République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

UNIVERSITE MUSTAPHA STAMBOULI DE MASCARA

FACULTÉ DES SCIENCES EXACTES

Polycopié de Cours

Réseaux de Communication

Présenté par **:**

SMAIL Omar

Ce cours est destiné aux étudiants de 2eme Année Licence-LMD
Informatique- Semestre 04

Algérie
2016

Avant-propos

Nous présentons dans ce document un support de cours portant sur le module Réseaux de Communication, destiné à la filière deuxième année LMD Informatique, pour le semestre quatre.

Notre but est d'apporter des connaissances de base sur les réseaux de manière à mieux comprendre les réseaux d'entreprise, leurs fonctionnements et leurs équipements de base.

Nous traitons les transmissions de données et des réseaux en général sous l'angle de l'architecture des systèmes ouverts (modèle OSI). Une grande partie est réservée à la description du modèle TCP/IP qui est le modèle de base d'internet.

Enfin, chaque chapitre est clôturé par une série d'exercices afin de mieux assimiler les notions présentées dans le chapitre.

Table de Matières

Chapitre 5 : Les couches hautes : Session, Présentation et Application

Chapitre 6 : Le modèle TCP/IP

Liste des graphes

Liste des tableaux

Introduction aux réseaux

Les réseaux sont nés du besoin de transporter des données d'un ordinateur à un autre ordinateur. Ces données étant mises sous la forme de fichiers, l'application de base des réseaux est donc le transfert de fichiers. Un peu plus tard, le «transactionnel» est apparu pour permettre à un utilisateur de réaliser des transactions avec un ordinateur distant, par exemple réserver une place d'avion. La session correspond à l'ensemble des transactions d'un même utilisateur pour réaliser une tâche donnée.

Avec l'apparition du Web, le service transactionnel s'est diversifié afin de permettre la recherche d'information par le biais de liens. Ces applications s'appellent client/serveur, c'est-à -dire qu'un client s'adresse à un serveur pour obtenir de l'information.

L'étape suivante des réseaux a été caractérisée par le pair-à -pair, ou P2P (peer-to-peer). Dans cette application, tous les éléments connectés au réseau sont équivalents et peuvent être distribués dans le réseau. Les applications pair-à-pair sont bien connues, en particulier de ceux qui recherchent des fichiers audio ou vidéo sur Internet. Les applications sous-jacentes sont en fait très nombreuses, allant de la téléphonie à la recherche d'informations diverses et variées.

Sans que cela supprime les applications de transfert de fichiers, qu'elles soient client-serveur ou pair à pair, le nouveau service Internet en vogue au début des années 2010 est le Cloud, ou « nuage ». Jusqu'à l'arrivée des Clouds, le réseau Internet avait pour objectif de transporter des données pour réaliser un service à distance. Les entreprises permettaient aux itinérants de se connecter à leurs serveurs par le biais d'lnternet. Elles possédaient tous les serveurs nécessaires à cela, comme la messagerie électronique, les applications métier ou les serveurs d'archivage, ainsi que la puissance de calcul nécessaire. Aujourd'hui, il est possible de réaliser dans le Cloud ce qui se faisait auparavant dans l'entreprise : calcul, stockage, application métier, messagerie, téléphonie, etc. Les avantages sont nombreux : le client peut accéder a ces services de n'importe où; ils peuvent être sécurisés par de la redondance ; on peut ajouter instantanément de nouveaux services, de la puissance de calcul, de l'espace de stockage, etc., au fur et à mesure des besoins, en ne payant que ce qui est utilisé.

Cette nouvelle génération met en oeuvre le concept de virtualisation, par lequel les ressources dont l'entreprise ou le particulier à besoin peuvent se trouver n'importe ou, voire se déplacer en fonction du coût des serveurs.

L'architecture d'un réseau est définie principalement par la façon dont les paquets sont transmis d'une extrémité à une autre du réseau. De nombreuses possibilités existent pour cela, comme celles consistant à faire passer les paquets toujours par la même route ou, au contraire, à les faire transiter par des routes distinctes de façon à minimiser les délais de traversée.

Le modèle de référence

Pour identifier correctement toutes les composantes nécessaires à la bonne marche d'un réseau à transfert de paquets, un modèle de référence a été mis au point. Ce modèle définit une partition de l'architecture en sept niveaux, connu par le modèle de référence à sept couches ou le modèle OSI (Open Systems Interconnection). Ce modèle prend en charge l'ensemble des fonctions nécessaires au transport et à la gestion des paquets, comme l'illustre la figure 1. Ces sept couches de protocoles ne sont pas toutes indispensables, notamment aux réseaux sans visée généraliste.

Chaque niveau, ou couche, offre un service au niveau supérieur et utilise les services du niveau inférieur. Pour offrir ces services, les couches disposent de protocoles, qui appliquent les algorithmes nécessaires à la bonne marche des opérations.

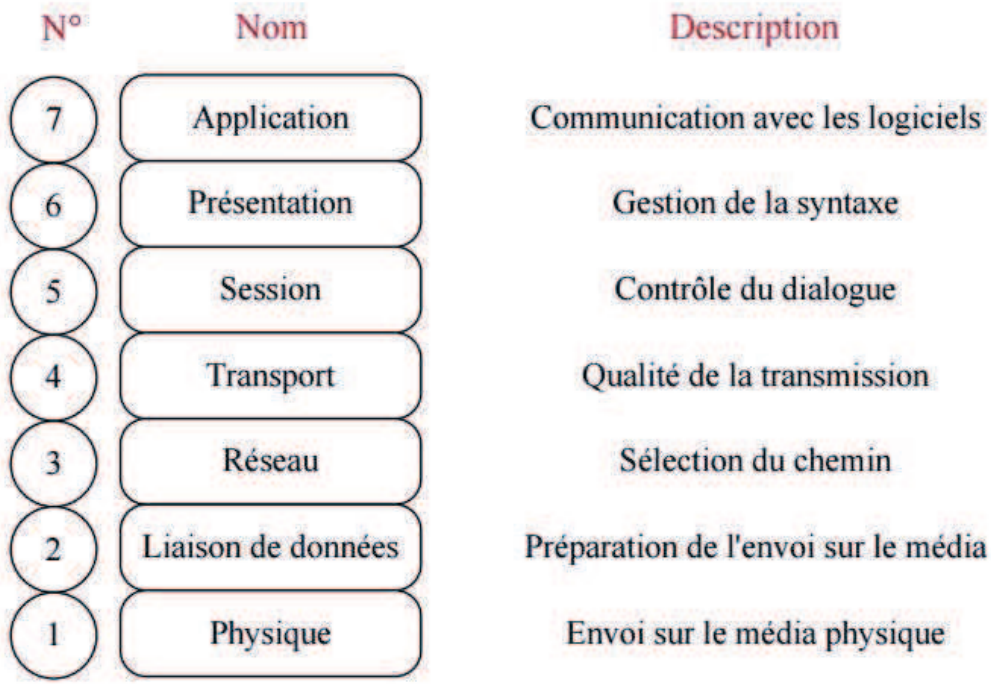


Figure 1- Les 7 couches du modèle OSI

Nous supposons ici que l'architecture protocolaire est découpée en sept niveaux, ce qui est le cas du modèle de référence. La couche 3, ou couche réseau, représente le niveau paquet, qui définit les algorithmes nécessaires pour que les entités de cette couche, les paquets, soient acheminées correctement de l'émetteur au récepteur. La couche 7 correspond au niveau application. Le rôle du protocole de la couche 7 est de transporter correctement l'entité de niveau application, le message utilisateur, de l'équipement émetteur à l'équipement récepteur. La couche 2, ou couche liaison de données, représente le niveau trame. Elle permet

de transférer le paquet sur une ligne physique. En effet, un paquet ne contenant pas de délimiteur, le récepteur ne peut en déterminer 1a fin ni identifier le commencement du paquet suivant. Pour transporter un paquet, il faut donc le mettre dans une trame, qui, elle, comporte des délimiteurs. On peut aussi encapsuler un paquet dans un autre paquet, lui-même encapsulé dans une trame.

Dans cet ouvrage, nous distinguons les mots « paquet» et « trame» de façon à bien différencier les entités qui ne sont pas transportables directement, comme le paquet IP, et les entités transportables directement par la couche physique, comme les trames Ethernet. La structure en couches de l'architecture protocolaire des réseaux simplifie considérablement leur compréhension globale et facilite leur mise en œuvre. Il est possible de remplacer une couche par une autre de même niveau sans avoir à toucher aux autres couches. On peut, par exemple, remplacer la couche 3 par une couche 3 prime (3') sans modifier les couches 1, 2, 4, 5, 6 ou 7. On ne modifie de la sorte qu'une partie de l'architecture, 1a couche 3, sans toucher au reste. Les interfaces entre les couches doivent être respectées pour réaliser ces substitutions: l'interface de la couche 3' avec les couches 2 et 4 doit garantir que les couches 2 et 4 n'ont pas à être modifiées.

L'architecture illustrée dans la figure 1, sert de référence à toutes les architectures réseau, d'où son nom de modèle de référence. Une autre architecture protocolaire, l'architecture TCP/IP (Transmission Control Protocol/lnternet Protocol), a été définie un peu avant le modèle de référence par le ministère américain de la Défense. Son rôle premier était d'uniformiser l'interface extérieure des différents réseaux utilisés par le département d'état américain de façon à les interconnecter facilement. C'est cette architecture TCP/IP qui a été adoptée pour le réseau Internet, ce qui lui a offert une diffusion massive.

Une autre architecture provenant de l'utilisation de la trame plutôt que du paquet a été proposée par l'UIT-T (Union internationale des télécommunications standardisation du secteur télécommunications), pour les applications utilisant à la fois les données, la téléphonie et l'image. Provenant principalement du monde des télécommunications, cette architecture est bien adaptée au transport de flux continus, comme la parole téléphonique. C'est la trame ATM (Asynchronous Transfer Mode) qui représente le mieux cette architecture. Cependant, cette architecture est en forte perte de vitesse, la trame ATM étant progressivement remplacée par la trame Ethernet.

La couche physique.

1.1 Définition.

La couche physique fournit les moyens mécaniques, électriques, fonctionnels et procéduraux nécessaires à l'activation, au maintien et à la désactivation des connexions physiques destinées à la transmission de bits entre deux entités de liaison de données.

Ici, on s'occupe donc de transmission des bits de façon brute, l'important est que l'on soit sûr que si l'émetteur envoie un bit à 1 alors le récepteur reçoit un bit à 1. Les normes et standards de la couche physique définissent le type de signaux émis (modulation, puissance, portée...), la nature et les caractéristiques des supports (câble, fibre optique...), les sens de transmission... Tout d'abord une liaison entre 2 équipements A et B peut être simplex (unidirectionnelle), dans ce cas A est toujours l'émetteur et B le récepteur. C'est ce que l'on trouve par exemple entre un banc de mesure et un ordinateur recueillant les données mesurées. La communication est half-duplex (bidirectionnelle à l'alternat) quand le rôle de A et B peut changer, la communication change de sens à tour de rôle (comme avec des talkies-walkies). Elle est full-duplex (bidirectionnelle simultanée) quand A et B peuvent émettre et recevoir en même temps (comme dans le cas du téléphone).
La transmission de plusieurs bits peut s'effectuer en série ou en parallèle. En série, les bits sont envoyés les uns derrière les autres de manière synchrone ou asynchrone. Dans le mode synchrone l'émetteur et le récepteur se mettent d'accord sur une base de temps (un top d'horloge) qui se répète régulièrement durant tout l'échange. À chaque top d'horloge (ou k tops d'horloge k entier fixé définitivement) un bit est envoyé et le récepteur saura ainsi quand lui arrive les bits. Dans le mode asynchrone, il n'y a pas de négociation préalable mais chaque caractère envoyé est précédé d'un bit de start et immédiatement suivi d'un bit de stop. Ces deux bits spéciaux servent à caler l'horloge du récepteur pour qu'il échantillonne le signal qu'il reçoit afin d'y décoder les bits qu'il transmet. En parallèle, les bits d'un même caractère sont envoyés en même temps chacun sur un fil distinct, mais cela pose des problèmes de synchronisation et n'est utilisé que sur de courtes distances (bus par exemple).
Quel que soit le mode de transmission retenu, l'émission est toujours cadencée par une horloge dont la vitesse donne le débit de la ligne en bauds, c'est-à-dire le nombre de tops d'horloge en une seconde. Ainsi, une ligne d'un débit de 100 bauds autorise 100 émissions par seconde. Si à chaque top d'horloge un signal représentant 0 ou 1 est émis, alors dans ce cas le débit en bit/s est équivalent au débit en baud. Cependant, on peut imaginer que le signal émis puisse prendre 4 valeurs distinctes (0, 1, 2, 3) dans ce cas le signal a une valence de 2 et le débit en bit/s est double de celui en baud. D'une manière générale, si le signal peut prendre 2n valeurs distinctes on dit alors que sa valence est de n, ainsi à chaque top n bits peuvent être transmis simultanément et si le débit de la ligne est de x bauds il est en fait de n.x bit/s.

1.2 Transmission en bande de base.

La transmission en bande de base consiste à envoyer directement les suites de bits sur le support à l'aide de signaux carrés constitués par un courant électrique pouvant prendre 2 valeurs (5 Volts ou 0 par exemple). On détaillera ci-après les différents codages des bits possibles, mais dans tous les cas l'émetteur envoie sur la ligne un signal carré du type de celui de la figure 1.1 pour la séquence de bits 1010 par exemple.

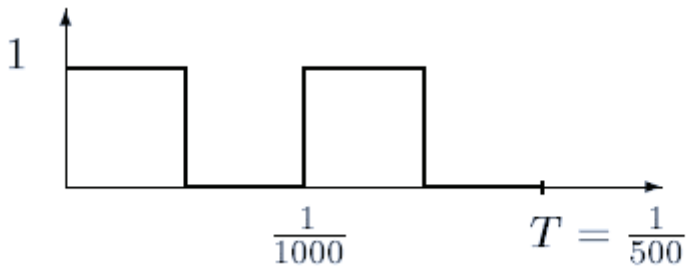


Figure 1.1 - Signal carré de la séquence de bits 1010.

En considérant ce signal g(t) comme périodique (il suffit de répéter une fois sur [T..2T] le signal donné sur [0..T] pour obtenir un signal périodique sur [0..2T]) on peut le décomposer en une série de Fourier de la forme :

$$g(t) = c/2 + \sum_{n=1}^{\infty} a_n \sin(2\Pi nft) + \sum_{n=1}^{\infty} b_n \cos(2\Pi nft)$$

Où

$f = 1/T$ La fréquence fondamentale du signal

$$c = 2/T \int_0^T g(t)\,dt$$

$$a_n = 2/T \int_0^T g(t) \sin(2\Pi nft)\,dt$$

$$b_n = 2/T \int_0^T g(t) \cos(2\Pi nft)\,dt$$

On dit que le signal carré est décomposé en une somme infinie d'harmoniques, la première étant dénommée fondamentale, et cette approximation mathématique permet de savoir quel signal électrique sera réellement reçu au bout du câble.
Cependant, le câble sur lequel est émis le signal possède une bande passante qui est l'intervalle des fréquences possibles sur ce support, donc à la réception on ne retrouve pas toute la richesse du signal initial et dans la plupart des cas le signal carré sera très déformé. Par exemple, le câble téléphonique a une bande passante de 300 à 3400 Hz, donc tous les signaux de fréquence inférieure à 300 ou supérieure à 3400 seront éliminés.
Dans notre exemple nous obtenons

$$g(t) = 1/2 + 2/\Pi \sin(2000\,\Pi\,t) + 2/2\Pi \sin(6000\,\Pi\,t) + 2/5\Pi \sin(10000\,\Pi\,t) + ...$$

Dans la figure 1.2 on remarque que plus la fréquence augmente plus l'amplitude diminue. À droite nous avons le signal réellement perçu par le récepteur si l'on considère que le câble ne laisse passer que ces 3 harmoniques-ci. Dans ce cas le signal reçu reste assez proche du carré émis et le récepteur n'aura pas trop de mal à le décoder.

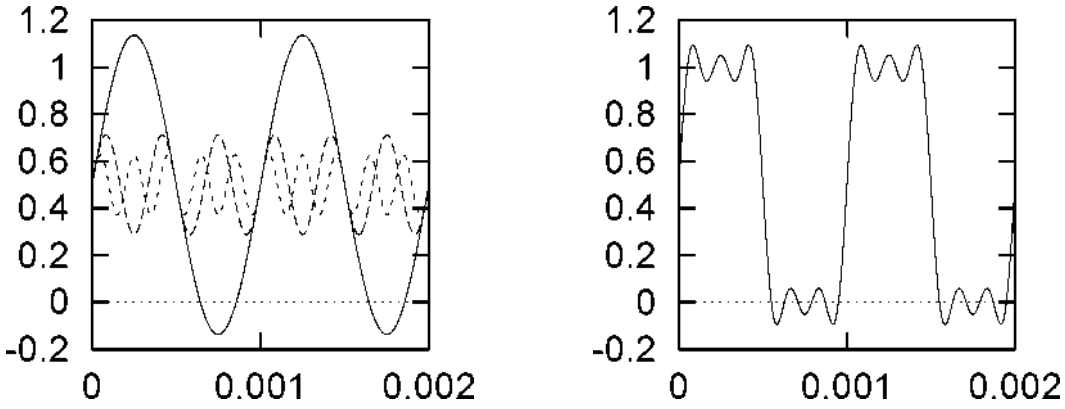


Figure 1.2- Harmoniques et transformée de Fourrier de la séquence de bits 1010.

Sans entrer dans des détails relevant de la théorie du signal, nous indiquerons simplement que sur une ligne téléphonique dont la bande passante est de 3100Hz et pour un rapport signal/bruit de l0dB on peut atteindre une capacité de l0Kbits/s.

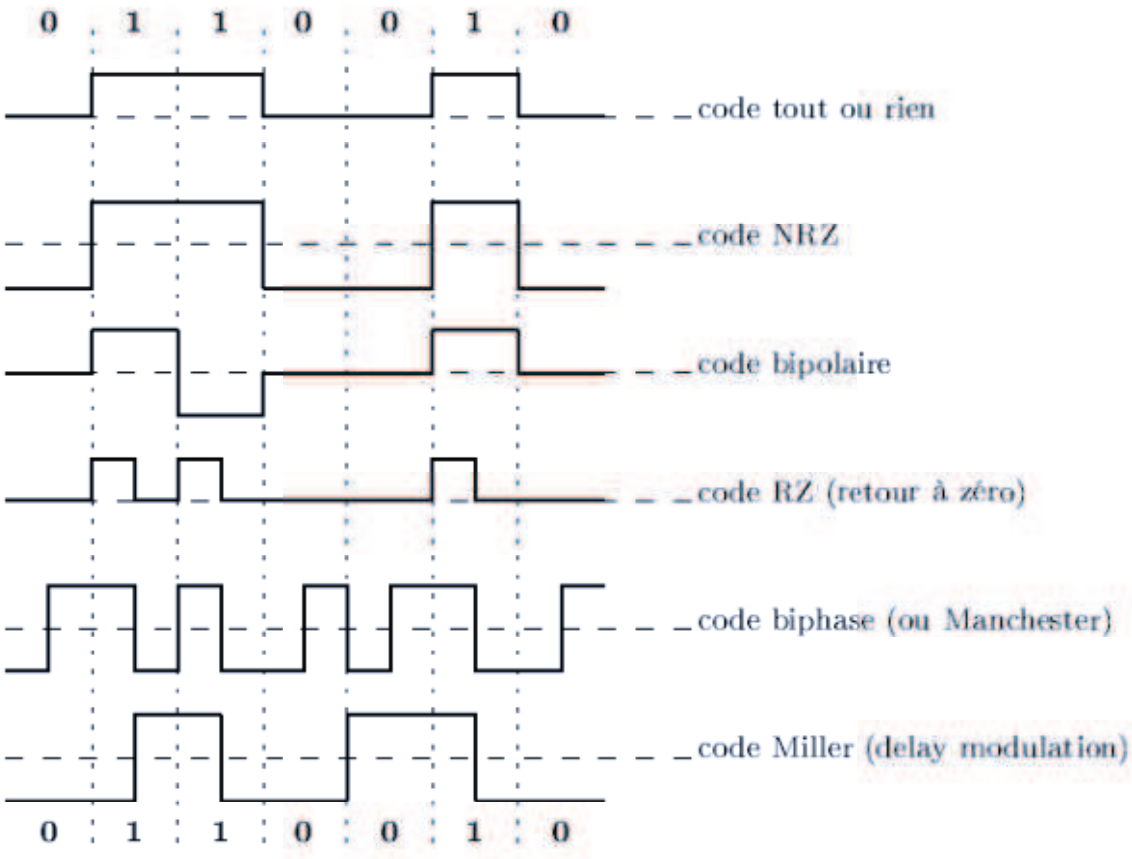


Figure 1.3 - Différents codages en bande de base de la séquence 0110010.

Dans la figure 1.3 nous trouvons quelques exemples de codage de l'information pour une transmission en bande de base.

- le code tout ou rien : c'est le plus simple, un courant nul code le 0 et un courant positif indique le 1

- le code NRZ (non retour à zéro): pour éviter la difficulté à obtenir un courant nul, on code le 1 par un courant positif et le 0 par un courant négatif.
- le code bipolaire : c'est aussi un code tout ou rien dans lequel le 0 est représenté par un courant nul, mais ici le 1 est représenté par un courant alternativement positif ou négatif pour éviter de maintenir des courants continus.
- le code RZ : le 0 est codé par un courant nul et le 1 par un courant positif qui est annulé au milieu de l'intervalle de temps prévu pour la transmission d'un bit.
- le code Manchester : ici aussi le signal change au milieu de l'intervalle de temps associé à chaque bit. Pour coder un 0 le courant sera négatif sur la première moitié de l'intervalle et positif sur la deuxième moitié, pour coder un 1, c'est l'inverse. Autrement dit, au milieu de l'intervalle il y a une transition de bas en haut pour un 0 et de haut en bas pour un 1.
- le code Miller : on diminue le nombre de transitions en effectuant une transition (de haut en bas ou l'inverse) au milieu de l'intervalle pour coder un 1 et en n'effectuant pas de transition pour un 0 suivi d'un 1. Une transition est effectuée en fin d'intervalle pour un 0 suivi d'un autre 0.

1.3 Transmission modulée.

Le principal problème de la transmission en bande de base est la dégradation du signal très rapide en fonction de la distance parcourue, c'est pourquoi elle n'est utilisée qu'en réseau local (<5km). Il serait en effet trop coûteux de prévoir des répéteurs pour régénérer régulièrement le signal. C'est pourquoi sur les longues distances on émet un signal sinusoïdal qui, même s'il est affaibli, sera facilement décodable par le récepteur. Ce signal sinusoïdal est obtenu grâce à un modem (modulateur-démodulateur) qui est un équipement électronique capable de prendre en entrée un signal en bande de base pour en faire un signal sinusoïdal (modulation) et l'inverse à savoir restituer un signal carré à partir d'un signal sinusoïdal (démodulation). Autrement dit il permet de passer de signaux numériques discrets (0 ou 1) à des signaux analogiques continus.
Il existe trois types de modulation décrits dans la figure 1.4

- la modulation d'amplitude envoie un signal d'amplitude différente suivant qu'il faut transmettre un 0 ou un 1. Cette technique est efficace si la bande passante et la fréquence sont bien ajustées. Par contre, il existe des possibilités de perturbation (orage, lignes électriques...), car si un signal de grande amplitude (représentant un 1) est momentanément affaibli le récepteur l'interprétera à tort en un 0.
- la modulation de fréquence envoie un signal de fréquence plus élevée pour transmettre un 1. Comme l'amplitude importe peu, c'est un signal très résistant aux perturbations (la radio FM est de meilleure qualité que la radio AM) et c'est assez facile à détecter.

- la modulation de phase change la phase du signal (ici de 180°) suivant qu'il s'agit d'un 0 (phase montante) ou d'un 1 (phase descendante).

Dans les exemples donnés ci-dessus on a seulement 2 niveaux possibles à chaque fois, donc on a uniquement la possibilité de coder 2 valeurs différentes à chaque instant, dans ce cas 1 baud = lbit/s. De manière plus sophistiquée il existe des modems capables de moduler un signal suivant plusieurs niveaux, par exemple 4 fréquences différentes que le modem récepteur saura lui aussi distinguer. Dans ce cas, chaque signal envoyé code 2 bits donc 1 baud = 2bit/s. Il est même possible de transmettre des signaux mêlant les différentes modulations présentées comme dans le cas de la norme V29 qui module à la fois l'amplitude du signal sur 2 niveaux et la phase sur 8 niveaux (0°,45°,... ,315°). En combinant les 2 modulations, on obtient ainsi 16 signaux différents possibles à chaque instant, permettant de transmettre simultanément 4 bits à chaque top d'horloge (1 baud = 4 bit/s).

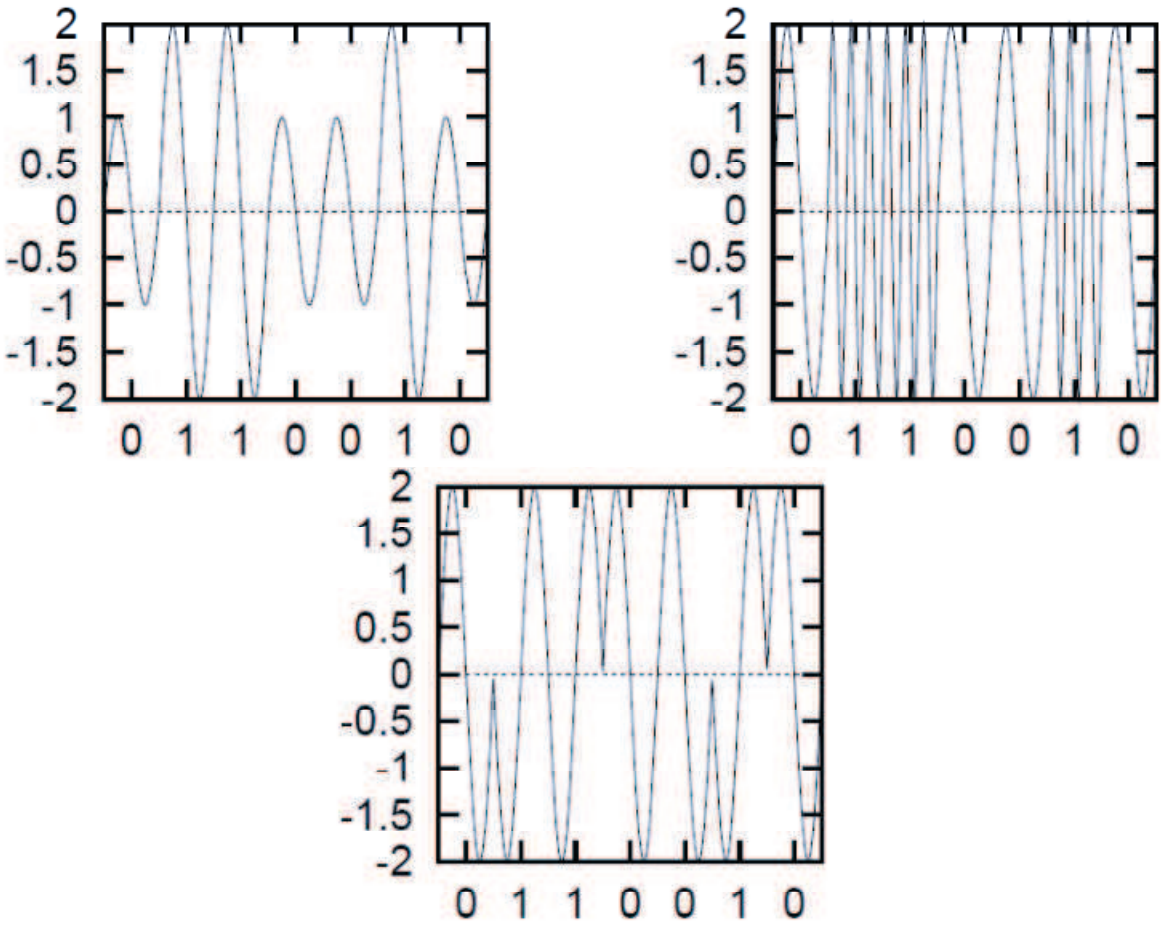


Figure 1.4 - Modulations d'amplitude, de fréquence et de phase de la séquence de bits 0110010.

1.4 Multiplexage.

Le multiplexage consiste à faire transiter sur une seule et même ligne de liaison, dite voie haute vitesse, des communications appartenant à plusieurs paires d'équipements émetteurs et récepteurs comme représenté dans la figure 1.5. Chaque émetteur (resp. récepteur) est raccordé à un multiplexeur (resp. démultiplexeur) par une liaison dit voie basse vitesse.

Plusieurs techniques sont possibles :
- le multiplexage fréquentiel consiste à affecter à chaque voie basse vitesse une bande passante particulière sur la voie haute vitesse en s'assurant qu'aucune bande passante de voie basse vitesse ne se chevauche. Le multiplexeur prend chaque signal de voie basse vitesse et le réémet sur la voie haute vitesse dans la plage de fréquences prévues. Ainsi plusieurs transmissions peuvent être faites simultanément, chacune sur une bande de fréquences particulières, et à l'arrivée le démultiplexeur est capable de discriminer chaque signal de la voie haute vitesse pour l'aiguiller sur la bonne voie basse vitesse.
- le multiplexage temporel partage dans le temps l'utilisation de la voie haute vitesse en l'attribuant successivement aux différentes voies basse vitesse même si celles-ci n'ont rien à émettre. Suivant les techniques chaque intervalle de temps attribué à une voie lui permettra de transmettre 1 ou plusieurs bits.
- le multiplexage statistique améliore le multiplexage temporel en n'attribuant la voie haute vitesse qu'aux voies basse vitesse qui ont effectivement quelque chose à transmettre. En ne transmettant pas les silences des voies basses cette technique implantée dans des concentrateurs améliore grandement le débit global des transmissions mais elle fait appel à des protocoles de plus haut niveau et est basée sur des moyennes statistiques des débits de chaque ligne basse vitesse.

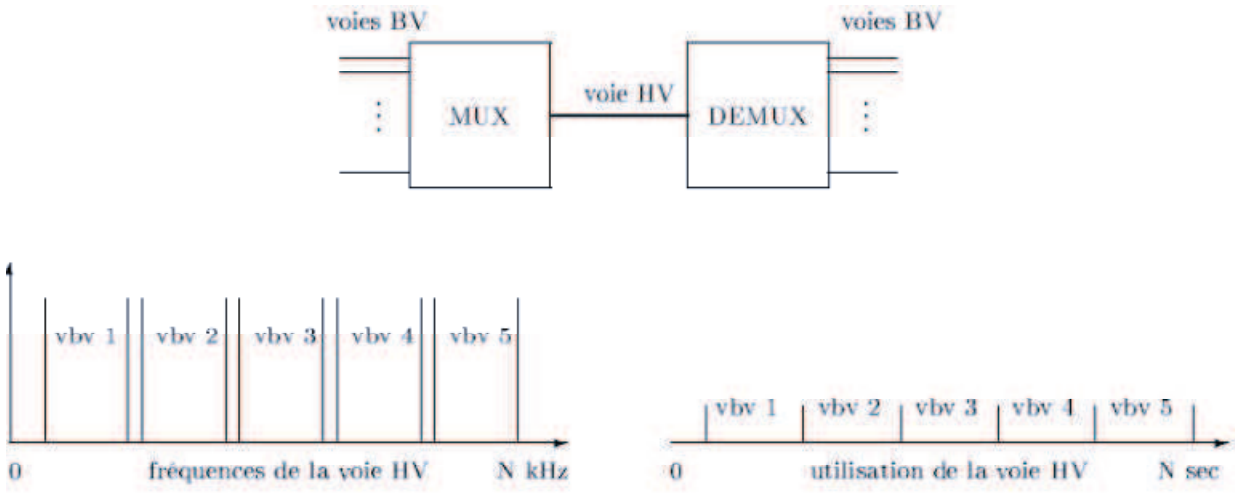


Figure 1.5 - Multiplexages d'une ligne.

1.5 Les supports de transmission.

L'objectif de la couche 1 du modèle OSI est aussi de fixer les caractéristiques des matériels utilisés pour relier physiquement les équipements d'un réseau. Nous décrivons succinctement quelques uns des supports de transmission les plus utilisés :

1.5.1 Les cables coaxiaux.

Les câbles électriques (cuivre) blindés **coaxiaux** qui ressemblent aux câbles TV. Malgré de bonnes qualités intrinsèques (faible sensibilité aux perturbations électromagnétiques), ils sont de moins en moins utilisés (voir la figure 1.6).

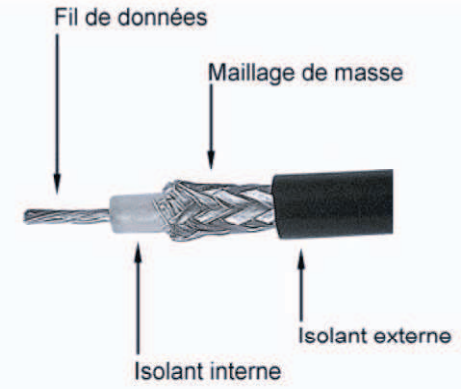


Figure 1.6 - Câble coaxial.

Le principe est de faire circuler le signal électrique dans le fil de données central. On se sert du maillage de masse, autrement appelé *grille*, pour avoir un signal de référence à 0 V. On obtient le signal électrique en faisant la *différence de potentiel* entre le fil de données et la masse.
Le nom scientifique donné au câble coaxial est donc le 10B2 ou 10B5 pour sa version encore plus ancienne.

le 10 indique le débit en Mbps (mégabits par seconde) ;
le B indique la façon de coder les 0 et les 1, soit ici la bande de **B**ase ;
le dernier chiffre indique la taille maximale du réseau, exprimée en mètres.

Cette taille est de 200 m pour le 10B2, et 500 m pour le 10B5. Par exemple, pour une longueur de 200 m, si je divise par 100, cela donne 2. Le nom scientifique est donc bien 10B2.

1.5.1.1 Le câble coaxial 10B5.

Le 10B5 est le plus ancien et le plus dur à utiliser. Le principe est de poser le câble partout dans les salles à informatiser. Ensuite, on peut brancher des machines sur le câble, mais seulement à certains endroits, la connexion se fait à l'aide de prises vampire.

En fait, il fallait faire un petit trou, à la main, dans le câble, pour atteindre le fil de données. Une fois cette manipulation effectuée, on mettait en place la prise vampire dans laquelle une petite pointe en métal venait en contact avec le fil de données et permettait de récupérer le signal (voir la figure 1.7).

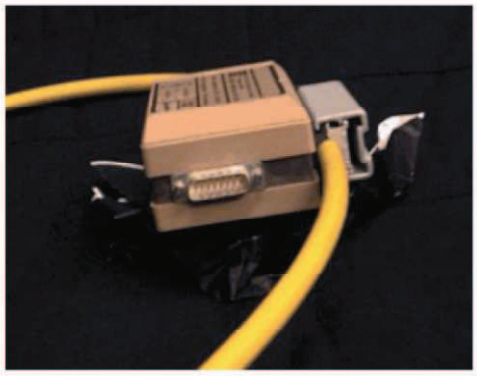


Figure 1.7 - Prise Vanpire.

Les câbles 10B5 étant très épais, il était difficile de les plier.

1.5.1.2 Le câble coaxial 10B2.

Le câble coaxial 10B2 possède la même structure que le 10B5, mais en plus fin. La connectique utilisée est aussi très différente, car la propagation de l'information ne se fait pas de la même façon.

1.5.1.3 Prise, T et bouchon BNC.

Pour mettre en place un réseau en 10B2, il fallait : des câbles 10B2 équipés de prises BNC ; des tés BNC ; des bouchons.
Voici aux figures suivantes, dans l'ordre, le câble équipé d'une prise BNC, le té BNC et le bouchon BNC.

Figure 1.8 - Prise BNC **Té BNC** **Bouchon BNC.**

Pour créer le réseau, on met un bouchon sur un côté du té, une carte réseau sur le deuxième côté (celui du milieu) et un câble sur la dernière prise. L'autre extrémité du câble était branchée sur un autre té, et ainsi de suite jusqu'à la fermeture du réseau par un bouchon.
Voici à la figure 1.9 suivante un exemple de connexion sur un té.

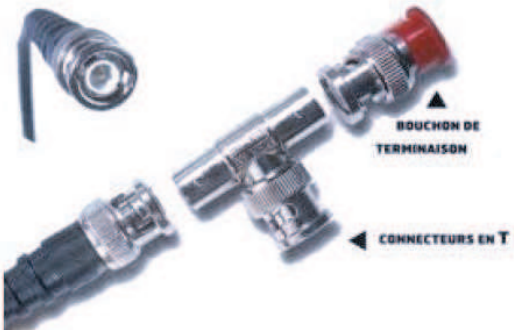


Figure 1.9 - Connexion BNC.

Le réseau utilisant le câble coaxial et les connecteurs BNC devient : (voir la figure 1.10)

Figure 1.10 - Réseau BNC.

1.5.2 Les paires torsadées.

Les câbles électriques (cuivre) à **paires torsadées**, qui ressemblent aux câbles téléphoniques. Les torsades diminuent la sensibilité aux perturbations électromagnétiques, la diaphonie (mélange de signaux entre paires) et l'atténuation du signal tout au long du câble. Il existe des versions **blindées (STP** Shielded Twisted Pair**)** et **non blindées (UTP** Unshielded Twisted Pair**)**. Les câbles à paires torsadées sont actuellement les plus employés (voir la figure 1.11).

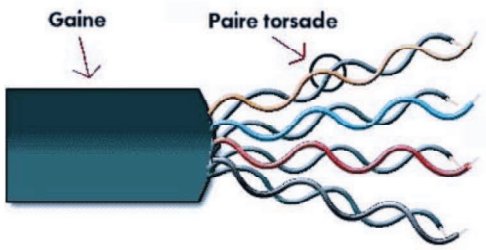


Figure 1.11 - Paires torsadées.

On utilise 2 paires, soit 4 fils, car nous utilisons une paire pour envoyer les données, et une paire pour les recevoir. Nous n'utilisons donc que 4 fils sur 8. Il existe déjà des technologies qui utilisent plus de 4 fils.
En effet, on s'est rendu compte qu'en torsadant les fils de la sorte, le câble était moins sujet à des perturbations électromagnétiques.
Les paires torsadées sont appelés aussi le 10BT, ou 100BT ou 1000BT, selon le débit utilisé (10 Mbps, 100 Mbps, 1000 Mbps) le T étant là pour «torsadé », ou *twisted* en anglais. On ajoute parfois un x derrière, pour dire que le réseau est commuté.

La prise RJ45 est utilisée dans ce type de câble. On peut voir les 8 petits connecteurs en cuivre qui sont reliés aux 8 fils, voir la figure 1.12.

Figure 1.12 - Prise RJ45.

Il faut utiliser des fils spécifiques, qui sont les fils 1, 2, 3 et 6. Voici en figure suivante le branchement d'un câble et les fils utilisés (avec les couleurs), appelé aussi câble droit (figure 1.13).

Figure 1.13 – Couleurs câble paires torsadées droit.

De plus, cette prise doit être branchée dans une autre prise pour être connectée. On appelle cette prise une prise femelle, elle est généralement située sur un hub ou un switch.
Pour pouvoir relier la transmission de la machine A avec la réception de la machine B, il faudrait que les fils 1 et 2 soient en relation avec les fils 3 et 6... Ce qui reviendrait à *croiser* les fils
Comme vous pouvez le constater sur la figure suivante (figure 1.14), nous avons bien la transmission de la machine A en relation avec la réception de la machine B, utilisant un câble croisé.

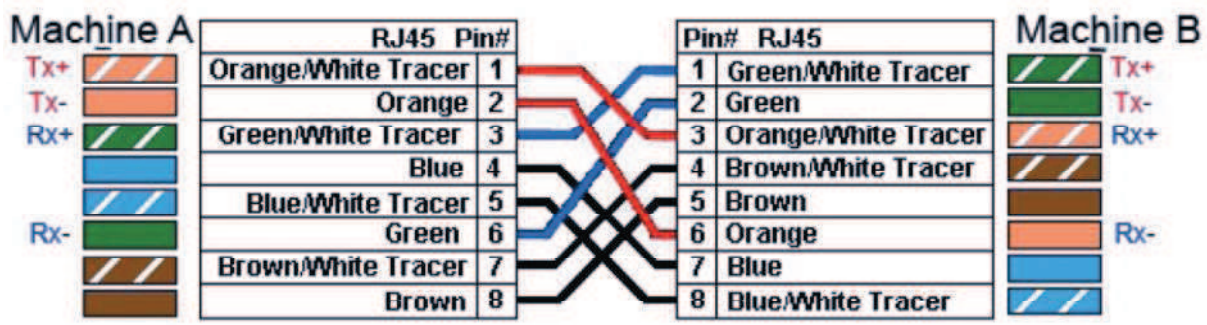


Figure 1.14 - Couleurs câble paires torsadées croisé.

A priori, même si cela coûte encore très cher, **la fibre optique** est amenée à remplacer la paire torsadée, notamment en raison des débits qu'elle peut offrir.

1.5.3 La fibre optique.

Les câbles à **fibres optiques** (voir la figure 1.15) qui transmettent les informations par modulation d'un faisceau lumineux. Ils ont composé d'une fibre d'émission et une fibre de réception. Les câbles à fibres optiques ont de nombreux avantages :

- Ils sont extrêmement rapides (bande passante élevée).
- Ils sont insensibles à toute perturbation électromagnétique et n'en génèrent pas eux-mêmes.
- Ils génèrent très peu d'atténuation sur le signal lumineux, ce qui permet d'utiliser un segment unique de très grande longueur.
- Ils sont très peu encombrants et nettement plus légers que les câbles en cuivre
- Ils assurent une meilleure confidentialité des données (difficulté de réaliser une connexion pirate).

Cependant, en raison de leur coût global élevé (adaptateur, câble, installation, réglages délicats…), leur utilisation dans les réseaux locaux est plutôt réservée aux épines dorsales

(backbones), c'est à dire aux arrivées centrales d'immeubles ou encore lorsqu'une bande passante considérable est indispensable (multimédia, visiophonie, transmission de gros fichiers…).
Le nom scientifique de la fibre est communément le 1000BF. Du gigabit avec le F pour Fibre
Il existe aujourd'hui globalement deux types de fibre :

- la fibre monomode ;
- la fibre multimode.

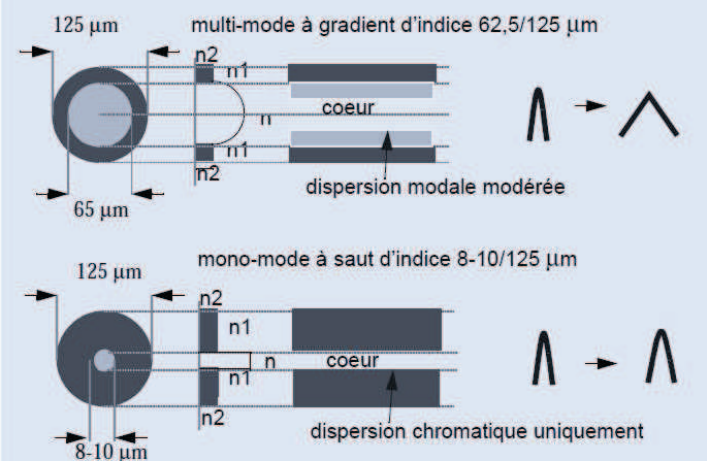


Figure 1.15 – Fibre monomode & multimode.

La fibre monomode fait passer **une seule longueur d'onde** lumineuse, soit une seule couleur. Elle fonctionne donc avec du laser qui peut être vert, bleu, rouge, etc.
La fibre multimode fonctionne avec de la lumière blanche, et donc **toutes les longueurs d'ondes** (la lumière blanche est la somme de toutes les lumières possibles, comme celle du soleil).
On pourra ainsi parcourir une plus longue distance avec de la fibre monomode. En gros :

- 2 km pour la fibre multimode ;
- 60 km pour la fibre monomode.

Même si les distances parcourues aujourd'hui peuvent être beaucoup plus grandes (le record étant de l'ordre de 8000 km) c'est un bon ordre de grandeur.
C'est ainsi que l'on a relié les États-Unis et l'Europe, en passant de la fibre monomode dans l'Atlantique, et en répétant le signal lumineux tous les 60 km.

Exercices du Chapitre 1 : Couche Physique

Exercice 1.1 :

Soit à transmettre en bande de base l'information binaire : 11100110,

_ Donner le chronogramme du signal numérique en utilisant les encodages cités en cours ?

Exercice 1.2 :

Donner le signal modulé de l'information binaire de l'exercice 1, en utilisant les trois modulations citées en cours?

Exercice 1.3 :

Soit le signal numérique : 0111010111 on suppose que le signal est transmis en 01 seconde :

_ Donner le chronogramme du signal en bande de base en choisissant l'encodage le plus performant ? justifier ?

_ Donner la représentation du signal dans le cas d'une modulation de phase ?

Exercice 1.4 :

On veut communiquer entre deux ordinateurs reliés par une ligne physique dont la bonde passante est de 1600 HZ. Les modems utilisés ont 04 états de fréquence.

_ Calculer Le débit binaire de la ligne de transmission ?
_ Calculer la durée de transmission pour l'envoi d'un paquet contenant un seul caractère ?
_ Expliquer pourquoi on ne peut utiliser le code NRZ dans cette transmission ?
_ Est se que le changement du type de trame influe sur la durée de transmission ?
_ Dans le contexte de durée de transmission, donnez le type le plus intéressant ?

Exercice 1.5 :

Soit le signal numérique avec une parité paire d(t)=01100011. On suppose que ce signal est transmis en 01 seconde et va moduler une porteuse en fréquence selon deux cas :

- A deux états de fréquence
- A quatre états de fréquence :

_ Donner le chronogramme du signal en bande de base dans le cas d'un encodage Miller ?
_ Calculer la rapidité de modulation et le débit binaire ?
_ En déduire la relation entre le débit et la rapidité de modulation ?
_ Quel est l'intérêt de l'encodage dans la transmission numérique ?

Exercice 1.6 :

Parmi les liaisons utilisées en connexion : Les liaisons hertziennes radio. Expliquer pourquoi cette liaison n'est pas opérationnelle dans le domaine des réseaux informatique ? y a t'il un avenir pour ce type de liaisons ? sachant que les fréquences hertziennes sont limitées par la bande passante, existe t'il une relation entre le débit et la bande passante ?

La couche liaison de données.

2.1 Définition.

La couche liaison de données fournit les moyens fonctionnels et procéduraux nécessaires à l'établissement, au maintien et à la libération des connexions de liaison de données entre entités du réseau. Elle détecte et corrige, si possible, les erreurs dûes au support physique et signale à la couche réseau les erreurs irrécupérables. Elle supervise le fonctionnement de la transmission et définit la structure syntaxique des messages, la manière d'enchainer les échanges selon un protocole normalisé ou non.

Une connexion de liaison de données est réalisée à l'aide d'une ou plusieurs liaisons physiques entre deux machines adjacentes dans le réseau donc sans nœuds intermédiaires entre elles.
Nous commençons par examiner les différentes techniques de détection et correction d'erreur (changement de 1 par 0 ou vice-versa), puis nous étudierons deux familles de protocoles de liaison de données.

2.2 Détection et correction d'erreurs.

Le taux d'erreurs de transmission est de l'ordre de 10^{-5} sur une ligne téléphonique, de 10^{-7} à 10^{-8} sur un coaxial et de 10^{-10} à 10^{-12} sur une fibre optique. À ce niveau-là il ne s'agit pas d'assurer la correction globale d'un échange, mais de détecter et d'éventuellement corriger des erreurs de transmissions dans un bloc de bits acheminé par le support physique. En effet, puisque la couche 2 ne connaît pas le propriétaire des paquets qu'elle manipule, elle ne peut pas se substituer aux couches de niveau supérieur.
Les techniques employées ici reposent sur l'utilisation de codes correcteurs ou codes détecteurs d'erreurs qui chacun transforme la suite de bits à envoyer en lui ajoutant de l'information à base de bits de redondance ou bits de contrôle. Le récepteur se sert de cette information ajoutée pour déterminer si une erreur s'est produite et pour la corriger si la technique employée le permet.

- la parité ajoute à chaque bloc de i bits ($i=7$ *ou* 8) émis un bit de parité de telle sorte que parmi les $i + 1$ bits émis le nombre de bits à 1 soit toujours pair (ou impair). Par exemple, pour une parité paire si le bloc initial est de 7 bits et est égal à *1000001* le bloc de 8 bits émis est *10000010*, pour envoyer *0110100* le bloc *01101001* est émis. À la réception, le décodeur calcule le nombre de bits à 1 et dans le cas d'une parité paire si ce nombre de bits est pair on suppose qu'il n'y a pas eu d'erreur. Sinon, on sait alors qu'il y a eu une erreur de transmission mais on ne sait pas la localiser et il faut alors demander la réémission du bloc. La technique de parité est simple à mettre en œuvre cependant elle ne permet pas de détecter 2n erreurs dans le même bloc de bits transmis, car dans ce cas la parité ne sera pas changée.

- les codes à redondance cyclique (CRC) ajoutent des bits qui sont des combinaisons linéaires des bits de l'information à transmettre. La suite de bits à transmettre $u_1,u_2,\ldots u_k$ est considérée comme un polynôme $M(x) = u_1x^{k-1} + u_2x^{k-2} + \ldots + u_k$ par exemple *1101011011* est représenté par $x^9+x^8+x^6+x^4+ x^3+x+1$. À l'émission, on calcule la division du polynôme M multiplié par x^r par le polynôme générateur G de degré r. On appelle Q le polynôme quotient et R le polynôme reste de cette division, on a donc : $x^rM(x) = Q(x).G(x) + R(x)$. La suite de bits correspondant au polynôme R constitue le CRC qui est ajouté à l'information à transmettre, le polynôme total émis est donc $E(x) = x^rM(x) + R(x)$ Par exemple, à l'aide du polynôme générateur $G(x) = x^4 + x + 1$, la suite 1101011011 sera transmise accompagnée du CRC 1110 car

$x^4M(x) = x^{13} + x^{12} + x^{10} + x^8 + x^7 + x^5 + x^4$
$= (x^9 + x^8 + x^3 + x)(x^4 + x + 1) + x^3 + x^2 + x$

À la réception, on divise le polynôme M' correspondant à la suite totale de bits reçus (informa- tion+CRC) par le polynôme générateur. Si le reste calculé est non nul, c'est qu'une erreur s'est produite dans la transmission. Si le reste est nul, on est à peu près sûr (99,975% avec le polynôme générateur $x^{16} + x^{12} + x^5 + 1$ de la norme V41 du ITU-T) que la transmission s'est faite sans erreur.

Pourquoi cela fonctionne-t-il? Il est évident que $x^rM(x)$-$R(x)$ est divisible par $G(x)$, mais en arithmétique modulo 2 addition et soustraction sont équivalentes (ce sont des OU exclusifs en fait) donc on a également $E(x) = x^rM(x) + R(x) = G(x)Q(x)$ montrant que E est un polynôme multiple de G. Si lors de la transmission des erreurs se sont produites, cela se traduit par le fait que le polynôme reçu $M'(x) = E(x) + T(x)$, T étant le polynôme correspondant aux erreurs ($T(x) = x^i$ si le i^e bit a été inversé). À la réception le décodeur calcule le reste de $E(x)+T(x)/G(x)$ qui est en fait le reste de $T(x)/G(x)$ puisque E est un multiple de G. Si ce résultat est non nul, c'est que T est non nul et que des erreurs se sont produites. Évidemment, le résultat est également nul si T est un multiple de G ce qui masque des erreurs, mais le choix judicieux de G permet de minimiser ces erreurs non détectées. Enfin, il faut aussi remarquer un inconvénient de cette méthode qui signale des erreurs de transmission même si celles-ci ont eu lieu dans le CRC et non dans l'information à transmettre initialement. Dans ce cas il ne devrait pas être nécessaire de retransmettre l'information, or c'est ce qui est fait puisque globalement le transfert (info+CRC) a subi des perturbations.

- le code de Hamming est un code correcteur d'erreurs basé sur la notion de distance de Hamming. Soit un alphabet composé de 4 caractères (00,01,10,11). Si une erreur se produit alors le caractère émis est transformé en un autre caractère et il n'y a pas moyen de retrouver le caractère original. Par contre, en ajoutant de l'information de telle sorte que les caractères soient très différents les uns des autres cela devient possible. Par exemple, on peut coder les 4 caractères de la manière

caractère initial	00	01	10	11
caractère émis	00000	01111	10110	11001
caractères erronés possibles	00001	01110	10111	11000
	00010	01101	10100	11011
	00100	01011	10010	11101
	01000	00111	11110	10001
	10000	11111	00110	01001

Table 2.1 - Codage de Hamming.

illustrée dans la table 2.1. Ainsi si un bit (parmi les 5 émis) est erroné on sait quand même déterminer quel caractère a été émis, car comme on peut le voir dans la table 1.1 la modification d'un bit ne peut pas faire passer d'un caractère initial à l'autre. On a des ensembles «Terreurs possibles totalement disjoints. Par contre la modification de 2 bits dans cet exemple peut amener à des confusions et à l'impossibilité de corriger les erreurs.

Soit x et y deux caractères d'un alphabet A et soit N la longueur du codage des mots de cet alphabet, x_i et y_i désignent respectivement le i^e bit de x et y. On peut alors définir la distance $d(x,y) = \sum_{i=1}^{N} (x_i - y_i) \bmod 2$ qui permet de compter le nombre de bits qui diffèrent entre x et y.

On définit alors la distance de Hamming par $d_H = \inf_{(x,y)\in A^2 x\neq y} d(x,y)$. Dans l'exemple choisi ci-dessus, $d_H = 1$ dans le premier codage sur 2 bits et $d_H = 3$ dans le codage sur 5 bits. Chaque

erreur sur un bit d'un caractère *x* donne un caractère *x'* tel que $d(x, x') = 1$, donc pour pouvoir détecter et corriger une seule erreur il faut que $d_h \geq 3$ et pour corriger 2 erreurs il faut que $d_h \geq 5$. D'une manière générale on détecte et corrige n erreurs quand la distance de Hamming est $2n + 1$.
On peut voir ceci dans la figure 2.1 où sont représentés x et y (2 caractères parmi les plus proches de l'alphabet), x_i et y_i, des « déformations » de *x* et *y* après une erreur et x'_1 et y'_1 des « déformations » après deux erreurs. Ainsi, quand on reçoit un caractère x (erroné ou non on ne peut pas le savoir à l'avance) il suffit de chercher le caractère $c \in A$ le plus proche de x selon la distance d pour obtenir le caractère émis.
Un exemple de code de Hamming est donné par la technique suivante où l'on veut envoyer des caractères codés sur 4 bits de données ABCD. Pour cela on va émettre la suite $ABCP_2DP_1P_0$,

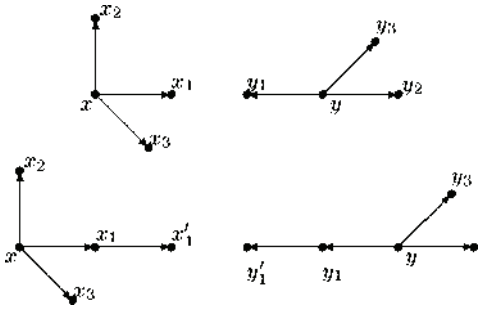


Figure 2.1 - Distance de Hamming.

dans laquelle les bits de contrôle P_i sont placés sur les bits de rang 2^i et sont définis par

$$\begin{cases} P_0 = A \oplus C \oplus D \\ P_1 = A \oplus B \oplus D \\ P_2 = A \oplus B \oplus C \end{cases}$$

Les P_i sont des bits de parité définis à l'aide des bits de données de rang k tels que la décomposition de k en somme de puissances de *2* contienne 2^i. Par exemple, A est un bit de donnée de rang $7=2^0+2^1+2^2$ donc A sert au calcul de P_0, P_1 et P_2. De même, le rang de *D* est $3= 2^0+2^1$ donc il sert au calcul de P_0 et P_1.
À la réception on calcule

$$\begin{cases} P'_0 = P_0 \oplus A \oplus C \oplus D \\ P'_1 = P_1 \oplus A \oplus B \oplus D \\ P'_2 = P_2 \oplus A \oplus B \oplus C \end{cases}$$

si on obtient $P'_2=P'_1=P'_0=0$ alors c'est que la transmission s'est passée sans problème. Sinon, la valeur binaire de $P'_2P'_1P'_0$ donne la place de l'erreur dans les bits reçus (en commençant par la droite). On corrige alors l'erreur et on recalcule les P'_i s'ils sont devenus tous nuls l'erreur a été corrigée, sinon il y avait eu au moins deux erreurs et on peut rejeter la suite de bits mais pas la corriger.
Par exemple, si l'on veut envoyer les 4 bits 0010, on va finalement émettre 0011001. Si l'on reçoit *0010001*, on trouve $P'_2P'_1P'_0 = 100$ c'est-à-dire 4, donc l'erreur était en 4e place. On corrige, pour obtenir 0011001 et un nouveau calcul donne $P'_2=P'_1=P'_0=0$ assurant que l'on a corrigé l'erreur. On peut remarquer qu'en fait on a corrigé une erreur qui n'était pas sur les données initiales mais sur les bits de parité rajoutés.

2.3 Protocoles de liaison de données.

2.3.1 Les protocoles de niveau trame.

Les protocoles définissent les règles à respecter pour que deux entités puissent communiquer de façon coordonnée. Pour cela, il faut que les deux entités communicantes utilisent le même protocole. Pour simplifier les communications de niveau trame, de nombreux protocoles ont été normalisés. Le plus ancien, HDLC, n'est quasiment plus utilisé mais reste un bon exemple de procédure de niveau trame. Un protocole de niveau trame que l'on utilise souvent sans le savoir est PPP, qui permet de relier deux PC entre eux. Nous allons le décrire succinctement dans la suite. Nous terminerons par le protocole ATM qui entre dans sa phase de décroissance, mais qui est encore fortement utilisé, et par la trame Ethernet, dont on prédit qu'elle occupera l'ensemble du marché.

2.3.2 Le protocole PPP (Point-to-Point Protocol).

PPP est utilisé dans les liaisons d'accès au réseau Internet ou sur une liaison entre deux équipements, qu'ils soient des ordinateurs personnels ou des nœuds de réseau. Son rôle est essentiellement d'encapsuler un paquet IP afin de le transporter vers le nœud suivant.
Tout en étant fortement inspiré du protocole HDLC, sa fonction consiste à indiquer le type des informations transportées dans le champ de données de la trame. Le réseau Internet étant multiprotocole, il est important de savoir détecter, par un champ spécifique de niveau trame, l'application qui est transportée de façon à pouvoir l'envoyer vers la bonne porte de sortie.
La trame du protocole PPP ressemble à celle de HDLC. Un champ déterminant le protocole de niveau supérieur vient s'ajouter juste derrière le champ de supervision. La figure 2.2 illustre la trame PPP.

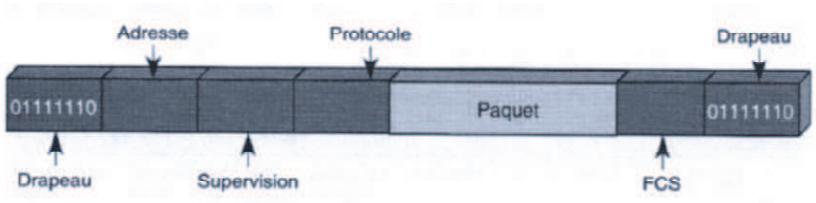


Figure 2.2 - Structure de la trame PPP

Les valeurs les plus classiques du champ de protocole sont les suivantes :

- 0x0021 : protocole IPv4 ;
- 0x002B : protocole IPX (Internetwork Packet eXchange) ;
- 0x002D : TCP/IP en-tête compressé ;
- 0x800F : protocole IPv6.

Les fonctionnalités de PPP étant très semblables à celles du protocole HDLC, nous convions les lecteurs à se reporter à l'annexe F.

2.3.3 La trame Ethernet.

La trame Ethernet a été conçue pour transporter des paquets dans les réseaux d'entreprise au moyen d'une méthode originale de diffusion sur un réseau local. Cette solution a donné naissance aux réseaux Ethernet partagés, dans lesquels la trame est émise en diffusion et où seule la station qui se reconnaît a le droit de recopier l'information. À cette solution de diffusion s'est ajoutée la commutation Ethernet.

Avant de se pencher sur les divers types de commutations Ethernet, indiquons qu'Ethernet utilise bien une trame puisque le bloc Ethernet est précédé d'une succession de 8 octets commençant par 10101010101010101, et ainsi de suite jusqu'à la fin du huitième octet, qui se termine par 11. Ce préambule est suffisamment long pour garantir qu'il ne soit pas possible de retrouver la même succession entre deux préambules, la probabilité de retrouver cette succession étant de $1/2^{64}$.

La structure de la trame Ethernet a été normalisée par l'IEEE (Institute of Electrical and Electronics Engineers), après avoir été défini à l'origine par le triumvirat d'industriels Xerox, Digital et Intel. Deux trames Ethernet coexistent donc, la version primitive du triumvirat fondateur et celle de la normalisation par l'IEEE. Le format de ces deux trames est illustré à la figure 2.3.

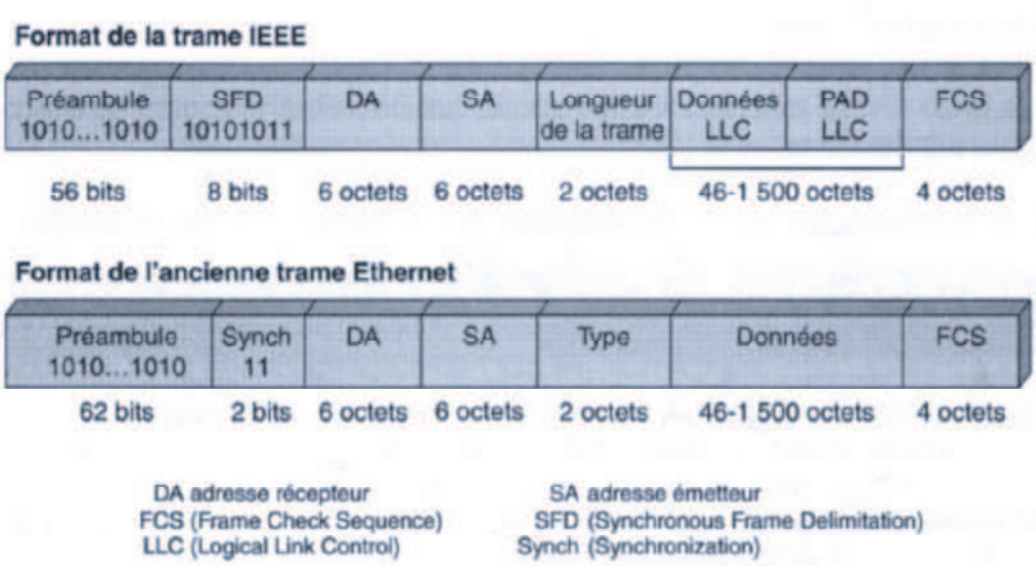


Figure 2.3 - Format des deux types de trames Ethernet

Dans le cas de la trame IEEE, le préambule est suivi d'une zone de début de message, appelée SFD (Start Frame Délimiter), dont la valeur est 10101011. Dans l'ancienne trame, il est suivi de 2 bits de synchronisation. Ces deux séquences sont en fait identiques, et seule la présentation diffère d'une trame à l'autre.

La trame contient l'adresse de l'émetteur et du récepteur, chacune sur 6 octets. Ces adresses sont dotées d'une forme spécifique du monde Ethernet, conçue de telle sorte qu'il n'y ait pas deux coupleurs dans le monde qui possèdent la même adresse. Dans cet adressage, dit plat, les trois premiers octets correspondent à un numéro de constructeur, et les trois suivants à un numéro de série. Dans les trois premiers octets, les deux bits initiaux ont une signification particulière. Positionné à 1, le premier bit indique une adresse de groupe. Si le deuxième bit est également à la valeur 1, cela indique que l'adresse ne suit pas la structure normalisée.

Regardons dans un premier temps la suite de la trame IEEE. La zone Longueur (Length) indique la longueur du champ de données provenant de la couche supérieure. La trame encapsule ensuite le bloc de niveau trame proprement dit, ou trame LLC (Logical Link Control). Cette trame encapsulée contient une zone PAD, qui permet de remplir le champ de données de façon à atteindre la valeur de 46 octets, qui est la longueur minimale que doit atteindre cette zone pour que la trame totale fasse 64 octets en incluant les zones de préambule et de délimitation.

L'ancienne trame Ethernet comporte en outre un type, qui indique comment se présente la zone de données (Data). Par exemple, si la valeur de cette zone est 0800 en hexadécimal, cela signifie que la trame Ethernet transporte un paquet IP.

La détection des erreurs est assurée par le biais d'un polynôme générateur $g(x)$ selon la formule :

$$g(x) = x^{32} + x^{26} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x^1$$

Ce polynôme donne naissance à une séquence de contrôle (CRC) sur 4 octets.

Pour se connecter au réseau Ethernet, une machine utilise un coupleur, c'est-à-dire une carte que l'on insère dans la machine et qui supporte le logiciel et le matériel réseau nécessaires à la connexion.

Comme nous l'avons vu, la trame Ethernet comporte un préambule. Ce dernier permet au récepteur de synchroniser son horloge et ses divers circuits physiques avec l'émetteur, de façon à réceptionner correctement la trame. Dans le cas d'un réseau Ethernet partagé, tous les coupleurs du réseau enregistrent la trame au fur et à mesure de son passage. Le composant électronique chargé de l'extraction des données incluses dans la trame vérifie la concordance entre l'adresse de la station de destination portée dans la trame et l'adresse du coupleur. S'il y a concordance, le paquet est transféré vers l'utilisateur après vérification de la conformité de la trame par le biais de la séquence de contrôle.

2.4 La commutation Ethernet.

Plusieurs sortes de commutations Ethernet se sont succédé. La première, qui est très répandue dans les entreprises, consiste à se servir de l'adresse MAC sur 6 octets comme d'une référence unique sur tout le chemin qui mène à la carte coupleur Ethernet. Toutes les trames qui se servent de la même référence vont au même endroit, c'est-à-dire à la carte coupleur qui possède l'adresse MAC indiquée. Cela signifie que tous les commutateurs du réseau doivent posséder une table de commutation, appelé «*lookup table*», comportant autant de lignes que de cartes Ethernet à atteindre.

Les mises à jour de ces fichiers sont complexes, car il faut ajouter ou retrancher des lignes sur l'ensemble des commutateurs du réseau pour toutes les cartes Ethernet qui s'activent ou se désactivent. La reconnaissance des adresses s'effectue par apprentissage. Lorsqu'une trame entre dans un nœud et que ce nœud ne possède pas l'adresse source de la trame dans sa table de commutation, le nœud ajoute une ligne à sa table, indiquant la nouvelle référence et la direction d'où vient la trame.Lorsqu'une trame arrivant dans un nœud ne trouve pas dans la table de commutation *(lookup table)* l'adresse du destinataire, la solution est d'émettre la trame en diffusion de telle sorte que le récepteur finisse par la recevoir.

Malgré ces solutions d'apprentissage automatique, la commutation Ethernet n'a pu se développer sur les très grands réseaux. En effet, la gestion des tables de commutation devenait

vite trop contraignante. On a là l'exemple d'un réseau commuté sans véritable système de signalisation. Cette solution s'est fortement développée dans les réseaux d'entreprise, en structurant le réseau en plusieurs sous-réseaux de façon à éviter les inondations de trames lorsque la table de commutation est incomplète.
La seconde solution de commutation a été apportée par MPLS. Elle consiste à introduire dans la trame Ethernet une référence spécifique, le *shim label,* ou *shim* MPLS, dans une nouvelle zone ajoutée à la trame Ethernet derrière l'adresse MAC.

La trame Ethernet est commutée de façon classique en utilisant la ligne d'entrée et la référence. Pour cela, il faut mettre en place les références tout le long du chemin. Les trames se succèdent en restant dans l'ordre d'émission. C'est pourquoi une signalisation explicite est indispensable dans ce type de réseau. MPLS a adopté le réseau IP comme système de signalisation.

Exercices du Chapitre 2 : La Couche Liaison de Données

Exercice 2.1 :

Une transmission de données engendre des erreurs en réception. Discutez les causes et les solutions de ses erreurs ?

Exercice 2.2 :

Soit l'information à emmètre i(x)=1110010001 avec le polynôme générateur $g(x)=x^4+x^3+x+1$

_ Donner le CRC Correspondant
_ Donner la trame complète à emmètre
_ Supposant que la séquence reçue de i(x) est 1110011001(sans les bits de parité). En utilisant le code de HAMMING détectez et corrigez les erreurs ? (on considère que les bits de parité sont correctement reçus).

Exercice 2.3 :

Soit le signal numérique I(x)=11100101101 on suppose que le signal est transmis en 01 seconde :

_ Donner le chronogramme du signal en bande de base en choisissant l'encodage le plus performant ? justifier ?

_ Soit à émettre I(x) donnez la représentation de cette séquence en code de Hamming ?

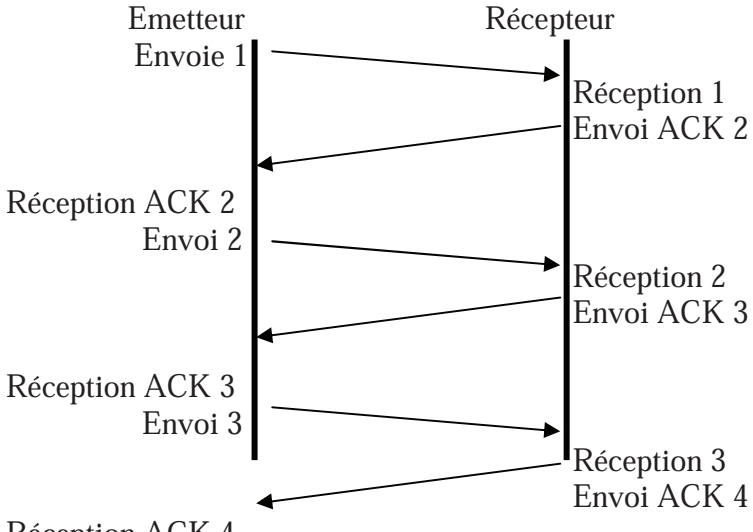
_ Soit le code générateur $G(x)=x^4+1$, donnez la trame finale à émettre ?

Exercice 2.4 :

En matière de durée de transmission, la trame Ethernet est la plus fiable parmi les autres trames. Expliquer ?

Exercice 2.5 :

Soit à émettre un bloc d'un certain nombre de paquets on suivant les règles du protocole décrit ci-dessous :



_ Décrire le fonctionnement de ce protocole ?
_ Quels sont les avantages et les inconvénients constatés ?
_ Rectifier ce protocole en le rendant plus fiable ?

La Couche Réseau.

3.1 Définition.

Le rôle de la couche réseau est de transporter d'une extrémité à l'autre du réseau des blocs de données provenant d'une fragmentation des messages du niveau supérieur, le niveau transport.
Le paquet est l'entité de la couche 3 qui possède l'adresse du destinataire ou la référence nécessaire à son acheminement dans le réseau. Le niveau paquet est en outre responsable du contrôle de flux, qui, s'il est bien conçu, évite les congestions dans les nœuds du réseau. Comme nous l'avons vu, les fonctionnalités de la couche réseau peuvent se trouver au niveau trame. Un paquet ne peut être transmis directement sur un support physique, car le récepteur serait incapable de reconnaître les débuts et fins de paquet.
L'ensemble des paquets allant d'un même émetteur vers un même destinataire s'appelle un flot. Celui-ci peut être long ou court, suivant la nature du service qui l'a créé. Si un gros fichier donne naissance à un flot important, une transaction ne produit qu'un flot très court, d'un à quelques paquets. Le niveau paquet peut faire appel ou non à une connexion pour négocier une qualité de service avec le destinataire du flot.
Avant d'aborder en détail les fonctionnalités de la couche réseau, nous commencerons par rappeler les caractéristiques de ce niveau, définies dans le cadre du modèle de référence. Le rôle de la couche 3 (réseau) est de transporter les paquets d'une extrémité à l'autre du réseau.

3.2 Caractéristiques de la couche réseau.

Le niveau paquet, ou couche réseau, est situé au troisième niveau de la hiérarchie de l'architecture du modèle de référence, comme illustré à la figure 3.1.

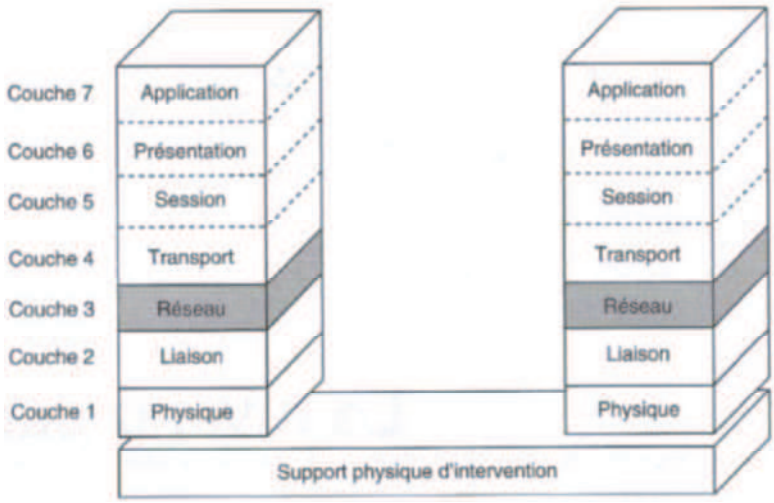


Figure 3.1 - Le niveau paquet, ou couche réseau.

Cette couche existe quand le réseau utilise un niveau paquet. Les fonctions définies par la normalisation internationale dans le cadre du modèle de référence sont les suivantes :

- Mise en place des connexions réseau pour acheminer les paquets. Cette fonction n'est toutefois pas une obligation dans le niveau paquet, et le protocole IP travaille en mode sans connexion. En revanche, la technique X.25 qui a presque disparu utilisait une mode avec connexion.
- Prise en charge de l'acheminement des paquets et des problèmes de passerelles pour atteindre un autre réseau.
- Multiplexage des connexions réseau.
- Prise en charge de la segmentation et du groupage.

- Détection de perte de paquets et reprise des paquets perdus. Cette fonction n'est pas obligatoire au niveau paquet. IP, par exemple, protocole de niveau paquet, n'a aucun moyen de récupérer les paquets perdus.
- Maintien en séquence des données remises à la couche supérieure, sans que cela soit une obligation.
- Contrôle de flux, de façon qu'il n'y ait pas de débordement des mémoires en charge de la transmission des paquets.
- Transfert de données expresses (non obligatoire).
- Réinitialisation de la connexion réseau (non obligatoire).
- Qualité de service (non obligatoire).
- Gestion de la couche réseau.

3.3 Les modes avec et sans connexion.

Dans le mode avec connexion, il faut établir une connexion entre les deux extrémités avant d'émettre les paquets de l'utilisateur, de façon que les deux entités communicantes s'échangent des informations de contrôle. Dans le mode sans connexion, au contraire, les paquets peuvent être envoyés par l'émetteur sans concertation avec le récepteur. On ne s'occupe pas de savoir si l'entité destinataire est prête à recevoir les paquets, et l'on suppose que le fait d'avoir une connexion au niveau de la session et, le cas échéant, de la couche transport est suffisant pour assurer un transfert simple de l'information à la couche 3. Il est vrai que les protocoles en mode sans connexion sont beaucoup plus simples que les protocoles en mode avec connexion.

En mode avec connexion, le réseau emploie généralement une technique de commutation. Quitte à envoyer un paquet de supervision pour demander l'ouverture de la connexion, autant se servir de la traversée du réseau par ce paquet de supervision, que l'on appelle aussi paquet d'appel, pour mettre en place des références, lesquelles permettront d'émettre à très haut débit sur le chemin ainsi mis en œuvre. En mode sans connexion, les gestionnaires réseau préfèrent le routage puisqu'il n'y a pas de signalisation. Il faut néanmoins noter qu'un réseau avec connexion peut se satisfaire d'une technique de routage et qu'un réseau sans connexion peut utiliser une commutation.

En mode sans connexion, une entité de réseau émet un paquet sans avoir à se soucier de l'état ni des désirs du récepteur. Comme pour l'ensemble des modes sans connexion, l'autre extrémité doit être présente, ou du moins représentée. Cette connexion implicite a été mise en place à un niveau supérieur, généralement le niveau session. Le mode sans connexion est beaucoup plus souple, puisqu'il ne tient pas compte de ce qui se passe au niveau du récepteur.

Les deux modes présentent des avantages et des inconvénients.

Les principaux avantages du mode avec connexion sont les suivants :

- sécurité de la transmission ;
- séquencement des paquets sur la connexion ;
- réglage facile des paramètres du protocole réseau.
- Ses principaux désavantages sont les suivants :
- lourdeur du protocole mis en œuvre, en particulier pour les paquets de petite taille ;
- difficultés à atteindre les stations en multipoint ou en diffusion, du fait de la nécessité d'ouvrir autant de connexions qu'il y a de points à atteindre ;
- débit relativement faible acheminé sur la connexion.

Les avantages du mode sans connexion sont les suivants :

- diffusion et émission en multipoint grandement facilitées ;

- simplicité du protocole, permettant des performances assez élevées.

Ses désavantages sont les suivants :

- faible garantie de la sécurité du transport des paquets ;
- réglage plus complexe des paramètres en vue d'atteindre les performances désirées.

3.4 Les principaux protocoles de niveau paquet.

Le protocole de niveau paquet quasiment exclusif est IP. Il est également un des plus anciens puisque les premières études provenant du ministère de la Défense aux États-Unis et du projet Cyclades en France ont démarré à la fin des années 1960. Le protocole IP est devenu stable au tout début des années 1980.

IP est une norme de fait de l'IETF (Internet Engineering Task Force). Cet organisme, qui n'a aucun pouvoir de droit propose des protocoles, dont certains finissent par s'imposer de par le nombre d'industriels qui les choisissent.

Un deuxième protocole de la couche réseau a connu son heure de gloire entre les années 1980 et 2000. Il a été normalisé par l'ISO (International Organization for Standardization) et l'UIT-T, les deux organismes de normalisation de droit puisque dépendant des États et représentant les utilisateurs et les industriels des télécommunications. Ce protocole est connu par son numéro de recommandation, X.25.3, ou X.25 PLP (Packet Level Protocol), et par son numéro de norme, ISO 8208.

3.5 Les grandes fonctionnalités de la couche réseau.

Comme expliqué précédemment, le rôle de la couche réseau consiste à prendre en charge les paquets et à les transporter d'une extrémité à l'autre du réseau vers le bon point de destination et dans les meilleures conditions possibles. Il existe pour cela deux façons de procéder : mettre en place un chemin, ou circuit virtuel, entre l'émetteur et le récepteur ou bien utiliser le mode sans connexion. Le mot chemin supplante l'expression circuit virtuel, le monde IP n'appréciant guère cette dernière expression, qui rappelle les technologies anciennes de téléphonie commutée. L'expression anglaise path est de plus en plus utilisée. Lorsque le chemin utilise une technique de commutation, l'expression consacrée est label-switched path, ou chemin commuté.

Dans le mode chemin, les paquets circulent de façon ordonnée pour arriver dans l'ordre où ils ont été émis. Pour ouvrir le chemin, il est nécessaire de se servir d'une signalisation. Celle-ci doit déposer au fur et à mesure de sa progression dans les nœuds du réseau les références qui seront utilisées par les paquets de données, comme illustré à la figure 3.2.

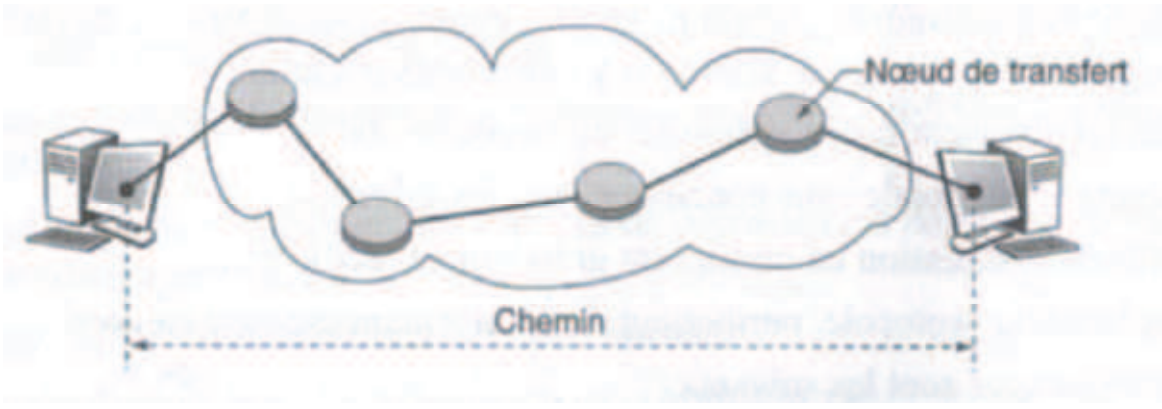


Figure 3.2 – Pose des références dans les nœuds du réseau.

Dans le mode sans connexion, dit aussi mode datagramme, chaque paquet est considéré comme indépendant des autres, même si tous les paquets appartiennent au même flot. Les

paquets peuvent prendre des chemins différents et arriver dans n'importe quel ordre au récepteur, contrairement à ce qui se produit dans le mode chemin, où les paquets arrivent toujours dans l'ordre d'émission. Le contrôle des différents paquets isolés demande des algorithmes spécifiques, qui sont présentés plus loin dans les sections consacrées aux contrôles de flux et de congestion.
Rien n'empêche de réaliser un réseau avec connexion en utilisant en interne un mode datagramme pour le transport des paquets, comme illustré à la figure 3.3. Un paquet de supervision est acheminé vers le récepteur pour établir la connexion. L'ouverture de la connexion peut avoir lieu sans l'existence d'un chemin.

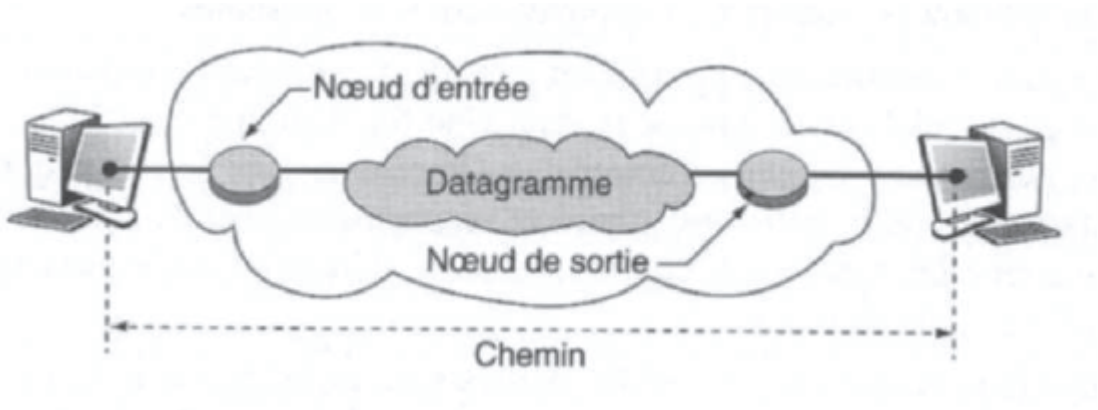


Figure 3.3 - Un chemin au-dessus d'un datagramme.

À l'inverse, rien ne s'oppose à bâtir un réseau sans connexion utilisant des chemins, mais cela n'apporte strictement rien. Il suffit d'envoyer un paquet de contrôle, qui dépose les références sans demander de connexion à l'hôte de destination.
Les trois fonctionnalités principales prises en charge par un protocole de niveau paquet sont le contrôle de flux, c'est-à-dire les moyens d'éviter que les flux ne grossissent trop par rapport aux ressources du réseau, la gestion des adresses ou des références et les algorithmes liés au routage.

3.6 Le contrôle de flux.

Le contrôle de flux est la première fonctionnalité demandée au niveau paquet. Il s'agit de gérer les paquets pour qu'ils arrivent au récepteur dans le laps de temps le plus court et, surtout, d'éviter des pertes par écrasement dans les mémoires tampons des nœuds intermédiaires en cas de surcharge. Les réseaux à transfert de paquets sont comme des autoroutes: s'il y a trop de paquets, personne ne peut avancer. La régulation du flux est toutefois un problème complexe.
De très nombreuses méthodes ont été testées dans des contextes spécifiques. Dans tous les cas, le contrôle s'effectue par une contrainte sur le nombre de paquets circulant dans le réseau. Cette limitation s'exerce soit sur le nombre de paquets en transit entre une entrée et une sortie ou sur l'ensemble du réseau, soit sur le nombre de paquets qu'on laisse entrer à l'intérieur du réseau par unité de temps. À ces contrôles peuvent s'ajouter des techniques d'allocation des ressources pour éviter toute congestion.

3.7 Le routage.

Dans un réseau maillé, le routage des paquets fait partie d'une algorithmique complexe, de par la distribution des décisions à prendre, qui relèvent à la fois de l'espace et du temps. Un nœud devrait connaître l'état de l'ensemble des autres nœuds avant de décider où envoyer un paquet, ce qui est impossible à réaliser.

Dans un premier temps, regardons les composantes nécessaires à la mise en place d'un routage. Il faut tout d'abord une table de routage, qui se présente comme illustré à la figure 3.4.

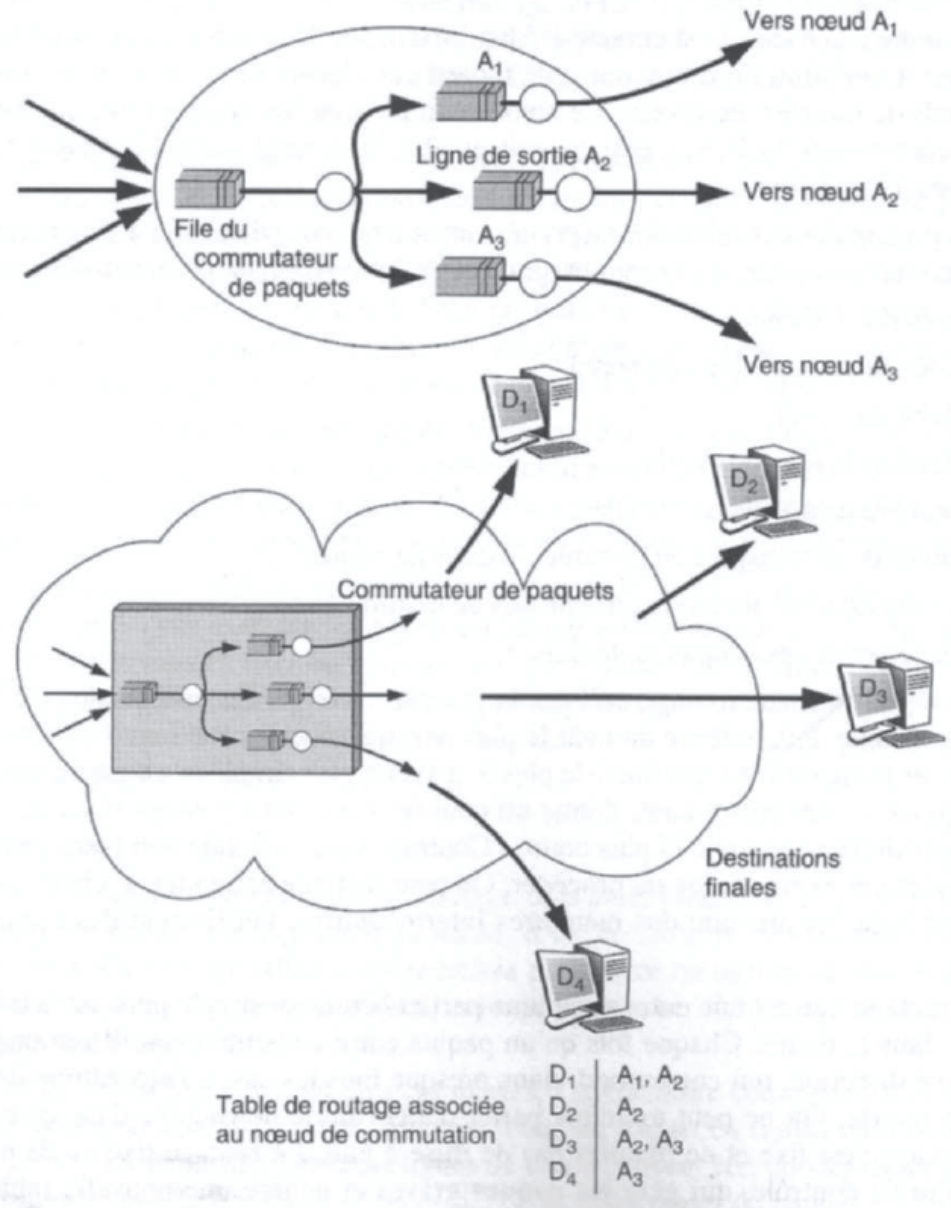


Figure 3.4 – Table de routage.

On voit qu'un nœud de transfert est formé de lignes de sortie, qui émettent des trames obtenues à partir de paquets. Les paquets sont routés par le nœud vers une ligne de sortie grâce à la table de routage. Si un paquet se présente au nœud avec, pour destination finale, le nœud D1 le nœud peut envoyer ce paquet vers la ligne de sortie A1, ou vers la ligne de sortie A2. La décision s'effectue sur des critères locaux dans le cas considéré. Par exemple, on envoie le paquet sur la file la plus courte. Si la destination finale est D2, le paquet est placé dans la file A2.

3.7.1 Le routage centralisé.

Le routage centralisé est caractérisé par l'existence d'un centre, qui prend les décisions quant à la définition d'une nouvelle table et de l'envoi de cette table à l'ensemble des nœuds de transfert du réseau. Ce nœud central reçoit les informations de la part de tous les composants du réseau, et il conçoit sa table de routage suivant des algorithmes déterminés à l'avance.
Les principales considérations à prendre en compte pour déterminer les meilleures routes dans un réseau, que ce soit en routage ou pour l'ouverture d'un chemin, sont les suivantes :

- coût des liaisons ;
- coût du passage dans un nœud ;
- débit demandé ;
- délai de transit demandé ;
- nombre de nœuds à traverser ;
- sécurité du transport de certaines classes de paquets ;
- occupation des mémoires des nœuds de commutation ;
- occupation des coupleurs de ligne.

Les algorithmes de routage utilisent la plupart du temps des critères de coût. On trouve, par exemple, l'algorithme du coût le plus bas, qui, comme son nom l'indique, consiste à trouver le chemin qui minimise le plus le prix. Le plus simple des algorithmes, et presque toujours le plus performant, donne un coût de 1 à chaque passage dans un nœud. C'est l'algorithme de la route la plus courte. Contrairement à ce que l'on pourrait penser, c'est souvent une bonne façon de procéder. On peut facilement ajouter des biais pour prendre en compte l'occupation des mémoires intermédiaires, l'utilisation des lignes de sortie, etc.
Le routage fixe est une autre technique particulièrement simple puisque la table ne varie pas dans le temps. Chaque fois qu'un paquet entre dans un nœud, il est envoyé dans la même direction, qui correspond, dans presque tous les cas, à l'algorithme de la route la plus courte. On ne peut toutefois parler d'algorithme de routage dans ce cas, puisque le routage est fixe et ne requiert pas de mise à jour. Le routage fixe va de pair avec un centre de contrôle, qui gère les pannes graves et génère une nouvelle table lorsqu'un nœud tombe en panne ou qu'une ligne de communication est rompue. On appelle ce routage fixe entre les mises à jour.
On peut améliorer le routage fixe en tenant compte d'événements indiqués par le réseau, telles des congestions ou des occupations de lignes ou des mémoires trop importantes. Toutes les dix secondes, tous les nœuds du réseau envoient un paquet de contrôle indiquant leur situation. À partir de ces comptes rendus, le nœud central élabore une nouvelle table de routage, qui est diffusée.
L'envoi des tables de routage d'une façon asynchrone est une technique plus élaborée. Le nœud central diffuse vers l'ensemble des nœuds une nouvelle table de routage dès que cette table a suffisamment changé par rapport à celle en vigueur. En d'autres termes, le centre de contrôle dresse des tables de routage au fur et à mesure de l'arrivée de nouvelles informations puis envoie à tous les nœuds la première table de routage qui lui paraît suffisamment différente de la précédente. L'adaptation est ici asynchrone et non pas synchrone, comme précédemment.
Les performances de ce routage centralisé dépendent de l'architecture et de la topologie du réseau. En effet, le principal problème du routage et de l'adaptation est qu'ils doivent s'effectuer en temps réel. Entre le moment où un nœud envoie un compte rendu impliquant un nouveau routage et celui où la nouvelle table de routage arrive, il ne doit pas y avoir de changement substantiel de l'état du système. Cette condition est très mal réalisée si le réseau est important et les artères surchargées, les paquets de contrôle étant peu prioritaires par rapport aux paquets transportant des informations.

La qualité du routage correspond à première vue à une adaptation de plus en plus sophistiquée. C'est là que se pose le deuxième grand problème concernant les performances, d'ailleurs lié au premier : la sophistication entraîne une surcharge du réseau par des paquets de contrôle, laquelle peut empêcher un fonctionnement en temps réel.
On voit qu'un algorithme de routage donné n'a pas la même efficacité pour un réseau à trois nœuds, par exemple, que pour un réseau à vingt nœuds. La première conclusion que nous pouvons en tirer est qu'il n'existe pas d'algorithme meilleur qu'un autre, même pour un réseau bien déterminé, puisque tout dépend du trafic. Par ailleurs, il semble qu'il existe un optimum dans la complexité de l'algorithme d'adaptation pour ne pas surcharger le réseau inutilement.

3.7.2 Le routage distribué.

La plus simple des techniques de routage distribué, l'inondation, n'est pas adaptative. Lorsqu'un paquet est reçu dans un nœud, il est retransmis vers toutes les destinations possibles. Ce routage efficace est toutefois pénalisant en termes de flux et ne peut être adopté que dans des cas spécifiques, comme les réseaux dans lesquels le temps réel est primordial et le trafic faible.
Dans les algorithmes un peu plus complexes, l'adaptabilité commence à apparaître. Elle ne concerne qu'une dimension, le temps. Pour un paquet en transit dans le nœud i et se dirigeant vers le nœud j, plusieurs lignes de sortie peuvent être choisies. Dans la méthode de routage appelée hot-potatoe, on essaie de se débarrasser du paquet le plus rapidement possible en le transmettant sur la première ligne de sortie vide. En réalité, on ne se sert jamais d'une méthode hot-potatoe pure. On préfère des techniques plus élaborées, dans lesquelles des coefficients sont affectés aux différentes lignes de sortie pour une destination donnée (voir figure 3.5).

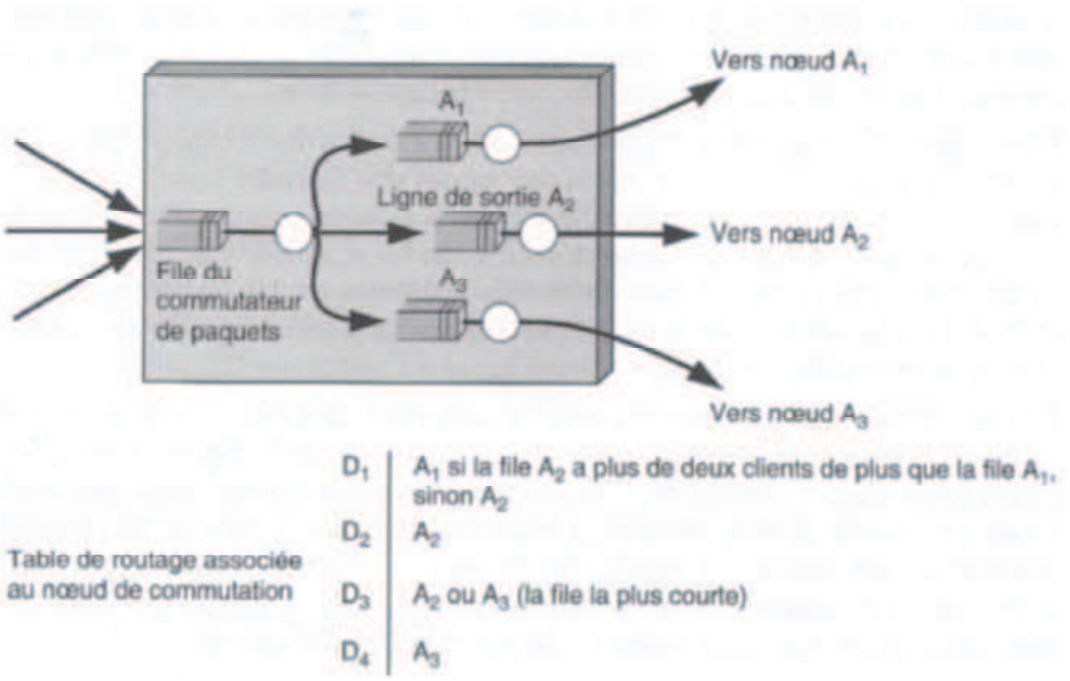


Figure 3.5 - Routage hot-potatoe avec biais.

Il existe presque toujours une ligne de sortie plus appropriée que les autres. Toutes ces techniques sont locales, puisque les états des autres nœuds ne sont pas pris en compte. Pour adapter l'algorithme dans l'espace, il convient en premier lieu de se faire une idée de ce qui se passe dans les nœuds voisins. Sans utiliser de paquets de contrôle, on obtient un échantillon

du trafic des nœuds voisins en comptabilisant les arrivées en provenance de ces nœuds. Ces derniers peuvent également envoyer de façon synchrone ou asynchrone des comptes rendus de leur état. En tenant compte de ces informations implicites ou explicites, il est possible de choisir la file de sortie en connaissance de cause. L'adaptation spatiale est encore limitée, puisqu'une cassure, deux chaînons plus loin, risque de ne pas être prise en compte par les comptes rendus des nœuds voisins. Une technique plus fine est définie permettant à un nœud de déterminer sa table de routage en fonction de l'ensemble des états du réseau.
L'algorithme de routage distribué, qui doit s'adapter pleinement à la fois dans l'espace et dans le temps, demande une connaissance complète de l'état de tous les nœuds du réseau. Les divers nœuds doivent donc s'échanger des messages. Si chaque nœud transmet un message à tous les autres, le trafic total risque d'augmenter de façon inquiétante. Pour rester dans des limites raisonnables, un nœud ne transmet un compte rendu qu'à ses voisins. Ceux-ci doivent en tenir compte dans leur propre compte rendu. De nouveau, les instants de mise à jour peuvent être synchrones ou asynchrones. Les inconvénients et les avantages de ces deux types de mesure sont les mêmes que dans le cas centralisé.

3.8 L'adressage.

Les données situées chez l'utilisateur ou sur des serveurs d'un réseau ne peuvent être atteintes que par l'intermédiaire d'un adressage spécifiant l'interface de sortie ou par une référence permettant d'acheminer le paquet jusqu'à l'interface recherchée. Dans ce dernier cas, il faut se servir de l'adresse du destinataire pour que le paquet de signalisation ouvre un circuit virtuel. Nous ne nous intéressons dans un premier temps qu'à l'adressage et revenons ensuite sur les systèmes utilisant des références.
L'adressage peut être physique ou logique. Une adresse physique correspond à une jonction physique à laquelle est connecté un équipement terminal. Une adresse logique correspond à un utilisateur, un terminal ou un programme utilisateur qui peut se déplacer géographiquement. Le réseau téléphonique offre un premier exemple d'adressage physique : à un numéro correspond un utilisateur, ou plus exactement une jonction. Dans ce réseau, l'adressage est hiérarchique. Il utilise un code différent pour le pays, la région et l'autocommutateur, les quatre derniers chiffres indiquant l'abonné. Si l'abonné se déplace, il change de numéro. Les autocommutateurs temporels peuvent dérouter l'appel vers un autre numéro à la demande de l'abonné, mais l'adressage n'est pas conservé.
Un second exemple est proposé par le réseau Ethernet et plus globalement par les réseaux locaux. Il s'agit d'un adressage de niveau trame et non de niveau paquet. Nous l'introduisons dans ce chapitre comme exemple car il aurait très bien pu être implémenté dans un paquet.
Par l'intermédiaire de l'IEEE (Institute of Electrical and Electronics Engineers), à chaque coupleur est affecté un numéro unique. Il n'y a donc pas deux coupleurs portant la même adresse. Si la partie portant l'adresse ne peut être déplacée, l'adressage est physique. En revanche, si l'utilisateur peut partir avec son terminal et son interface et se reconnecter ailleurs, l'adressage devient logique. Dans ce dernier cas, le routage dans les grands réseaux est particulièrement complexe.
Dans le cas du réseau Ethernet, l'adressage est absolu. Il n'y a donc pas de relation entre des adresses situées sur des sites proches l'un de l'autre. Comme indiqué à la figure 3.6, le premier bit de l'adressage Ethernet précise si l'adresse correspond à un seul coupleur (adresse unique) ou si elle est partagée par d'autres coupleurs pour permettre des communications en multipoint ou en diffusion. Le deuxième bit indique si l'adressage utilisé est celui défini par l'IEEE, c'est-à-dire si les champs d'adresse possèdent bien l'adresse Ethernet du coupleur ou si l'utilisateur a remplacé les deux champs par une adresse spécifique. Il est fortement conseillé de garder l'adresse IEEE d'origine du coupleur pour éviter toute collision avec une autre adresse IEEE.

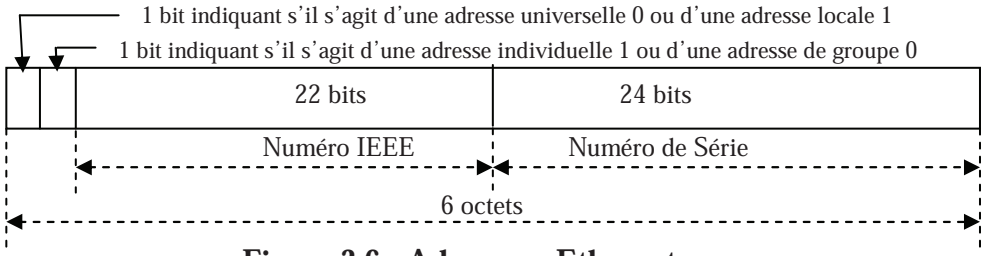


Figure 3.6 - Adressage Ethernet.

Dans l'adressage Ethernet que nous venons de décrire, la situation géographique de l'abonné est impossible à connaître, et le routage délicat à mettre en œuvre. C'est la raison pour laquelle le routage de niveau trame ne s'est jamais développé. Il aurait fallu pour cela développer un adressage hiérarchique.
Le réseau Internet donne un bon exemple d'adressage hiérarchique logique. L'adresse est décomposée en deux parties donnant deux niveaux de hiérarchie. La première identifie une machine terminale sur un réseau déterminé, et la seconde le numéro de ce réseau. On voit que l'adresse n'est pas forcément géographique, puisqu'un réseau peut contenir cinq PC, un dans chaque continent. Il faut trouver une route pour aller dans le réseau d'appartenance puis rechercher à l'intérieur du réseau la route allant au destinataire. Les adresses IP sont examinées en détail au chapitre 6.

3.9 Les fonctionnalités de la couche réseau.

Comme expliqué en début de chapitre, le niveau paquet (couche 3), également appelé couche réseau, offre un service au niveau message, ou couche transport. Ce service doit être fourni par le protocole réseau en tenant compte du service qui est offert par la couche inférieure.
Les services qui doivent être rendus par le niveau paquet doivent satisfaire les cinq critères suivants :

- Indépendance par rapport aux supports de transmission sous-jacents. Le service réseau libère ses utilisateurs de toutes les préoccupations liées à la façon dont sont utilisés les divers sous-réseaux pour assurer le service réseau. Il masque à l'utilisateur du service réseau la façon dont ses paquets sont transportés. En effet, ce dernier ne souhaite généralement pas connaître les protocoles utilisés pour le transport de ses paquets mais désire seulement être sûr que ses paquets arrivent au destinataire avec une qualité définie.
- Transfert de bout en bout. Le service réseau assure le transfert d'une extrémité à l'autre des données utilisateur. Toutes les fonctions de routage et de relais sont assurées par le fournisseur du service réseau, y compris dans le cas où diverses ressources de transmission, similaires ou différentes, sont utilisées en tandem ou en parallèle.
- Transparence des informations transférées. Le service réseau assure le transfert transparent des informations sous la forme d'une suite d'octets de données utilisateur ou d'informations de contrôle. Il n'impose aucune restriction quant au contenu, au format ou au codage des informations et n'a pas besoin d'interpréter leur structure ou leur signification.

- Choix de la qualité de service. Le service réseau offre aux utilisateurs la possibilité de demander ou d'accepter la qualité de service prévue pour le transfert de données utilisateur. La qualité de service est spécifiée par des paramètres de QoS exprimant des caractéristiques telles que le débit, le temps de transit, l'exactitude et la fiabilité.

- Adressage de l'utilisateur du service réseau. Le service réseau utilise un système d'adressage qui permet à chacun de ses utilisateurs d'identifier de façon non ambiguë d'autres utilisateurs du service réseau.

Pour respecter ces critères, le service réseau dispose d'une panoplie de possibilités. L'une des plus importantes consiste à établir une connexion entre les deux entités communicantes. Comme nous l'avons vu à plusieurs reprises, une connexion est un lien entre l'émetteur et le récepteur, qui doit permettre à la communication de s'effectuer dans les meilleures conditions possibles. Ce lien donne la possibilité à l'émetteur de négocier avec le récepteur les caractéristiques et la qualité de service qu'il souhaite obtenir.
On peut reprocher au mode avec connexion d'être relativement lourd, puisque, avant toute émission de paquets, une discussion avec le récepteur est nécessaire pour que les deux entités distantes se mettent d'accord. Le mode avec connexion est également pénalisant dès que l'on veut mettre en place des communications en diffusion ou en multipoint, puisqu'il faut ouvrir autant de connexions qu'il y a de points à atteindre. En contrepartie, son avantage évident réside dans une plus grande fiabilité de la communication grâce au contrôle effectué sur la connexion.
Le mode sans connexion est plus souple puisque l'émetteur envoie ses paquets sans avoir à se préoccuper du récepteur. Il est bien entendu que l'utilisateur distant est présent ou représenté par une boîte aux lettres et que cette présence peut être garantie par l'ouverture d'une connexion à un niveau supérieur. C'est le plus souvent au niveau session que l'assurance de cette présence est garantie.

3.10 La qualité de service.

La notion de qualité de service, ou QoS, concerne certaines caractéristiques d'une connexion réseau relevant de la seule responsabilité du fournisseur du service réseau.
Une valeur de QoS s'applique à l'ensemble d'une connexion réseau. Elle doit être identique aux deux extrémités de la connexion, même si cette dernière est prise en charge par plusieurs sous-réseaux interconnectés offrant chacun des services différents.
La QoS est décrite à l'aide de paramètres. La définition d'un paramètre de QoS indique la façon de mesurer ou de déterminer sa valeur, en mentionnant au besoin les événements spécifiés par les primitives du service réseau.
Deux types de paramètres de QoS ont été définis :

- Ceux dont les valeurs sont transmises entre utilisateurs homologues au moyen du service réseau pendant la phase d'établissement de la connexion réseau. Au cours de cette transmission, une négociation tripartite peut avoir lieu entre les utilisateurs et le fournisseur du service réseau afin de définir une valeur pour ces paramètres de QoS.
- Ceux dont les valeurs ne sont ni transmises ni négociées entre les utilisateurs et le fournisseur du service réseau. Pour ces paramètres de QoS, il est toutefois possible d'obtenir, par des moyens locaux, l'information relative aux valeurs utiles au fournisseur et à chacun des utilisateurs du service réseau.

Les principaux paramètres de QoS sont les suivants :

- Délai d'établissement de la connexion réseau. Correspond au temps qui s'écoule entre une demande de connexion réseau et la confirmation de la connexion. Ce paramètre de QoS indique le temps maximal acceptable par l'utilisateur.
- Probabilité d'échec de l'établissement de la connexion réseau. Cette probabilité est établie à partir des demandes qui n'ont pas été satisfaites dans le temps normal imparti pour l'établissement de la connexion.

- Débit du transfert des données. Le débit définit le nombre d'octets transportés sur une connexion réseau dans un temps raisonnablement long (quelques minutes, quelques heures ou quelques jours). La difficulté à déterminer le débit d'une connexion réseau provient de l'asynchronisme du transport des paquets. Pour obtenir une valeur acceptable, il faut observer le réseau sur une suite de plusieurs paquets et considérer le nombre d'octets de données transportés en tenant compte du temps écoulé depuis la demande ou l'indication de transfert des données.
- Temps de transit lors du transfert des données. Le temps de transit correspond au temps écoulé entre une demande de transfert de données et l'indication de transfert des données. Ce temps de transit est difficile à calculer du fait de la distribution géographique des extrémités. La satisfaction d'une qualité de service sur le temps de transit peut de surcroît entrer en contradiction avec un contrôle de flux.
- Taux d'erreur résiduelle. Se calcule à partir du nombre de paquets qui arrivent erronés, perdus ou en double sur le nombre total de paquets émis. C'est donc un taux d'erreur par paquet. Désigne également la probabilité qu'un paquet n'arrive pas correctement au récepteur.
- Probabilité d'incident de transfert. Est obtenue par le rapport du nombre d'incident répertorié sur le nombre total de transfert effectué. Pour avoir une estimation correcte de cette probabilité, il suffit d'examiner le nombre de déconnexion du réseau par rapport au nombre de transfert effectué.
- Probabilité de rupture de la connexion réseau. Se calcule à partir du nombre de libération et de réinitialisation d'une connexion réseau par rapport au nombre de transfert effectué.
- Délai de libération de la connexion réseau. C'est le délai maximal acceptable entre une demande de déconnexion et la libération effective.
- Probabilité d'échec lors de la libération de la connexion réseau. C'est le nombre d'échec de libération demandée par rapport au nombre total de libération demandé.

Les trois paramètres additionnels suivants permettent de caractériser la qualité de service :

- Protection de la connexion réseau. Détermine la probabilité que la connexion réseau soit en état de marche durant toute la période où elle est ouverte par l'utilisateur. Il y a plusieurs moyens de protéger une connexion en la dupliquant ou en ayant une connexion de sauvegarde prête à être ouverte en cas de coupure. La valeur pour un réseau téléphonique est de 99,999 %, que l'on appelle les cinq neuf, ce qui équivaut à quelques minutes d'indisponibilité par an. La protection est beaucoup plus faible pour un réseau IP, avec une valeur de l'ordre de 99,9 %, ou trois neuf. Cette valeur pose d'ailleurs problème pour la téléphonie sur IP, qui demande une protection plus forte des connexions téléphoniques.
- Priorité de la connexion réseau. Détermine la priorité d'accès à une connexion réseau, la priorité de maintien d'une connexion réseau et la priorité des données sur la connexion.
- Coût maximal acceptable. Détermine si la connexion réseau est tolérable ou non. La définition du coût est assez complexe puisqu'elle dépend de l'utilisation des ressources nécessaires à la mise en place, au maintien et à la libération de la connexion réseau.

Exercices du Chapitre 3 : Couche Réseau

Exercice 3.1:

Quelles approches de commutation sont utilisées par la couche réseau ? Les rappeler, ainsi que leurs avantages et inconvénients.

Exercice 3.2:

La couche session d'une station A située sur le LAN1 envoie un message à son homologue de la station B située sur le LAN2. Schématiser le cheminement du message au travers des différentes couches sur les stations et les équipements d'interconnexion.

Exercice 3.3:

Tracer un tableau comparatif entre le routage centralisé vs le routage distribué.

Exercice 3.4:

Définir la Qos au niveau de la couche réseau, et donner les quatre qualités de service les plus répondus ?

Exercice 3.5:

Pour transmission multimédia quelles les considérations principales de choix pour un routeur pour sélectionner la meilleure route ?

La couche transport.

4.1 Définition.

La couche transport assure un transfert de données transparents entre entités de session et en les déchargeant des détails d'exécution. Elle a pour rôle d'optimiser l'utilisation des services de réseau disponibles afin d'assurer au moindre coût les performances requises par la couche session.

C'est la première couche à résider sur les systèmes d'extrémité. Elle permet aux deux applications de chaque extrémité de dialoguer directement indépendamment de la nature des sous-réseaux traversés et comme si le réseau n'existait pas. Au niveau inférieur de la couche réseau seule la phase d'établissement de la liaison logique s'effectue de bout en bout, alors que les transferts d'information se font de proche en proche.

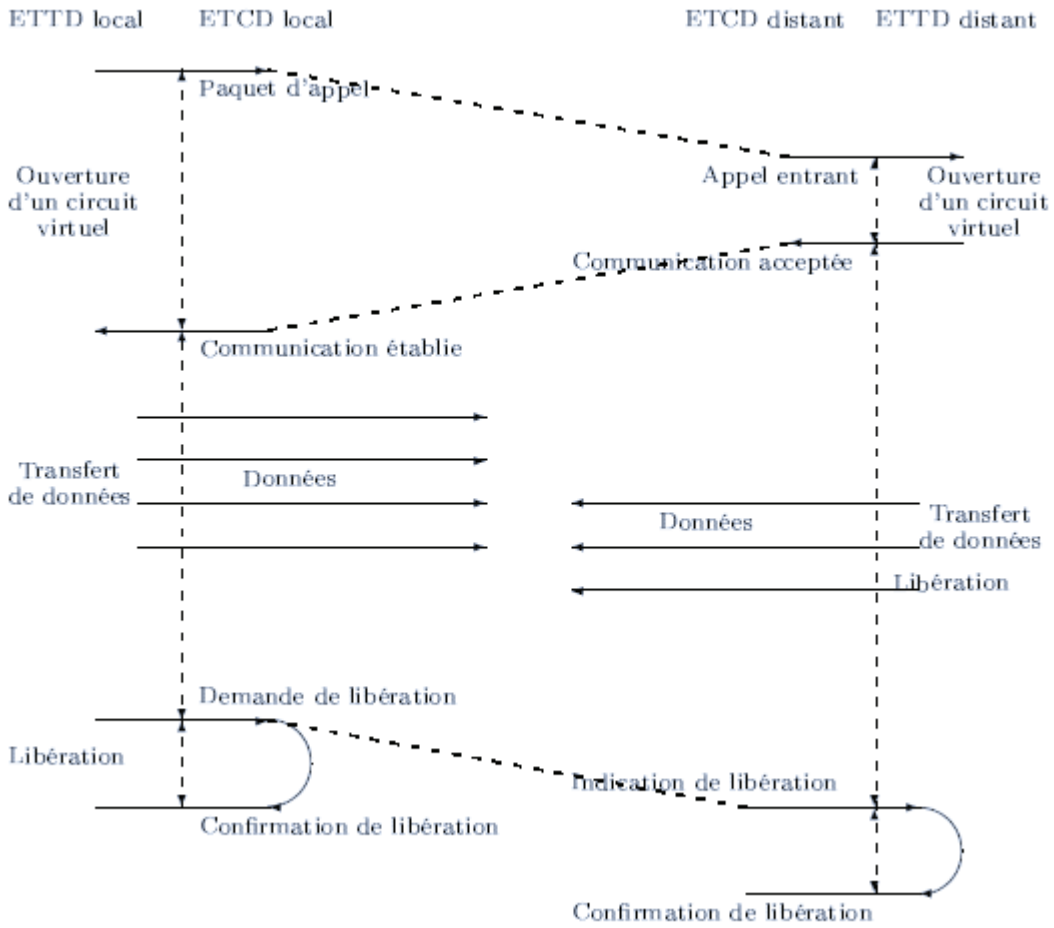


Figure 4.1 - Vie d'un circuit virtuel.

La couche transport doit assurer, en mode connecté ou non connecté, un transfert transparent de données entre utilisateurs de service réseau en leur rendant invisible la façon dont les ressources de communication sont mises en œuvre. Cette notion de transparence implique par exemple de pouvoir acheminer de bout en bout des TPDU (Transport Protocol Data Unit)

dont la taille peut varier de 128 à 8192 octets. Cette taille est cependant fixe une fois que la valeur a été négociée.

4.2 Qualité de service.

Une autre façon de définir la couche transport est de la considérer garante de la qualité de service (QOS) rendue par la couche réseau. Si cette dernière est sans faille, alors le travail de la couche transport est minime. Dans le cas contraire elle assure la jonction entre ce que désire l'utilisateur et ce que la couche réseau met à disposition en terme de qualité. La qualité est évaluée sur certains paramètres avec trois types de valeurs possibles : préféré, acceptable et inacceptable qui sont choisis lors de l'établissement d'une connexion (certains paramètres sont cependant disponibles pour un mode sans connexion). La couche transport surveille alors ces paramètres pour déterminer si la couche réseau sous-jacente assure la qualité de service demandée.

Les différents paramètres de la qualité de service sont :

- le temps d'établissement de la connexion transport: c'est la durée qui s'écoule entre le moment où une demande de connexion est émise et le moment où la confirmation de cet établissement est reçu et plus ce délai est court meilleure est la qualité de service.
- la probabilité d'échec d'établissement mesure la chance (ou plutôt malchance) qu'une connexion ne puisse s'établir dans un délai maximum défini. On ne tient pas compte ici du refus de l'entité distante d'établir cette connexion, mais on considère plutôt les problèmes d'engorgement de réseau.
- le débit de la liaison mesure le nombre d'octets utiles qui peuvent être transférés en une seconde, ce débit est évalué séparément dans les deux sens.
- le temps de transit mesure le temps écoulé entre le moment où l'utilisateur du service de transport envoie un message et celui où l'entité de transport réceptrice le reçoit, ce temps est évalué séparément dans les deux sens.
- le taux d'erreur résiduel est le rapport entre le nombre de messages perdus ou mal transmis et le nombre total de messages émis au cours d'une période considérée. Ce nombre, en théorie nul, a une valeur faible.
- la probabilité d'incident de transfert mesure le bon fonctionnement du service transport. Lorsqu'une connexion est établie, un débit, un temps de transit, un taux résiduel d'erreurs sont négociés. La probabilité d'incident mesure la fraction de temps durant laquelle les valeurs fixées précédemment n'ont pas été respectées.
- le temps de déconnexion mesure la durée s'écoulant entre une demande de déconnexion émise et la déconnexion effective du système distant.
- la probabilité d'erreur de déconnexion est le taux de demandes de déconnexion non exécutées pendant le temps maximum défini.
- la protection est définie comme la possibilité de se prémunir contre les intrusions passives (interférences sur une même ligne) et actives (écoute et modification des données transmises).
- la priorité permet à l'utilisateur de privilégier certaines transmissions par rapport à d'autres.
- la résiliation est la liberté laissée à la couche transport de décider elle-même de la déconnexion suite à un problème.

En mode non connecté, ces paramètres indiquent uniquement les souhaits de l'utilisateur en ce qui concerne le débit, le délai de transit, le taux d'erreur résiduel et la priorité d'une transmission. Ces paramètres sont utilisées apr la couche transport pour fixer des options de protocoles avant d'être soumis à la couche réseau.

Lorsqu'une connexion est demandée, tous ces paramètres sont transmis par l'utilisateur à la couche transport, les valeurs désirées et minimales sont spécifiées. Les différents cas de figure et étapes peuvent alors se présenter.

- Si la demande semble irréaliste pour certains paramètres alors la demande de connexion n'est même pas exécutée et un message d'erreur est renvoyé pour expliquer le problème.
- Si la demande ne peut pas être satisfaite complètement mais partiellement (par exemple avec un débit moindre mais acceptable), alors c'est cette demande avec des objectifs moindres qui est soumise.
- Si l'ordinateur distant ne peut satisfaire complètement la demande, mais reste au-dessus du minimum requis, alors il modifie aussi le paramètre.
- S'il ne peut pas rester au-dessus de ce minimum, il rejette la demande de connexion.
- Enfin, la couche transport avertit son utilisateur de la bonne fin (ou non) de la procédure de connexion et lui transmet les paramètres acceptés.

Cette procédure s'appelle la négociation des options qui, une fois fixées, restent inchangées pendant toute la connexion. Pour que tous les utilisateurs ne demandent pas une qualité de service optimale, les prix pratiqués par les fournisseurs de réseaux croissent avec cette qualité de service.

Primitives du service transport.

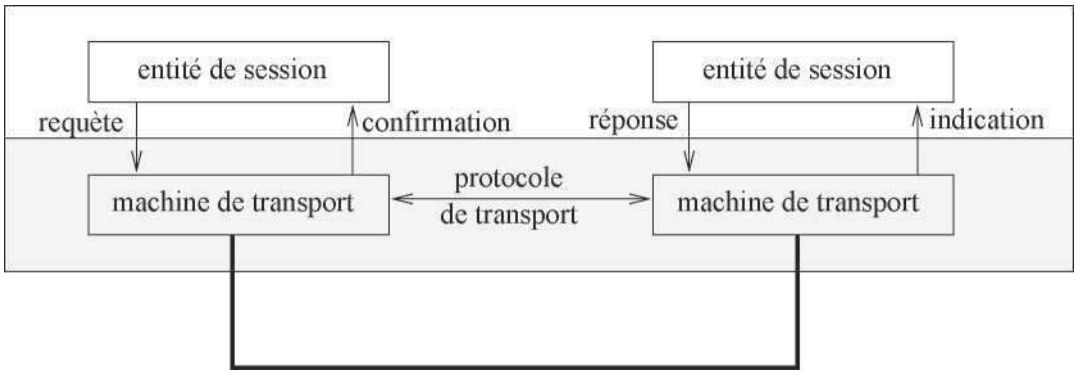


Figure 4.2 - Dialogue de niveau transport.

Les services qu'offrent, la couche transport en mode connecté sont rendues par les primitives données ci-dessous. Celles-ci se décomposent comme dans tout dialogue entre couches en quatre catégories comme illustré dans la figure 4.2.

phase d'établissement, de la connexion

- T_CONNECT.request(adresse source, adresse distante, données_exprès, qos, données_utilisateur) pour demander une connexion
- T_CONNECT.indication(adresse source, adresse distante, données_exprès, qos, données_utilisateur) pour indiquer une connexion de transport
- T_CONNECT.response(adresse source, adresse distante, données_exprès, données_utilisateur) pour répondre à une demande de connexion de transport
- T_CONNECT.confirm(adresse source, adresse distante, données_exprès, qos, données_utilisateur) pour confirmer rétablissement d'une connexion de transport

phase de transfert de données

- T_DATA.request(données_utilisateur) pour demander le transfert de données
- T_DATA.indication(données_utilisateur) pour indiquer un transfert de données
- T_EXPEDITED_DATA.request(données_utilisateur) pour demander le transfert de données exprès
- T_EXPEDITED_DATA.indication(données_utilisateur) pour indiquer un transfert de données exprès

phase de libération de la connexion

- T_DISCONNECT.request(données utilisateur) pour demander une déconnexion de transport
- T_DISCONNECT. indication (raison, données_utilisateur) pour indiquer une déconnexion de transport

La figure 4.3 détaille les différents enchaînements de primitives possibles. Pour chaque cas, un utilisateur est placé à gauche des lignes doubles et son homologue est à droite, la couche transport est entre les lignes et le temps s'écoule de haut en bas.

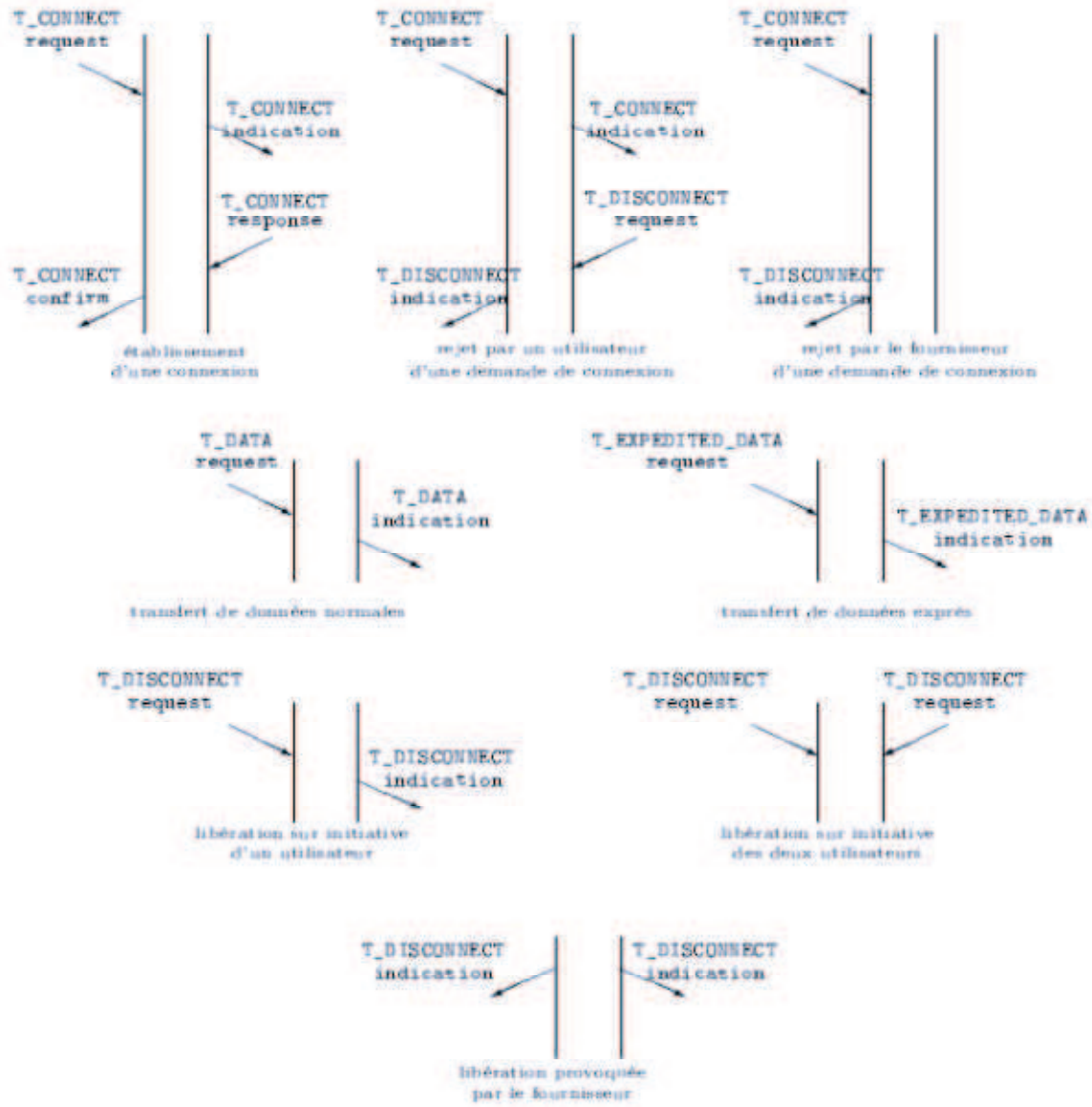


Figure 4.3 - Enchaînements de primitives en mode connecté.

Seules les successions de primitives décrites dans l'automate de la figure 4.4 sont autorisés assurant ainsi qu'une machine réalisant le service de transport ne peut se trouver que dans l'un des quatre états représentés à savoir

- veille : aucune connexion n'est établie, une demande de connexion peut être émise ou reçue
- connexion sortante en attente : la machine a demandé une connexion et la réponse de l'autre extrémité n'est pas encore arrivée
- connexion entrante en attente : la machine a reçu une demande de connexion qu'elle n'a pas encore acceptée ou rejetée.
- transfert de données prêt : une connexion a été établie, les transferts de données peuvent commencer

Pour ce qui est du mode non connecté seules les primitives suivantes sont disponibles.

TJJNIDATA.request(appelé, appelant, qos, données utilisateur)

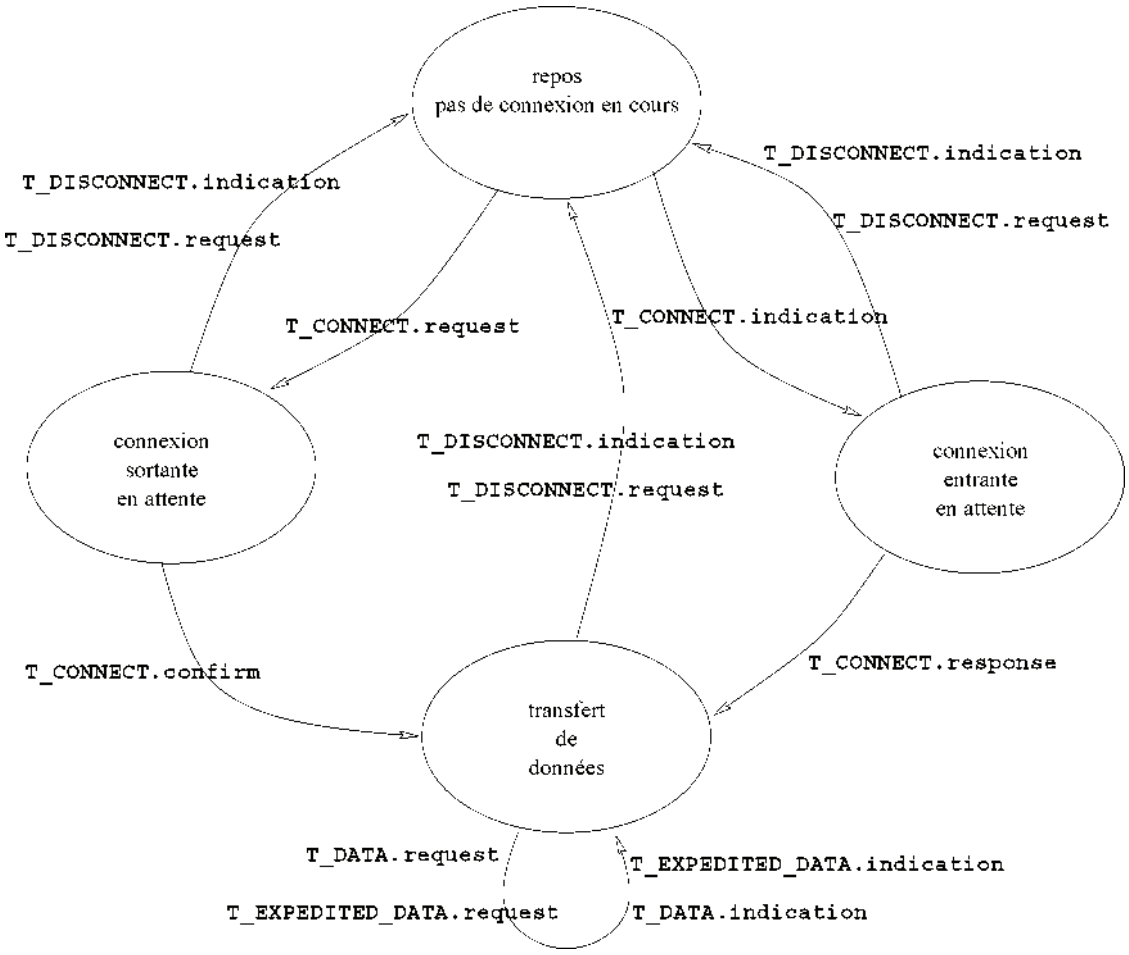


Figure 4.4 - Diagramme d'états d'une machine de service de transport.

TJJNIDATA.indication(appelé, appelant, qos, données utilisateur)

4.3 Le protocole de transport ISO en mode connecté (ISO 8073 ou X.224).

Les fonctions de la couche transport ISO sont établies dans le but de combler l'écart existant entre les services offerts par les couches basses et les services à offrir.
Les principales fonctions réalisées pendant l'établissement de la connexion sont les suivantes.

- La sélection d'une connexion réseau appropriée.
- La décision de mettre en place un multiplexage ou un éclatement. Le multiplexage a pour but de regrouper plusieurs connexions de transport sur une même connexion de réseau, ce qui va permettre de minimiser les ressources utilisées au niveau réseau, donc de diminuer les coûts de communication. Ceci est surtout utile si l'on a de nombreuses connexions de faible débit à établir. À l'inverse, l'éclatement consiste à répartir une connexion de transport simultanément sur plusieurs connexions réseau dans le but d'accroître le débit et la fiabilité (voir la figure 4.5).
- la détermination de la taille optimale des TPDU

- la mise en relation des adresses transport et réseau
- l'identification de la connexion

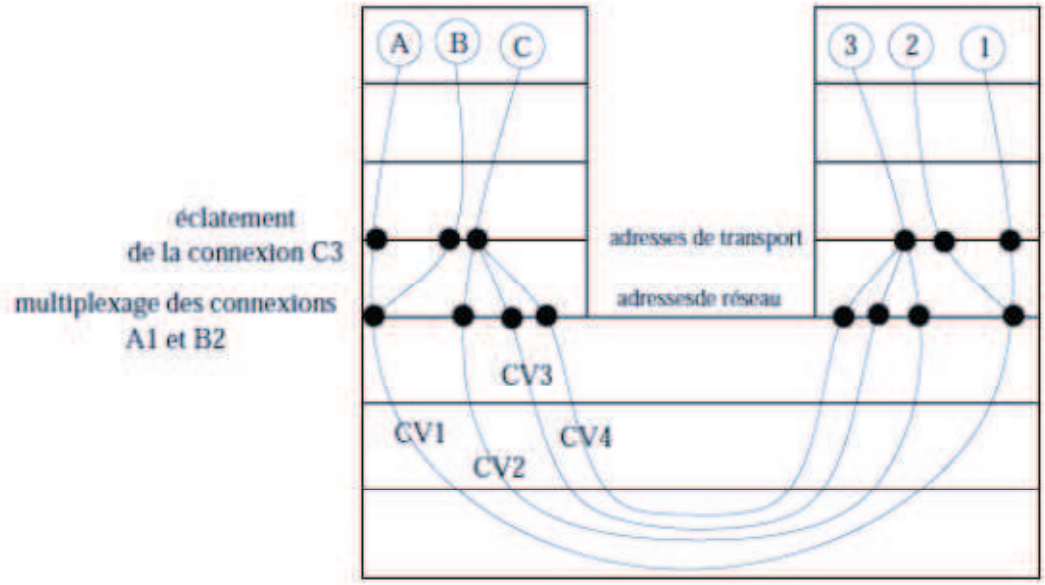


Figure 4.5 - Adressage, multiplexage et éclatement de connexions transport.

Lors de la phase de transfert les données exprès (interruptions, alarmes de petite taille) ne sont pas soumises au contrôle de flux et sont assurées d'être délivrées avant toutes données normales émises après elle. La fragmentation-réassemblage permet, si nécessaire, de découper des TSDU (Transport Service Data Unit) afin de tenir dans un seul TPDU. Ainsi les «lettres» trop grandes sont découpées en fragments de taille fixe (sauf éventuellement le dernier), numérotés et expédiés chacun dans un paquet pour être réassemblés à l'arrivée. La numérotation des paquets permet également de détecter les paquets perdus ou dupliqués. Les erreurs de transmission étant normalement détectées par les couches inférieures, il n'est pas toujours nécessaire de protéger chaque TPDU. Cependant, si la qualité de service réseau est médiocre, on ajoute des mécanismes de détection d'erreur par total de contrôle associé à un mécanisme de retransmission. Le contrôle de flux est fondé sur une fenêtre coulissante de taille variable, similaire à une notion de crédit. Le destinataire impose la cadence de transfert en indiquant à l'expéditeur des crédits d'émission (numéro dans la séquence à ne pas dépasser) qu'il lui transmet dans ses acquittements.
Enfin, la libération de la connexion de transport peut être décidée et déclenchée de manière inconditionnelle par l'un quelconque des correspondants, ce qui peut provoquer des pertes de données. Cette libération est explicite quand il y a échanges d'unités de données spéciales ou implicite quand en fait c'est la connexion réseau sous-jacente qui est libérée.
Pour simplifier le choix des fonctions à mettre en œuvre, l'ISO a défini 5 classes de protocoles chacune étant adaptée à un type de réseau particulier11 et choisi lors de l'établissement de la connexion. L'appelant déclare sa classe préférée et des classes de repli, l'appelé choisi parmi ces propositions ; si aucune classe ne lui convient il refuse la connexion. Succinctement, on trouve

- la classe 0 : elle assure une connexion standard pour les réseaux de type A (établissement négociée de la connexion, libération implicite, transfert de données normales, segmentation et réassemblage des données.
- la classe 1 : pour les réseaux de type B elle ajoute à la classe 0 le transport de données exprès, la reprise sur erreurs signalées, la déconnexion et la reprise sur réinitialisation du réseau.
- la classe 2 : pour les réseaux de type A elle ajoute à la classe 0 le multiplexage et un contrôle de flux amélioré.
- la classe 3 : pour les réseaux de type B elle est constituée des fonctionnalités des classes 1 et 2.
- la classe 4 : pour les réseaux de type C elle correspond à la classe 3 à laquelle on ajoute la détection et la reprise d'erreurs non signalées et l'éclatement de connexions de transport.

Exercices du Chapitre 4 : Couche Transport

Exercice 4.1 :

Supposons que le client A ouvre une session Telnet avec le serveur S. Au même moment, le client B en ouvre également une autre avec le serveur S. Citez quelques numéros d'accès d'origine (source port number) et de destination (destination port number) possibles pour :

1. les segments envoyés de A à S.
2. les segments envoyés de B à S.
3. les segments envoyés de S à A.
4. les segments envoyés de S à B.
5. Si A et B sont deux machines différentes, est-il possible que les segments transmis de A à S aient
le même numéro d'accès d'origine que les segments transmis de B à S?
6. Qu'en est-il si A et B sont une seule et même machine?

Exercice 4.2 :

Une application peut-elle bénéficier d'un service de transfert de données fiable si elle repose sur UDP? Si oui, comment ?

Exercice 4.3 :

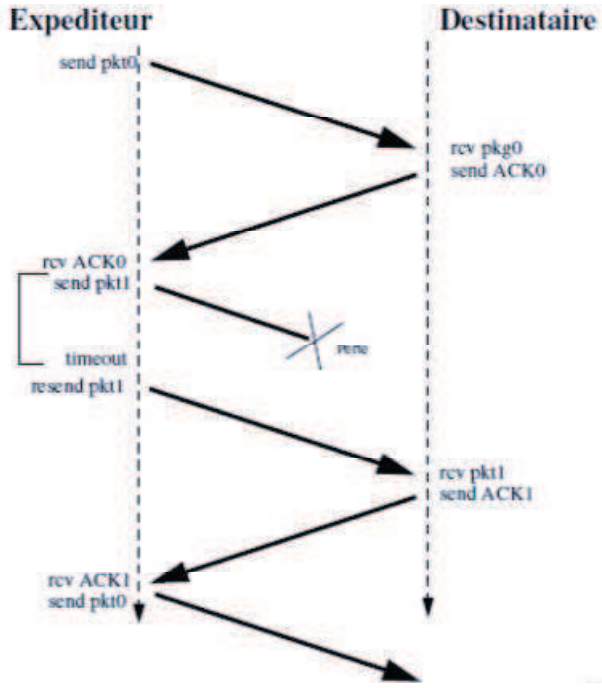
Supposons que le serveur A envoie deux segments TCP directement l'un après l'autre au serveur B sur une connexion TCP. Le premier segment porte le numéro de séquence 90, le second le numéro 110.

(a) Combien d'octets contient le premier segment.

(b) Supposons que seul le second segment arrive à bon port. Quel est le numéro de l'accusé de réception émis par B en réponse à ce segment ?

Exercice 4.4 :

Le diagramme schématise ce qui se passe lors d'un envoie avec perte d'un paquet de données. Faire des diagrammes similaires pour un ACK perdu et une expiration prématurée du temps imparti.



Les couches hautes: session, présentation et application.

Les couches session, présentation et application constituent les couches hautes du modèle OSI et offrent des services orientés vers les utilisateurs alors que les couches basses sont concernées pas la communication fiable de bout en bout. Elles considèrent que la couche transport fournit un canal fiable de communication et ajoutent des caractéristiques supplémentaires pour les applications.

5.1 La couche session.

La couche session fournit aux entités de la couche présentation les moyens d'organiser et synchroniser les dialogues et les échanges de données.
Une session peut par exemple être utilisée pour la connexion à distance d'un terminal à un ordinateur ou pour le transfert d'un fichier et ceci en mode connecté. Bien que très similaires la session et la connexion de transport ne sont pas identiques, les trois cas de la figure 5.1 peuvent se présenter.

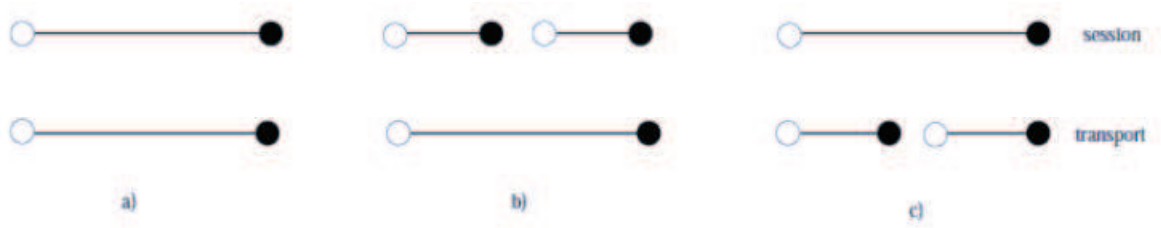


Figure 5.1 - Sessions et connexions de transport.

- Il y a correspondance exacte entre une session et une connexion de transport.
- Plusieurs sessions successives sont établies sur une seule et même connexion de transport. Par exemple, ceci peut être utilisé dans le contexte d'une agence de voyage dans laquelle chaque employé utilise un terminal relié à un ordinateur local. Celui-ci est relié à une base de données centrale de la compagnie pour enregistrer les réservations. Chaque fois qu'un employé veut faire une réservation, une session est ouverte avec l'ordinateur central et elle est fermée lorsque la réservation est terminée. Mais il est inutile de libérer la connexion de transport sous-jacente car celle-ci sera utilisée quelques instants plus tard par un autre employé.
- Plusieurs connexions de transport successives sont nécessaires pour une seule et même session. Ceci peut arriver lorsqu'une connexion de transport tombe en panne, la couche session établit alors une nouvelle connexion de transport de manière à poursuivre la connexion commencée.

Par contre, il n'est pas possible de multiplexer plusieurs sessions sur une même connexion de transport. Celle-ci ne transporte à tout instant qu'au plus une session.
Le transfert des données est régi par les trois phases habituelles : établissement de la session, transfert des données et libération de la session. Les primitives fournies par la couche session sont semblables à celles fournies par la couche transport à la couche session. L'ouverture d'une session nécessite la négociation de plusieurs paramètres entre les utilisateurs des extrémités, certains (qos et existence ou non de données exprès) sont identiques à ceux de la couche transport à qui ils sont passés tels quels. Un autre paramètre peut permettre de décider quelle extrémité aura l'initiative du dialogue dans le cas d'une session bidirectionnelle. Les différences entre transport et session vont se situer au niveau de la libération de la connexion.

Dans le cas du transport, la primitive T_DISCONNECT.request provoque une libération brutale de la connexion avec perte des données en cours de transfert. Une session, elle, sera terminée par la primitive S_RELEASE. request qui opère une libération ordonnée de la connexion, sans perte de données, appelée terminaison négociée. Les différences entre ces deux modes

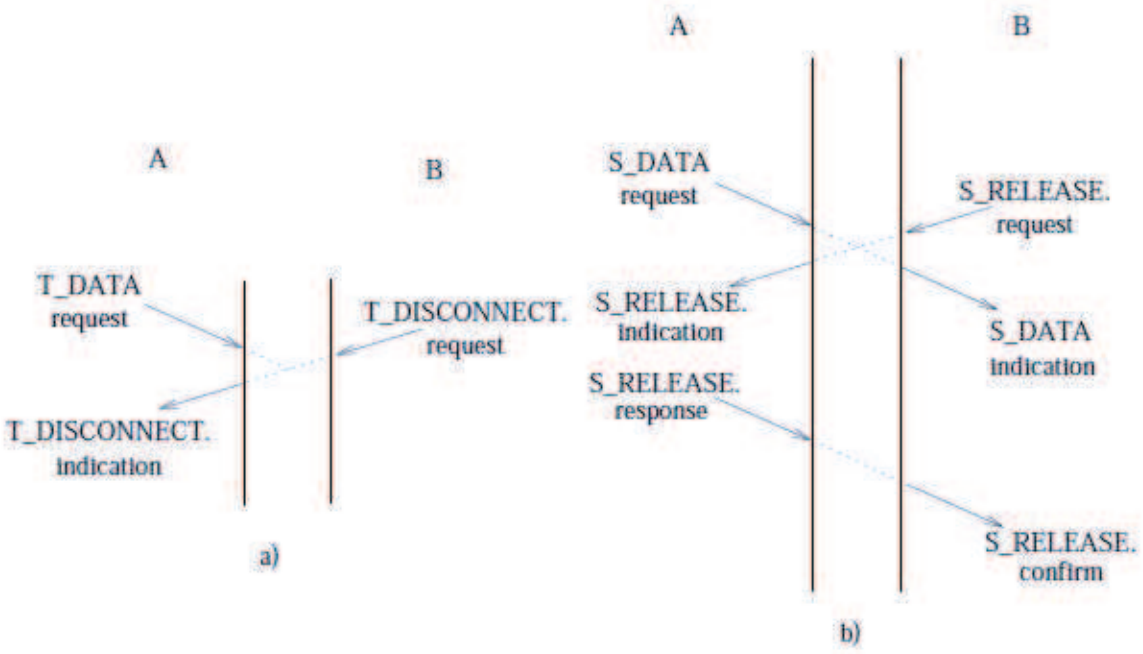


Figure 5.2 - Libération brutale a) et ordonnée b).

de libération sont illustrées dans la figure 5.2. Dans le premier cas, B ne peut plus recevoir le message qui est en transit à partir du moment où il a émis une demande de déconnexion (voir l'automate de la figure 4.4). Dans le deuxième cas, B continue d'accepter des données après avoir émis sa demande de libération jusqu'à ce que A confirme positivement cette déconnexion. Si A veut refuser la déconnexion, il le signale par l'un des paramètres de S_RELEASE.response et il peut ainsi continuer d'envoyer des données, la session se prolonge comme si de rien n'était.

Les sessions peuvent autoriser le dialogue bidirectionnel ou unidirectionnel ou à l'alternat. Ce dernier cas n'est pas forcément dû à des limitations matérielles mais est souvent inhérent au fonctionnement des applications de plus haut niveau. Par exemple, l'accès à une base de données distante est établi sur ce mode. Un utilisateur envoie depuis son terminal une requête à un ordinateur central, puis il attend la réponse de celui-ci. Lorsque l'ordinateur central a envoyé sa réponse, l'utilisateur peut envoyer une autre requête. Ainsi, le mode de fonctionnement à l'alternat est naturel. Dans ce cas il s'agit alors de gérer à qui c'est le tour d'envoyer des données. Ce service est mis en œuvre grâce à un jeton de données (data token). Seule l'extrémité qui possède le jeton peut envoyer des données, l'autre doit rester silencieuse. La négociation initiale à l'établissement de la connexion de session fixe quelle extrémité reçoit en premier le jeton. Lorsque le possesseur du jeton a fini d'expédier ses données il peut passer le jeton à son correspondant à l'aide de la primitive S_TOKEN_GIVE. request. Si l'extrémité qui ne possède pas le jeton veut expédier des données il peut le

demander, poliment, à l'aide de la primitive S_TOKEN_PLEASE. request. Le possesseur du jeton peut alors le lui envoyer ou lui refuser.
La synchronisation est également l'un des services de la couche session et consiste à placer les entités de session dans un état connu des deux interlocuteurs en cas d'erreur. Il ne s'agit pas ici de régler des erreurs qui seraient dûes à un problème de transmission de données puisque la couche transport doit les régler. Mais celle-ci ne peut effectuer des reprises sur erreurs dues aux couches supérieures puisqu'elle n'en a pas connaissance. Par exemple, si lors d'un transfert de fichiers très long une erreur se produit du côté du récepteur mais en dehors de la transmission proprement dite (problème d'accès à un disque par exemple), il faut que la couche session soit capable de s'en apercevoir et qu'elle évite de relancer simplement tout le transfert de fichiers. C'est pourquoi, grâce à des points de synchronisation dans le flot d'échanges de données, elle reprendra les transferts au niveau du dernier point de reprise correct. Ces points consistent à découper le flot de données à transmettre en pages et évidemment, le mécanisme nécessite que l'utilisateur de la session émetteur soit capable de mémoriser suffisamment longtemps les pages envoyées pour pouvoir en réexpédier certaines si nécessaires.

5.2 La couche présentation.

La couche présentation s'occupe de la syntaxe et de la sémantique des informations transportées en se chargeant notamment de la représentation des données.
Par exemple, sur un ordinateur à base d'un processeur de la famille des 68 000 les entiers sont représentés avec les bits de poids fort à gauche et ceux de poids faible à droite. Or, c'est l'inverse sur un ordinateur basé sur un processeur de la famille du 80x86. Cette difficulté sera prise en compte par la couche présentation qui effectuera les opérations nécessaires à la communication correcte entre ces deux familles de machines. Pour ce faire l'ISO a défini une norme appelée syntaxe abstraite numéro 1 (Abstract Syntax Notation 1) permettant de définir une sorte de langage commun (une syntaxe de transfert) dans lequel toutes les applications représentent leurs données avant de les transmettre.
C'est aussi à ce niveau de la couche présentation que peuvent être implantées des techniques de compression (code de Huffman par exemple) et de chiffrement de données (RSA, DSE, etc...).

5.3 La couche application.

5.3.1 Définition.

La couche application donne au processus d'application le moyen d'accéder à l'environnement OSI et fournit tous les services directement utilisables par l'application, à savoir:

- le transfert d'informations
- l'allocation de ressources
- l'intégrité et la cohérence des données accédées
- la synchronisation des applications coopérantes

En fait, la couche application gère les programmes de l'utilisateur et définit des standards pour que les différents logiciels commercialisés adoptent les mêmes principes, comme par exemple :

- Notion de fichier virtuel représenté sous forme d'arbre pour les applications de transfert de fichiers, opérations permises sur un fichier, accès concurrentiels, ...

- Découpage des fonctions d'une application de courrier électronique qui se compose d'un le contenu (en-tête et corps) et d'une enveloppe. Une fonctionnalité de l'application gère le contenu et une autre le transfert en interprétant l'enveloppe

5.3.2 Rôle de la couche Application.

La couche Application est la dernière couche du modèle OSI (Niveau 7). Elle regroupe les services qui traitent des aspects sémantiques de l'Application dans un système réparti. Les autres aspects (syntaxiques et de gestion de dialogue) sont définis respectivement dans les couches Présentation et Session. Les trois couches sont étroitement liées.

C'est l'interface entre les processus utilisateurs et le monde OSI. Les organismes de normalisation ont défini des fonctions génériques à différentes applications et des foncions spécifiques. Un ensemble d'éléments de services ont vu le jour dont la combinaison et la coordination permettent la conception d'applications dans un environnement réparti. Une structure modulaire (et non hiérarchique) de la couche Application a été définie: ALS (Application Layer Structure - ISO 9545).

La normalisation ne concerne que la communication entre entités d'application et en aucun cas la réalisation réelle des opérations de l'application. Ainsi dans la normalisation d'un transfert de fichiers, la lecture réelle ou l'écriture réelle des données sur disque n'est pas du ressort de la couche Application.

5.3.3 Les éléments constitutifs de la couche application.

Les fonctions mises en oeuvre dans la couche Application sont regroupées logiquement dans des Eléments de Service d'Application: ASE (Application Service Element). Leur nombre n'est pas limité puisqu'elles dépendent des applications à mettre en œuvre (voir la figure 5.3).

On distingue:

- Les ASE communs: CASE (Common Application Service Element) – Ce sont des éléments utilisables par plusieurs types d'applications (ex. ASE permettant la synchronisation de traitement entre plusieurs machines).

- Les ASE spécifiques: SASE (Specific Application Service Element) - Ce sont des éléments spécifiques à des applications (ex. ASE liés à la messagerie)

Les ASE sont combinés pour former une Entité d'Application: AE (Application Entity). L'AE utilise les services de la couche présentation à travers un ou plusieurs PSAPs. Un élément spécifique de l'AE appelé UE (User Element) permet au Processus d'application AP (Application Process) d'accéder aux services des ASE. L'AE est un objet statique rassemblant les fonctions et les données relatives à la communication. L'instanciation d'une AE est appelée AEI (Application Entity Invocation). Chaque AEI possède une information d'état .

Les AP sont les processus de l'application. L'activité d'un AP est représentée par une API (Application Process Invocation). Certains APs sont locaux et d'autres sont OSI (coopérant dans le système réparti). Une ou plusieurs AE peuvent être mises en oeuvre par un AP OSI. Ce sont ces AE qui représentent la partie communication de l'application.

L'ensemble des ASE ainsi que les règles d'interaction entre ASE et avec la couche présentation sont spécifiés par le Contexte d'Application (Application Context). Ce sont les Contextes d'Application qui son normalisés. Une AE peut avoir plusieurs contextes dont un seul est actif à un instant donné. Un ASE peut appartenir à plusieurs contextes.

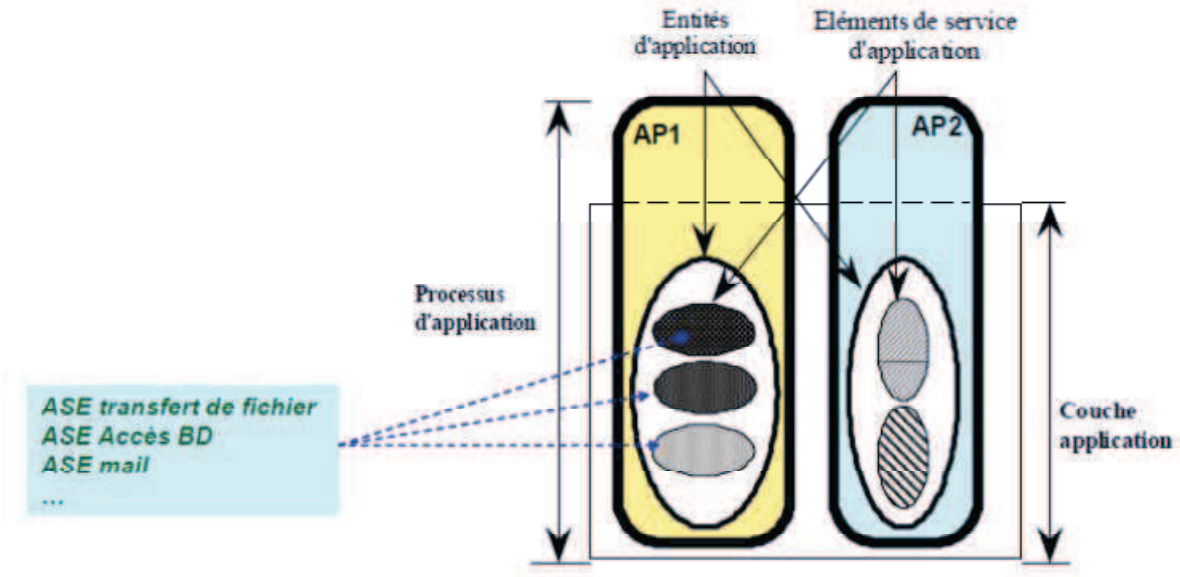


Figure 5.3 – Les éléments constitutifs de la couche application.

La relation entre entités d'application est appelée Association d'Application AA (Application Association). Attention, on ne parle pas à ce niveau de connexion. Un CASE sert à la gestion de l'association: ACSE (Association Control Service Element). Une fonction de contrôle d'association selon les règles du contexte d'application est mise en oeuvre: la SACF (Single Association Control Function). L'ensemble SACF, AA, ACSE et ASEs forment un objet d'association simple SAO (Single Association Object). Quand une application a besoin de communiquer avec plusieurs autres applications dont les associations doivent être coordonnées, une fonction MACF (Multiple Association Control Function) est mise en oeuvre.

Les AEI, API et AA sont identifiés par des identificateurs: AEII, APII et AAI.

Application	Protocole d'application	Protocole de transport
e-mail	SMTP [RFC 821]	TCP
Accès distant à un terminal	TELNET [RFC 854]	TCP
Web	HTTP [RFC 2068]	TCP
Transfert de fichier	FTP [RFC 959]	TCP
Streaming multimédia	Propriétaire (e.g. RealNetworks)	TCP ou UDP
Serveur de fichier distant	NFS	TCP ou UDP
Téléphonie sous IP	Propriétaire (e.g., Vocaltec, Skype)	Typiquement UDP

Table 5.1 – Exemples d'applications & les protocoles associés.

Exercices du Chapitre 5 : Couches Hautes (Session, Présentation et Application)

Exercice 5.1 :
Peut-on multiplexer plusieurs sessions sur une seule connexion transport ?

Exercice 5.2 :
En cas de fermeture brutale quelles sont les primitives qui entre en jeux afin de mieux reprendre la communication interrompue ?

Exercice 5.3 :
Quel est l'intérêt de la compression dans la couche présentation ? Expliquer son principe en prenant la suite des 1 comme base de compression ?

Exercice 5.4 :
Donner un exemple d'un processus d'application ? Citer ces ASE associés ?

Exercice 5.5 :
Parmi les couches du modèle de référence OSI, quelles sont les couches qui dépendent du système d'exploitation?

Le protocole TCP/IP.

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.

6.1 TCP/IP et le modèle OSI.

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(voir la figure 6.1), et obtenir ainsi un modèle en 4 couches.

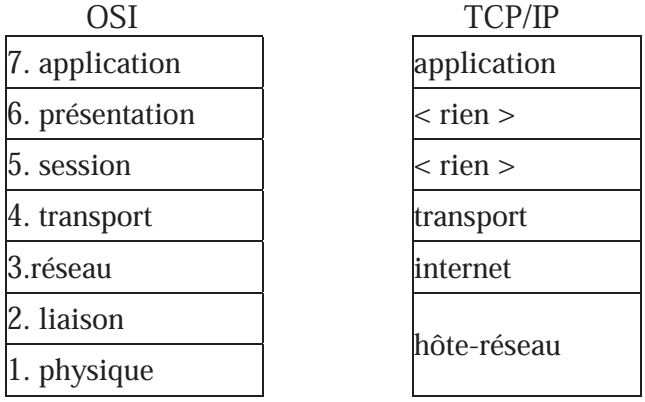


Figure 6.1 - Représentation du modèle TCP/IP.

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.

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, ...).

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.

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).

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.

6.2 Suite de protocoles.
Le modèle TCP/IP correspond donc à une suite de protocoles (voir la figure 6.2) 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.

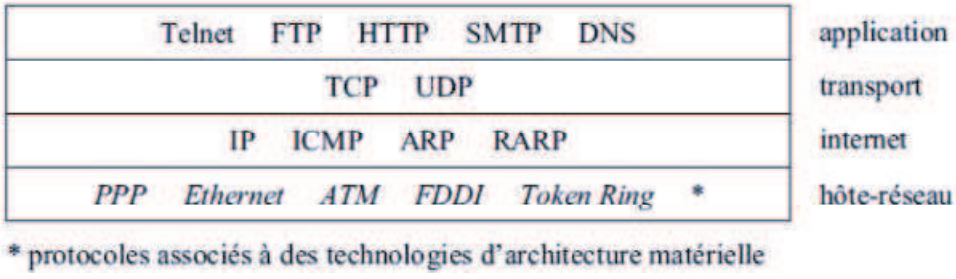


Figure 6.2 - Suite de protocoles du modèle TCP/IP.

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.

6.3 Le protocole IP.

Définitions.

Le protocole IP (Internet Protocol) , 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.

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.

6.3.1 L'adressage IP.

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).

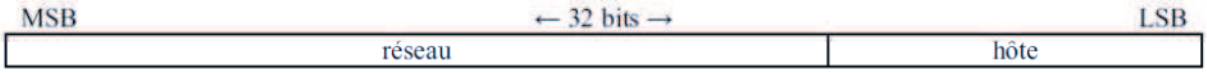


Figure 6.3 - Position des identifiants de réseau et d'hôte dans une adresse IPv4.

Ainsi, des hôtes possédant le même identifiant de réseau peuvent communiquer directement entre eux, 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 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

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.

Nb : Le modèle IPv4 arrive à bout de souffle, en effet, pouvant coder « seulement » 2^{32} adresses différentes ($\approx 4{,}3.10^{9}$), 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.
IPv4 se voit donc progressivement remplacé par le modèle IPv6, codé sur 16 octets, et ayant donc un potentiel de 2^{128} adresses différentes ($\approx 3{,}4.10^{38}$), 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.).

6.3.2 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 :

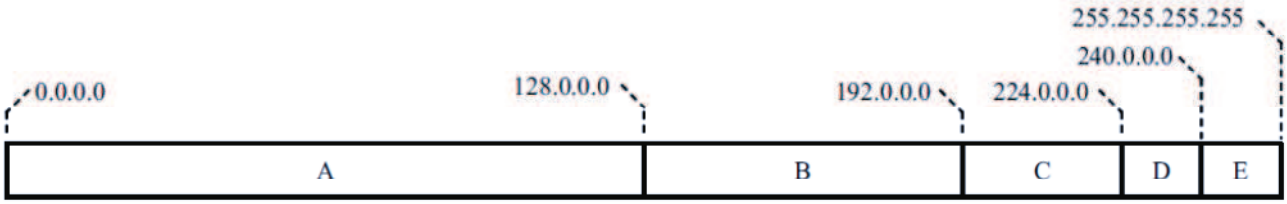


Figure 6.4 - Classes d'adresses IPv4.

Classe	octet de poids	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 .
Aux classes d'adresses A, B et C (unicast) on associe un masque différent, qui est généralement standard mais peut être modifié 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, 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 :

- 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 (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 (loopback) ;
- 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;
- adresse IP 255.255.255.255 : adresse de diffusion limitée (limited broadcast), s'adressant à tout nœud connecté au même segment (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 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

6.3.3 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.

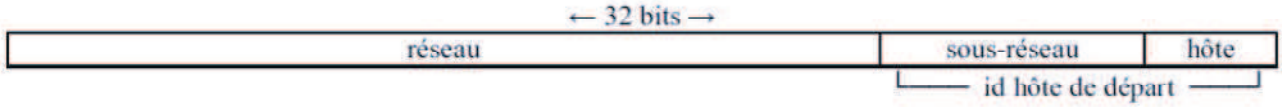


Figure 6.5 - Position des identifiants de réseau, de sous-réseau et d'hôte dans une adresse IPv4.

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.

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.

6.3.4 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 datagramme IP (PDU) : 20 - 65535 octets ;
(taille minimale conseillée de 576, et taille maximale de 65535 (64 ko) jamais atteinte)

- 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.

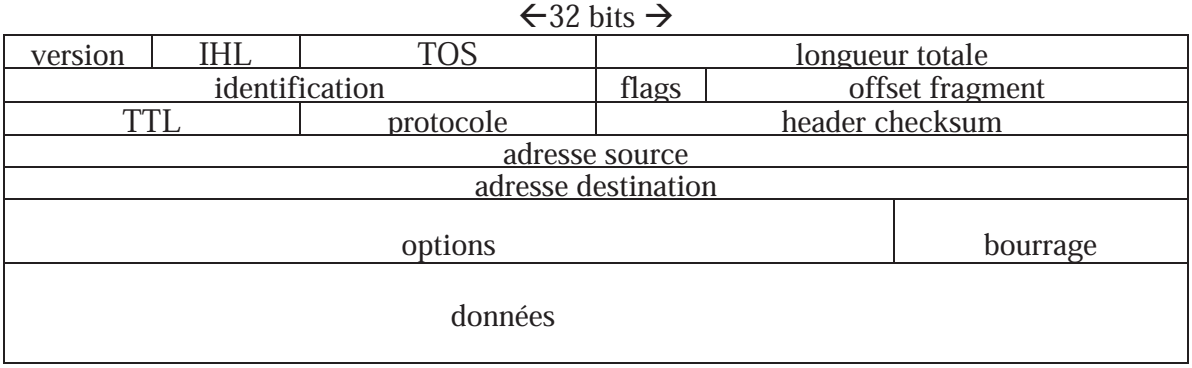


Figure 6.6 - Datagramme IPv4.

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

6.3.5 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. La taille du fragment est bien évidemment choisie la plus grande possible, l'unité étant le mot de 64 bits (8 octets).
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.

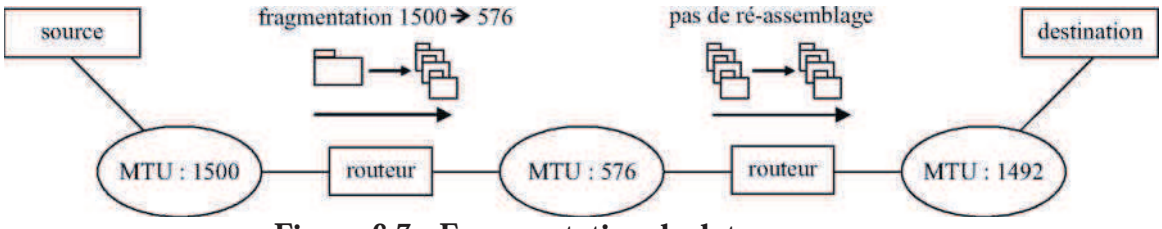


Figure 6.7 - Fragmentation de datagrammes.

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 est alors envoyé à l'émetteur pour lui signifier l'erreur de transmission.

6.4 Les protocoles TCP/UDP.

6.4.1 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 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.

6.4.2 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:

- 0 - 1023 : ports reconnus, 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, 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.

6.5 Le protocole TCP.

6.5.1 Définitions.

Le protocole TCP (Transmission Control Protocol) 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 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 :

- 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;
  - 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.
  -

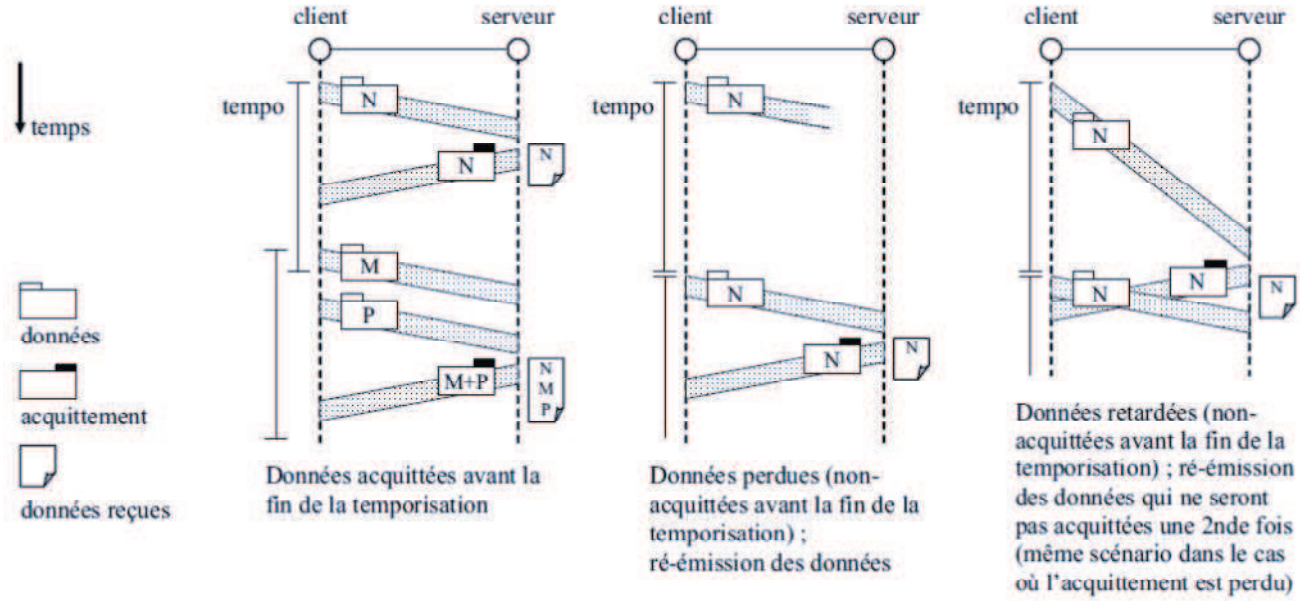


Figure 6.8 - Principe et usage de l'acquittement.

- 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, utilisé dans de nombreuses applications 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.

6.5.2 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.

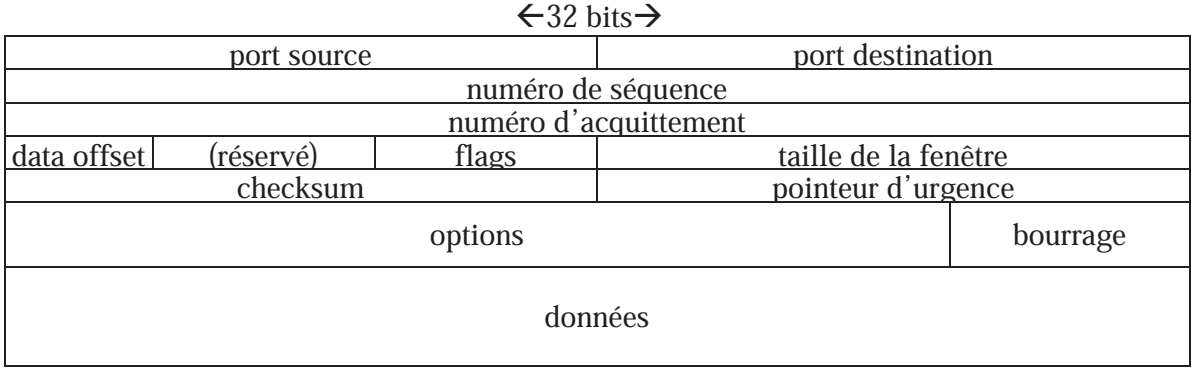


Figure 6.9 - Segment TCP.

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 entête 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.

6.5.3 Gestion d'une connexion TCP.

Le protocole TCP est orienté connexion suivant un mécanisme de « poignée de main » en trois temps 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.

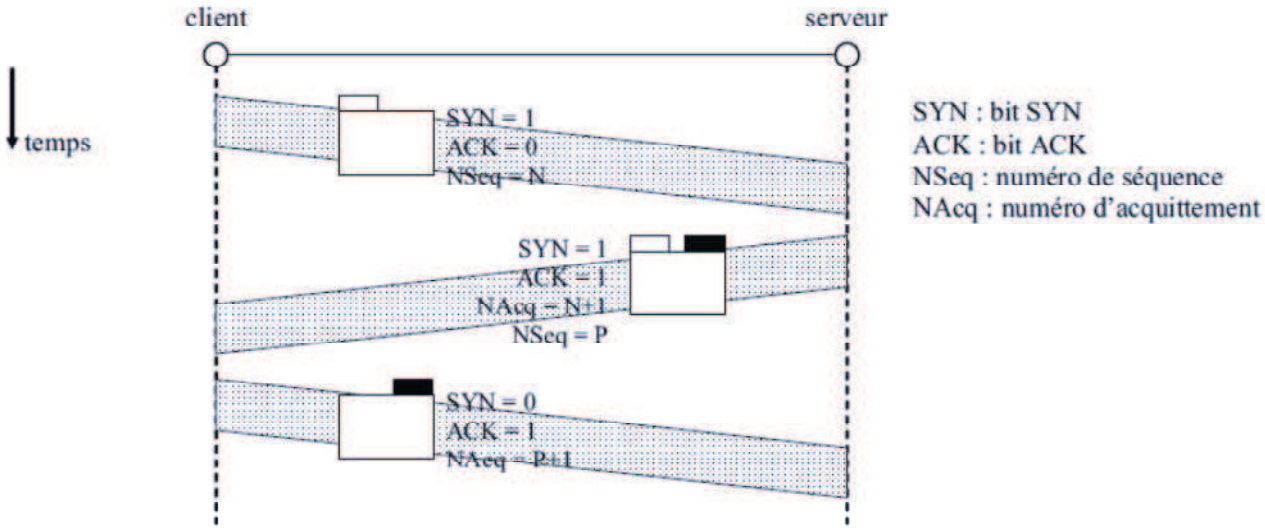


Figure 6.10 - Etablissement d'une connexion TCP.

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.

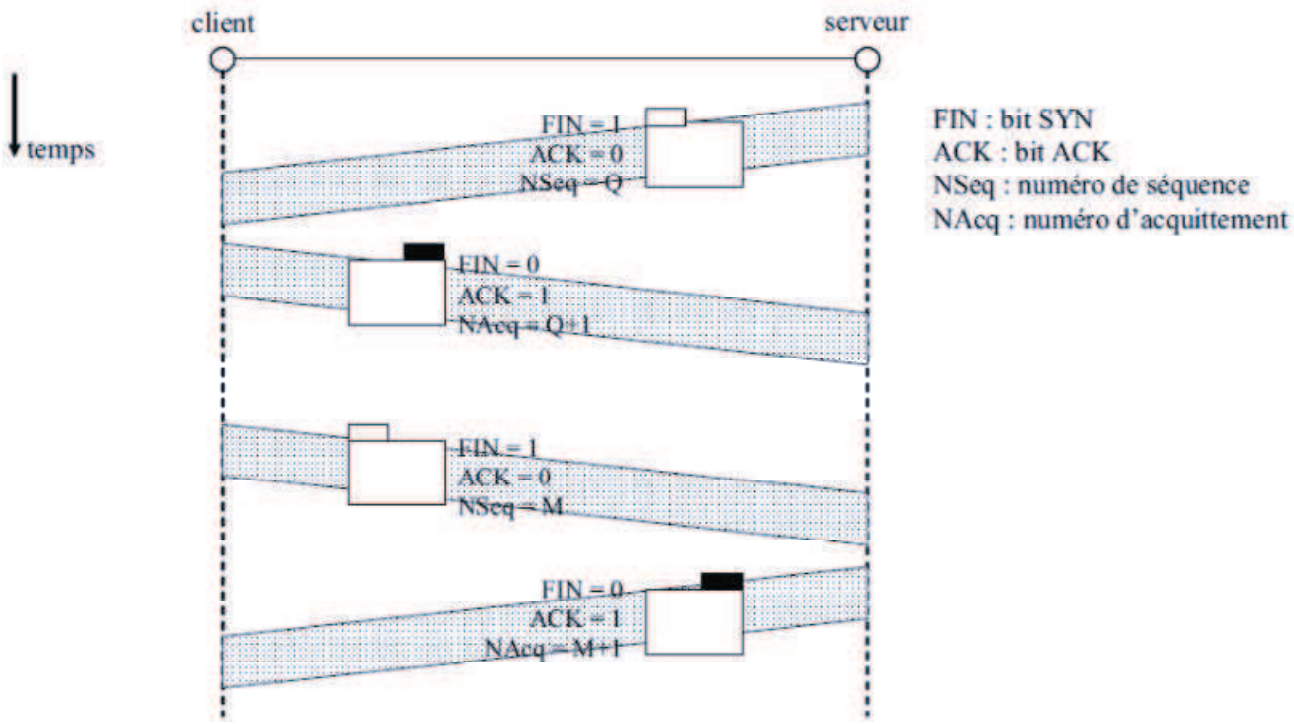


Figure 6.11 - Fermeture d'une connexion TCP.

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.

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

6.5.4 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.

6.6 Le protocole UDP.

6.6.1 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 mais extrêmement rapide, dédiant UDP à un usage spécifique aux transmissions de données de très faible volume, devant être transmises rapidement, ou étant peu sensibles à un faible taux d'erreur. 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.
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.

6.6.2 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.

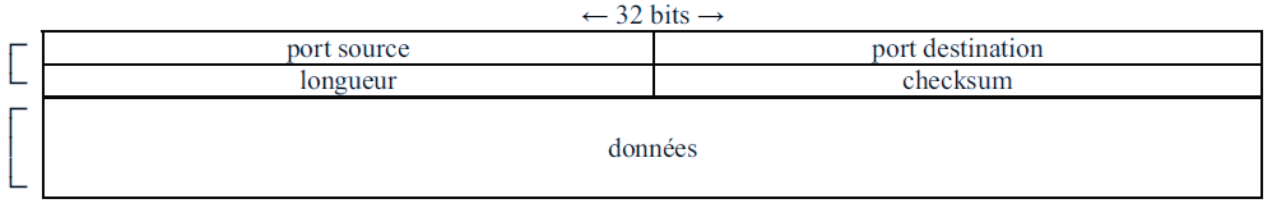


Figure 6.12 - Datagramme UDP.

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.

Exercices du Chapitre 6 : Le Modèle TCP/IP

Exercice 6.1 :

Soient 03 réseaux A B et C connectés par un routeur (Passerelle), utilisant le protocole TCP/IP de classe C avec un masque de sous réseaux par défaut.
Soient quatre machines suivantes :

Machine 1 du réseaux A identifiée par un IP : 192.44.77.70
Machine 2 du réseaux B identifiée par un IP : 192.44.76.80
Machine 3 du réseaux C identifiée par un IP : 192.44.77.90
Machine 4 du réseaux B identifiée par un IP : 192.44.77.81

Soient les tentatives suivantes :

La machine 1 veut joindre la machine 3
La machine 1 veut joindre la machine 2
La machine 2 veut joindre la machine 4

_ Ces tentatives sont-ils possibles ? expliquez ?
_ Proposez d'autres solutions ?

Exercice 6.2 :

Une organisation dispose d'un réseau avec un adressage de classe B.
- Donner son masque de s/réseau par défaut
- Donner le masque de s/réseau dans le cas ou les réseaux suivants 144.144.208.0, 144.144.144.0 font partie de cette organisation
- Quel est le nombre maximum de machines que peut inclure chacun des ces sous réseaux
- Indiquer les adresses : sous réseaux, diffusion, début et fin de chaque sous réseaux (choisir deux s/réseaux)

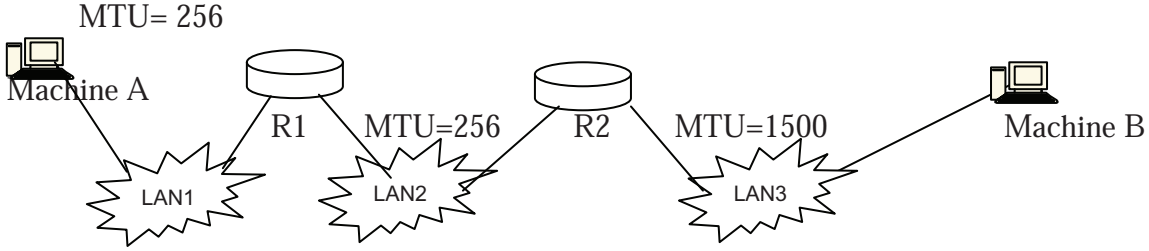
Exercice 6.3 :

Une organisation dispose d'un réseau d'adresse 192.153.0.0 et désire organiser son réseau en six (06) sous réseaux distincts. Indiquez :
- quel est le nombre maximum de machines que peut inclure chacun des ces sous réseaux ?
- le masque de sous réseaux
- les adresses : sous réseaux, diffusion, début et fin de chaque sous réseaux.

Exercice 6.4 :

Décrire l'entête du paquet IP version 04 ? Peut-on l'améliorer ?

Exercice 6.5 :



- Indiquez les fragments qui seront générés en envoyant un message de 715 octets d'une machine A du réseau LAN1 vers une autre machine B du réseau LAN3, via une connexion TCP (entête =20 octet).
Pour chaque fragment, il faut préciser les informations changeables de l'entête IP.

Exercice 6.6 :

Après la mise en place du protocole TCP/IP dans une entreprise constituée de plusieurs réseaux et de routeurs qui assurent le routage, sachant que le datagramme IP est contrôlé par le code CRC avec un code Générateur fixé à l'avance. Expliquer pourquoi à chaque réception le datagramme est rejeté après un contrôle de CRC (même dans le cas ou la donnée est correctement reçue) .

Références

1. Les Réseaux. Guy Pujolle, Collection Eyrolles, 7e ed., 2011.
2. Cours des Réseaux. Pascal Nicolas. Maitrise d'informatique-Université Angers, UFR Sciences de l'Université d'Angers, 2000.
3. Les réseaux, principes fondamentaux. Patrice Rolin, et al. Hermès.
4. TCP/IP, architectures, protocoles et applications. Douglas Corner, Interéditions, 5ed., 2009.
5. TCP/IP illustré - Vol 1,2,3. W. Richard Stevens, International Thomson Publishing, France, ed. Vuiberd, 1998.
6. Les réseaux informatiques. Dominique Lalot, faculté d'Aix en provence, 2000. http://www.httr.ups-tlse.fr/pedagogie/cours/
7. Cours de réseaux. Bruno Péan, EISTI Cergy Pontoise, 2001.