aux réseaux ;
.org : organismes à buts non lucratifs ;
.pro : professions libérales.

B.3.2 Domaines nationaux

Les domaines nationaux, notés ccTLD (Country Code Top Level Domain), sont des domaines de haut niveau correspondant aux différents pays.

code	pays	code	pays	code	pays
.ac	Île de l'Ascension	.cd	République démocratique du Congo	.ga	Gabon
.ad	Andorre			.gb	Grande-Bretagne
.ae	Émirats Arabes Unis	.cf	République Centrafricaine	.gd	Grenade
.af	Afghanistan	.cg	Congo	.ge	Géorgie
.ag	Antigua et Barbuda	.ch	Suisse	.gf	Guyane Française
.ai	Anguilla	.ci	Côte d'Ivoire	.gg	Guernesey
.al	Albanie	.ck	Îles Cook	.gh	Ghana
.am	Arménie	.cl	Chili	.gi	Gibraltar
.an	Antilles Néerlandaises	.cm	Cameroun	.gl	Groenland
.ao	Angola	.cn	Chine	.gm	Gambie
.aq	Antarctique	.co	Colombie	.gn	Guinée
.ar	Argentine	.cr	Costa Rica	.gp	Guadeloupe
.as	Samoa Américaines	.cu	Cuba	.gq	Guinée Équatoriale
.at	Autriche	.cv	Cap Vert	.gr	Grèce
.au	Australie	.cx	Île Christmas	.gs	Géorgie du Sud
.aw	Aruba	.cy	Chypre	.gt	Guatemala
.az	Azerbaïdjan	.cz	République Tchèque	.gu	Guam (USA)
.ba	Bosnie-Herzégovine	.de	Allemagne	.gw	Guinée-Bissau
.bb	Barbade	.dj	Djibouti	.gy	Guyana
.bd	Bangladesh	.dk	Danemark	.hk	Hong Kong
.be	Belgique	.dm	Dominique	.hm	Îles Heard and McDonald
.bf	Burkina Faso	.do	République Dominicaine	.hn	Honduras
.bg	Bulgarie	.dz	Algérie	.hr	Croatie
.bh	Bahreïn	.ec	Équateur	.ht	Haïti
.bi	Burundi	.ee	Estonie	.hu	Hongrie
.bj	Bénin	.eg	Égypte	.id	Indonésie
.bm	Bermudes	.eh	Sahara Occidental	.ie	Irlande
.bn	Brunei	.er	Érythrée	.il	Israël
.bo	Bolivie	.es	Espagne	.im	Île de Man
.br	Brésil	.et	Éthiopie	.in	Inde
.bs	Bahamas	.eu	Europe	.io	Territoires Britanniques de l'océan Indien
.bt	Bhoutan	.fi	Finlande		
.bv	Île Bouvet	.fj	Fidji	.iq	Irak
.bw	Botswana	.fk	Îles Falkland (Malouines)	.ir	Iran
.by	Biélorussie	.fm	Micronésie	.is	Islande
.bz	Belize	.fo	Îles Féroé	.it	Italie
.ca	Canada	.fr	France	.jm	Jamaïque
.cc	Îles Cocos	.fx	France (Territoire Européen)	.jo	Jordanie
				.jp	Japon

.ke	Kenya	.nf	Îles Norfolk	.st	Sao Tomé et Principe
.kg	Kirghizistan	.ng	Nigeria	.su	Union Soviétique
.kh	Cambodge	.ni	Nicaragua	.sv	Salvador
.ki	Kiribati	.nl	Pays-Bas	.sy	Syrie
.km	Comores	.no	Norvège	.sz	Swaziland
.kn	Saint Kitts et Nevis	.np	Népal	.tc	Îles Turks et Caicos
.kp	Corée du Nord	.nr	Nauru	.td	Tchad
.kr	Corée du Sud	.nt	Zone Neutre	.tf	Territoire Austral Français
.kw	Koweït	.nu	Niue	.tg	Togo
.ky	Îles Caïmans	.nz	Nouvelle-Zélande	.th	Thaïlande
.kz	Kazakhstan	.om	Oman	.tj	Tadjikistan
.la	Laos	.pa	Panamá	.tk	Tokelau
.lb	Liban	.pe	Pérou	.tm	Turkménistan
.lc	Sainte-Lucie	.pf	Polynésie française	.tn	Tunisie
.li	Liechtenstein	.pg	Papouasie Nouvelle-Guinée	.to	Tonga
.lk	Sri Lanka	.ph	Philippines	.tp	Timor Est
.lr	Libéria	.pk	Pakistan	.tr	Turquie
.ls	Lesotho	.pl	Pologne	.tt	Trinité et Tobago
.lt	Lituanie	.pm	Saint-Pierre et Miquelon	.tv	Tuvalu
.lu	Luxembourg	.pn	Pitcairn	.tw	Taïwan
.lv	Lettonie	.pr	Porto Rico (USA)	.tz	Tanzanie
.ly	Libye	.ps	Territoires palestiniens	.ua	Ukraine
.ma	Maroc	.pt	Portugal	.ug	Ouganda
.mc	Monaco	.py	Paraguay	.uk	Royaume-Uni
.md	Moldavie	.pw	Palau	.um	US Minor Outlying Islands
.mg	Madagascar	.qa	Qatar	.us	États-unis
.mh	Îles Marshall	.re	Réunion	.uy	Uruguay
.mk	Macédoine	.ro	Roumanie	.uz	Ouzbékistan
.ml	Mali	.ru	Fédération Russe	.va	Cité du Vatican
.mm	Myanmar	.rw	Rwanda	.vc	Saint-Vincent et Grenadines
.mn	Mongolie	.sa	Arabie Saoudite	.ve	Venezuela
.mo	Macao	.sb	Îles Salomon	.vg	Îles Vierges Britanniques
.mp	Îles Marianne du Nord	.sc	Seychelles	.vi	Îles Vierges Américaines
.mq	Martinique	.sd	Soudan	.vn	Viêt Nam
.mr	Mauritanie	.se	Suède	.vu	Vanuatu
.ms	Montserrat	.sg	Singapour	.wf	Wallis et Futuna
.mu	Île Maurice	.sh	Sainte-Hélène	.ws	Samoa de l'Ouest
.mv	Maldives	.si	Slovénie	.ye	Yémen
.mw	Malawi	.sj	Îles Svalbard et Jan Mayen	.yt	Mayotte
.mx	Mexique	.sk	République Slovaque	.yu	Yougoslavie
.my	Malaisie	.sl	Sierra Leone	.za	Afrique du Sud
.mz	Mozambique	.sm	San Marin	.zm	Zambie
.na	Namibie	.sn	Sénégal	.zr	Zaïre
.nc	Nouvelle-Calédonie	.so	Somalie	.zw	Zimbabwe
.ne	Niger	.sr	Surinam		

C BIBLIOGRAPHIE

Feneuil Bruno, *Réseaux*, Lycée Louis Rascol – Albi, 1998 ;

Gateau Guillaume, *Le minimum sur TCP/IP*, UFTI – IUFM Toulouse, 1997 ;

Alonso Stéphane, *Cours Réseaux*, TS IRIS – LEGT Louis Modeste-Leroy – Évreux, 2004 ;

Nicolas Pascal, *Cours de réseaux Master 1 informatique*, http://www.info.univ-angers.fr/pub/pn/,
UFR Sciences de l'Université d'Angers, 2006 ;

Péan Bruno, *Support de cours Réseaux EISTI*, EISTI – Cergy, 2001 ;

Aoun André, *Réseaux informatiques*, http://www.httr.ups-tlse.fr/pedagogie/cours/,
Université Paul Sabatier – Toulouse III, 2005 ;

Lalitte Éric dactylo. **Vigneau François-Régis**, *Cours réseau*, http://www.lalitte.com/faqs.html,
InTech INFO – Institut privé des nouvelles technologies de l'information – ESIEA, 2005 ;

Riveill Michel, *Réseaux partie 3 – réseau*, INP Grenoble – Laboratoire SIRAC – INRIA Rhône-Alpes, 1999 ;

Cousin Bernard, *Réseaux – généralités*,
http://www.irisa.fr/armor/lesmembres/cousin/Enseignement/enseignement.html,
IRISA – Campus de Beaulieu – Rennes, 2002 ;

Boniface P., Mabriez D., *Mémotech – Micro-informatique et réseaux – Bac Pro MRIM*, Casteilla, 2006 ;

Bulfone Christian, *La pile TCP/IP* et *Administration système et réseau*,
http://www.icp.inpg.fr/~bulfone/enseignement.php, Master IC2A/DCISS – INP Grenoble, 2007 ;

Laissus François, *Cours d'introduction à TCP/IP*, http://www.laissus.fr, Master SIO – École Centrale Paris, 2007 ;

Hart Robert trad. **Caillat-Vallet Laurent**, *Petit guide des sous-réseaux IP*,
http://www.ibiblio.org/pub/linux/docs/HOWTO/translations/fr/html-1page/, 1997 ;

Kanawati Rushed, *Techniques de détection & correction des erreurs de transmission*,
http://www-gtr.iutv.univ-paris13.fr/Cours/Mat/Reseaux1, IUT Villetaneuse, 2002 ;

Hardware.fr, *Forum Windows – software & réseaux*,
http://forum.hardware.fr/hfr/WindowsSoftwareReseaux/Reseaux/liste_sujet-1.htm, 2007 ;

Fontaine Sébastien « _SebF », *Les modèles* et *Les entêtes IP*, http://www.frameip.com/, 2006 ;

YBET Informatique, *Formation matériel informatique 2*, http://www.ybet.be/hardware/hardware2.htm/, 2005 ;

Desgeorge Guillaume, Degouet Sébastien, Corbel Steve, *Synthèse de protocoles courants*, ESEO, 1999 ;

Tanenbaum Andrew, *Réseaux – cours et exercices 3è édition*, Dunod, 1996 ;

Dekhinet Abdelkamel, *Communication de données et réseau*, Université de Batna, 2006 ;

CommentÇaMarche.net, http://www.commentcamarche.net/, 2007 ;

Wikipédia – l'encyclopédie libre, http://fr.wikipedia.org/, 2007 ;

Frissard Jean-Louis, *RNIS*, http://www.wellx.com/fr/prest/rnis.htm, WellX Telecom, 2007 ;

F6SS – site de la station radioamateur, *Codage NRZ, NRZI et Manchester*, http://f6css.free.fr/, 2006.