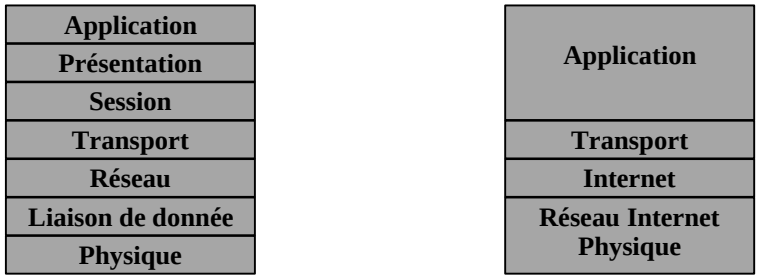
Introduction à TCP/IP

Sommaire

1. Rappel sur le modèle OSI

Tous les applicatifs réseaux doivent pouvoir communiquer quelque soit l'architecture ou la plate forme utilisée. Pour cela les opérations sur les réseaux ont été divisées en plusieurs phases de manières à simplifier le portage des applicatifs sur toute la plat-forme. C'est ce qu'on appelle le modèle en couche. Un standard alors a été créé, normalisé par l'**O**pen **S**ystem **I**nterconnection **OSI** utilisant sept couches distinctes.

L'architecture TCP/IP est similaire à ce modèle en couche, mais ne dispose que de quatre couches dans la plupart des cas.



Modèle de référence OSI Modèle TCP/IP

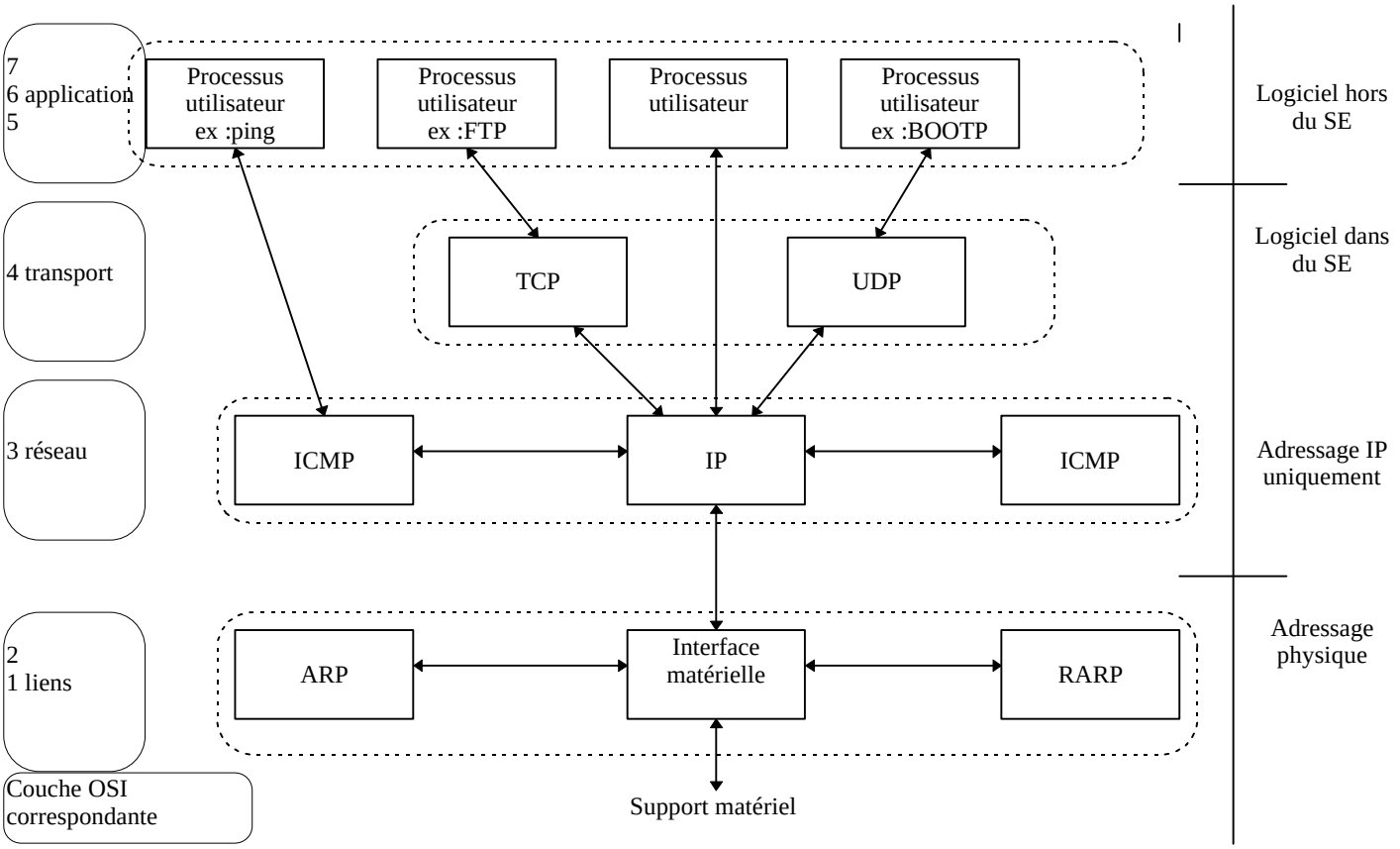
Le modèle TCP/IP ne suit pas tout à fait l'architecture en couche du modèle OSI. Après expérimentation on s'est aperçu qu'une carte réseau devait regrouper les couches 1 et 2 pour obtenir des performances correctes.

Toutefois, il existe quelque cas où les couches 1 et 2 sont différenciées dans le modèle TCP/IP. C'est le cas d'une connexion par modem qui comporte donc des couches de liaison de données.

Remarque : Dans le modèle TCP/IP, la couche utilise soit TCP (Transmission Control Protocol), soit UDP (User Datagram Protocol). Par contre il n'existe qu'un seul protocole de niveau réseau : IP (Internet Protocol).

2. Architecture des protocoles TCP/IP

Les logiciels TCP/IP sont structurés en quatre couches de protocoles qui s'appuient sur une couche matérielle.



2.1. Couches de liens

La couche de liens est l'interface avec le réseau et est constitué d'un driver du système d'exploitation et d'une carte d'interface de la station avec le réseau.

2.2. Couche réseau

La réseau ou la couche IP (Internet Protocol) gère la circulation des paquets à travers le réseau en assurant leur routage. Elle comprend aussi les protocoles ICMP (Internet Control Message Protocol) et IGMP (Internet Group Message Protocol).

2.3. Couche Transport

La couche transport assure tout d'abord une communication de bout en bout en faisant abstraction des machines intermédiaires entre l'émetteur et le destinataire. Elle s'occupe de réguler le flux de données et assure un transport fiable (données transmises sans erreurs et reçues dans l'ordre de leur émission de la cas de TCP) ou non fiable (dans la cas de UDP). Pour UDP il n'est pas garanti qu'un paquet (appelé dans ce datagramme) arrive à bon port, c'est à la couche application de s'en assurer.

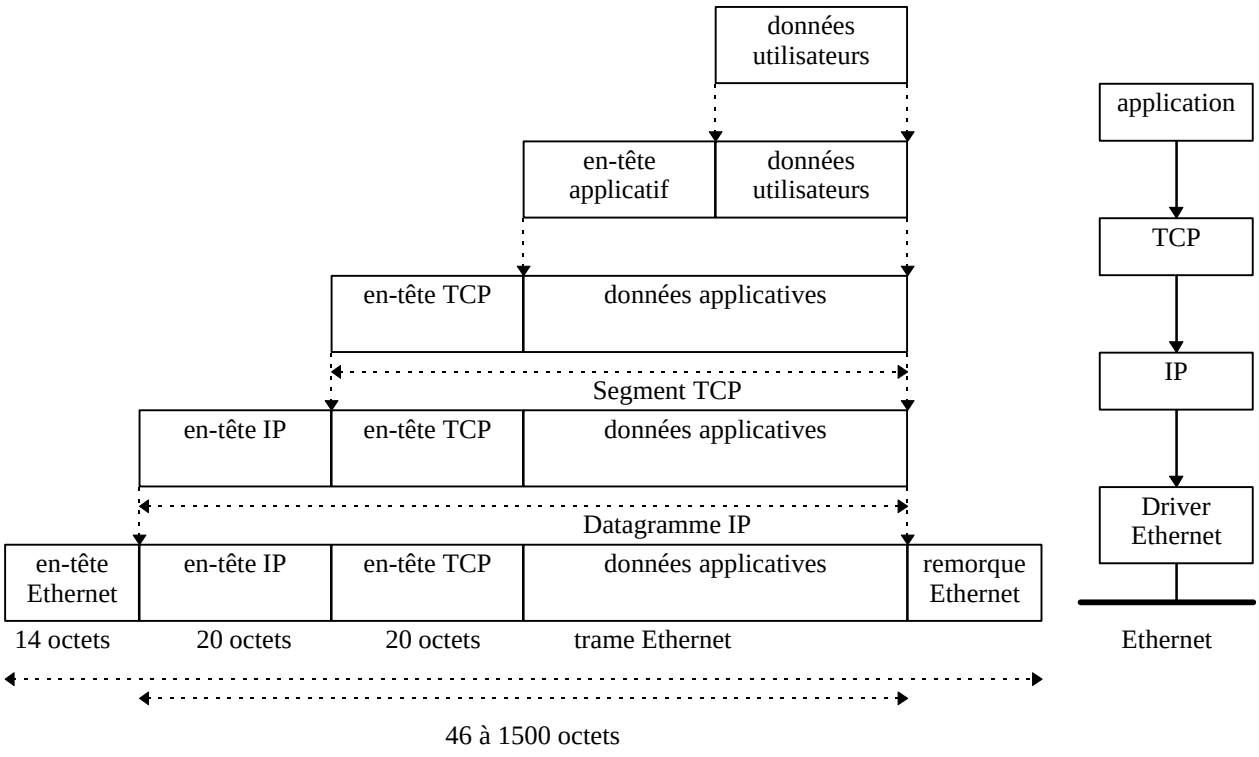
2.4. Couche application

C'est la couche des programmes utilisateurs comme Telnet (connexion à un ordinateur distant), FTP (File Transfert Protocol) ou SMTP (Simple Mail Transfert Protocol).

Cette architecture et ses différents protocoles permettent de faire fonctionner un réseau local, ceci permet surtout de constituer un Internet, c'est à dire, une interconnexion de réseaux éventuellement hétérogènes.

Lorsqu'une application envoie des données à l'aide de TCP/IP, les données traversent de haut en bas chaque couche jusqu'à aboutir au support physique où elles sont alors émises sous forme de suite de bits.

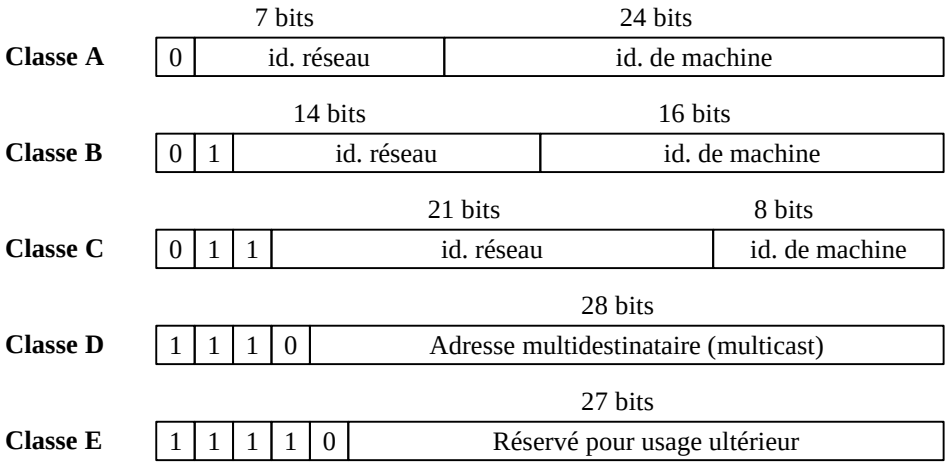
L'encapsulation consiste pour chaque couche à ajouter de l'information aux données en les commençants par des en-têtes, voir en ajoutant des informations remorque.



2.5. Adresssage

Chaque station du réseau Internet dispose d'une adresse IP unique codée sur 32 bits, plus précisément chaque interface dispose d'une adresse particulière. Une adresse IP est toujours représentée dans une notation décimale pointée constituée de 4 nombres (1 par octet) chacun compris entre 0 et 255 et séparés par un point.

Une adresse IP est plus précisément constituée d'une paire (id. de réseau, id. de machine) et appartient à une certaine classe (A, B, C, D ou E) selon la valeur de son premier octet.



Le tableau ci-après donne l'espace d'adresses possibles pour caque classe.

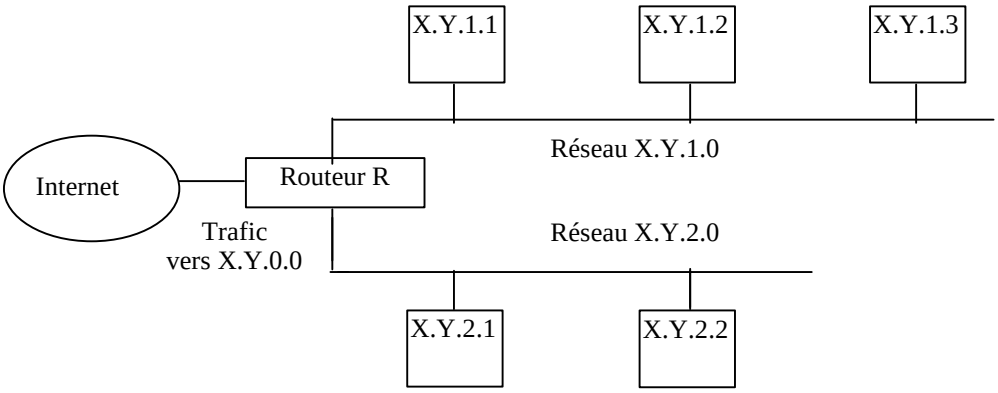
Classes	Adresses
A	**0.0.0.0 à 127.255.255.255**
B	**125.0.0.0 à 191.255.255.255**
C	**192.0.0.0 à 223.255.255.255**
D	**224.0.0.0 à 239.255.255.255**
E	**240.0.0.0 à 247.255.255.255**

Ainsi les adresses de classe A sont utilisées pour les très grands réseaux qui comporte plus de 2^{16} = 65536 ordinateurs. Les adresses de classes B sont utilisées pour les réseaux ayant entre 2^8 = 256 et 2^{16} = 65536. Seuls 256 machines sont possibles sur un réseau classe C dont le nombre possible dépasse les 2 millions.

Le système d'adresse IP permet également la définition d'adresses de sous réseaux en découpant la partie réservée à l'adresse des machines sur un réseau en deux parties dont la première sera un identificateur de sous réseau. Ainsi un seul réseau de classe B, sur lequel o, pourrait nommer 65536 machines, pourra être décomposé en 254 sous réseaux de 254 machines, de la manière décrite ci-dessous :

<Id. de réseau sur 16 bits>.<id. de sous réseau sur 8 bits>.<id. de machine sur 8 bits>.

On peut également adopter le même principe pour un réseau de classe C. Cette technique a pour effet de provoquer un routage hiérarchique.



Pour tout le reste d'Internet, il n'existe qu'un seul réseau X.Y.0.0 et tous les routeurs traitent les datagrammes à destination de ce réseau de même façon. Par contre, le routeur R se sert du troisième octet (égal 1 ou 2) de l'adresse contenue dans les datagrammes.

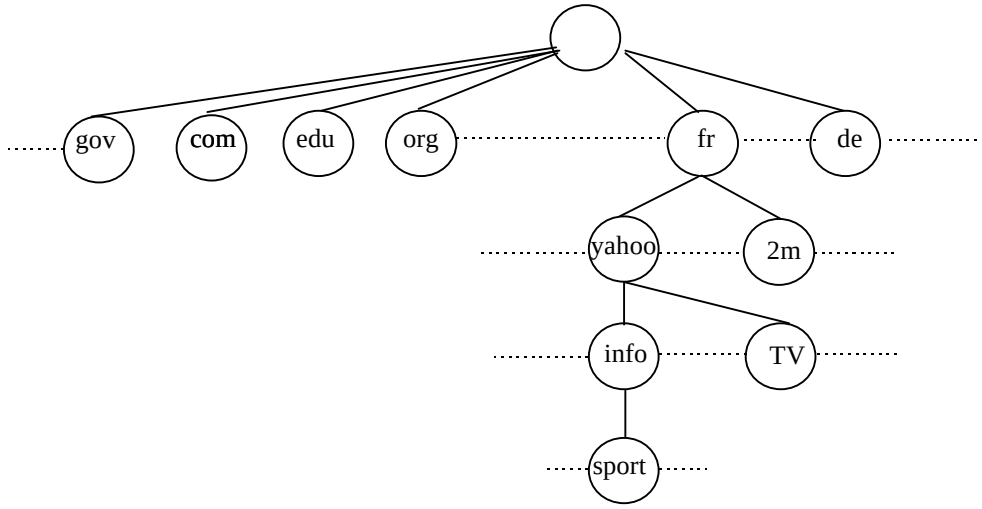
Outre l'adresse IP, une machine doit également connaître le nombre de bits attribués à l'identificateur du sous réseau et celui de la machine. Cette information est rendue possible grâce à un masque de sous réseau qui est un mot de 32 bits contenant des bits à 1 au lieu et place de l'identificateur réseau et de sous réseau et de bits à 0 au lieu et place de l'identificateur de machine. Ainsi le masque 255.255.255.0 indique que les 24 premiers bits d'une adresse désignent le sous réseau et les 8 derniers une machine.

Devant la pénurie d'adresse de classe B et l'exploitation des tables de routage le système CIDR (Classless Inter Domain Routing RFC 1518,1519) est apparu principalement dans le but d'agréger des réseaux. Cette agrégation se fait par région géographique et fournisseur d'accès. Ce système de sur réseau permet ainsi de faire apparaître dans les tables de routages plusieurs réseaux sous le même identifiant. Cependant les réseaux agrégés doivent des adresses contiguës de manière à avoir des préfixes identiques. Par exemple 193.127.32.0 et 193.127.33.0 peuvent être agrégés sous la notation 193.127.32.0/23. Le nombre 23 est le masque signifiant que les 23 bits de poids fort représentent l'adresse du dur réseau.

De cette manière si une société a besoin de 100.000 adresses on lui fournira un part du réseau de classe A en l'associant un masque de 15 bits. Ainsi il disposera de $2^{(32-15)}$ = 131072 adresses.

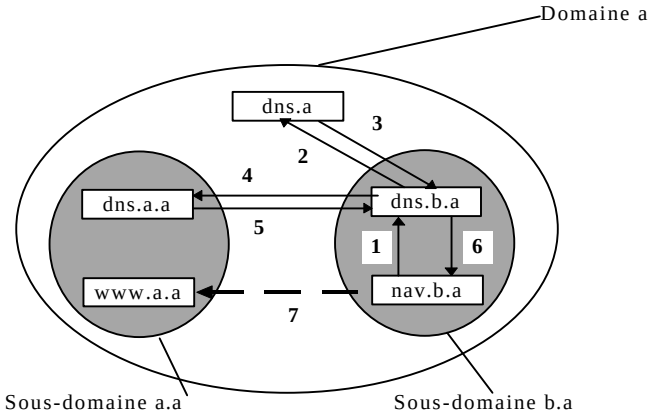
2.6. Nommage

Il est préférable pour un humain de désigner une machine par un nom explicite, afin de satisfaire cette tâche, le système de nom de domaine (DNS :Domain Name System) a été mis en place.



En fait le DNS est un espace de nom hiérarchisé. Chaque noeud d'au plus 63 caractères et la racine de l'arbre a un nom nul. Le nom de domaine d'un nom est la concaténation de son nom avec celui de ses ancêtres de l'arbre.

Voici un exemple qui montre la résolution du nom du serveur web www.a.a lorsque le navigateur sur la machine www.b.a cherche à joindre ce site.



1. Le navigateur envoie à son DNS dns b.a une requête de résolution pour le nom www.a.a.

2. dns.b.a ne connaissant pas cette adresse car elle ne dépend pas de la zone et qu'il ne l'a pas dans son cache, transmet cette adresse à dns.a puisque c'est le dns d'autorité de niveau supérieur qu'il connaît.

3. dns.a ne connaissant pas cette adresse car elle ne dépend pas de la zone et qu'il ne l'a pas dans son cache, transmet cette adresse à dns.a puisque c'est le dns d'autorité de niveau supérieur qu'il connaît.

4. Le serveur dns.b.a sa requête vers ce nouveau DNS dns.a.a.

5. Le serveur dns.a.a connaît l'adresse demandée car la machine www.a.a appartient à sa zone et peut donc envoyer l'adresse IP demandée à dns.b.a.

6. Le DNS mémorise dans son cache la réponse et la retourne au demandeur initial : le navigateur.

3. Réseau Ethernet

3.1. Description générale

Ethernet est le nom donné à une des technologies les plus utilisées pour les réseaux locaux en bus. Majoritairement, les réseaux Ethernet ont un débit de 10 Mbits/s et les informations sont transmises sur le bus sans garantie de remise. Chaque transceiver capte toutes les trames qui sont émises sur le câble et les redirige vers le contrôleur de la station qui rejettera les trames qui ne lui sont pas destinées et enverra au processeur celles qui le concerne, c'est à dire celles dont l'adresse de destination est celle de la carte réseau. Comme il n'y a pas d'autorité centrale qui gère l'accès au câble, il est possible que plusieurs stations veuillent émettre simultanément sur le câble. C'est pourquoi chaque transceiver écoute le câble pendant qu'il émet des données afin de détecter des éventuelles perturbations. Si une collection est détectée par le transceiver celui ci prévient le coupleur qui arrête d'émettre et attends un laps aléatoire compris entre 0 et une certaine durée δ avant de réemettre ses données.

Si il y a encore un problème de collision, alors un nouveau temps d'attente est tiré au sort entre 0 et 2δ, puis entre 0 et 4δ,etc. ..Jusqu'à ce que la trame soit émise. Ce principe est justifié par le fait que si une première collision se produit, il y a de fortes chances que les délais d'attente tirés au sort par chacune des deux stations soient très proches, donc il ne sera pas surprenant d'avoir une nouvelle collision.

Cette technologie s'appelle CSMA/CD (Carrier Sense Multiple Access With Collision Detect). Elle est efficace en générale mais elle a le défaut de ne pas garantir un délai de transmission maximal après lequel on est sûr que la trame a été émise, donc cela ne permet pas de l'envisager pour des application temps réels.

Les adresses physiques Ethernet sont codées sur 6 octets et sont censées être uniques car les constructeurs et les IEEE gèrent cet adressage de manière à ce que 2 coupleurs n'ont pas la même adresse.

Les adresses sont de trois types :

Unicast : dans le cas d'une adresse mono destinataire désignant un seul coupleur.

Broadcast : dans le cas d'une adresse de diffusion générale (tous les bits à 1) qui permet d'envoyer une trame à toutes les stations du réseau.

Multicast : dans le cas d'une adresse multi destinataire qui permet d'adresser une même trame à un ensemble de stations qui ont convenu de faire partie d'un groupe qui représente cette adresse Multipoint.

Le format des trames Ethernet est le suivant :

Adresse de destination	Adresse source	type	données	CRC
6	6	2	46-1500	4

Type : c'est le type de données transmises selon que c'est un datagramme IP, une requête ou réponse ARP ou RARP. Puis viennent les données transmises qui peuvent avoir une taille allant de 46 à 1500 octets. Dans le cas de données trop petites, comme pour les requêtes et réponse ARP et RARP on complète avec les bits de bourrage.

3.2. Protocoles ARP et RARP

Etant donné que le protocole IP et ses adresses peuvent être utilisées sur des architectures matérielles différentes (réseau Ethernet, Token Ring) possédant leur propre physique, il y a nécessité d'établir les correspondances biunivoques entre adresse IP et adresses matérielles des ordinateurs d'un réseau. Ceci l'objet des protocoles ARP (Adress Resolution Protocol) et RARP (Reverse Address Resolution Protocol). ARP fournit une correspondance dynamique entre une adresse IP connue et l'adresse matériel lui correspondant, RARP faisant l'inverse.

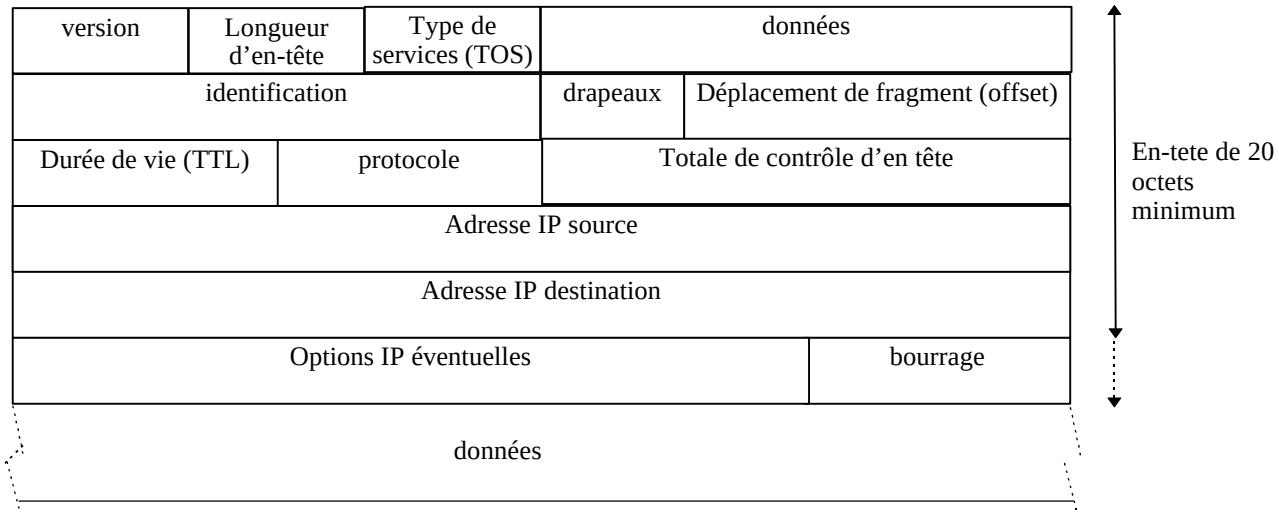
4. Couche réseau : le protocole IP

Le rôle du protocole IP est centré autour des trois fonctionnalités suivantes chacune étant décrite dans une des sous sections à venir.

- Définir le format du datagramme IP qui est l'unité de base des données circulant sur Internet.
- Définir le routage dans l'Internet
- Définir la gestion de la remise non fiable des datagrammes

4.1. Datagramme IP

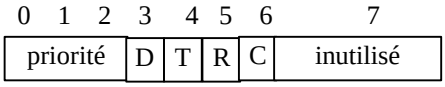
Le datagramme IP est constitué d'un en-tête suivi d'un champ de données. Sa structure précise est la suivante.



Version : Code sur 4 bits le numéro d version du protocole IP utilisé. Tout logiciel IP doit d'abord vérifier que le numéro de version du datagramme qu'il reçoit est un accord avec lui même, si ce n 'est pas le cas le datagramme est tout simplement rejeté.

Longueur d'en tête : représente sur 4 bits la longueur, en nombre de mots 32 bits de l'en tête du datagramme. Ce champ est nécessaire car une en-tête peut avoir une taille supérieure à 20 octets (taille de l'en tête classique) à cause des options que l'on peut y ajouter.

Type de service (TOE) : est codé sur 8 bits, indique la manière dont doit être géré le datagramme et se décompose en six sous champs comme suit :



Champ priorité : il varie de 0 et 7 (priorité maximale pour la supervision des réseaux) et permet d'indiquer l'importance de chaque datagramme. Même si ce champ n'est pas pris en compte par tous les routeurs, il permettrait d'envisager des méthodes de contrôle de congestion du réseau qui ne soient pas le problème qu'elles cherchent à résoudre.

Les 4 bits D, T, R et C permettent de spécifier ce que l'on veut privilégier pour la transmission de ce datagramme.

D=1, on essaye de minimiser le délai d'acheminement

T=1, pour maximiser le débit de transmission

R=1, pour assurer la plus grande fiabilité

C=1, pour minimiser les coûts de transmission.

Longueur totale : elle est codé sur 2 octets, utilisée avec la longueur d'en tête pour déterminer où commencent exactement les données transportées.

Champs identification : drapeau et déplacement de fragment interviennent dans le processus de fragmentation, des datagrammes IP.

Durée de vie (TTL) : indique le nombre maximale de routeurs que peut traverser le datagramme. Elle est initialisée à N (souvent 32 ou 64) par la station émettrice et décrémenté de 1 par chaque routeur qu'il reçoit un datagramme dont la durée de vie est nulle, il le détruit. Ainsi un message ne tournera pas indéfiniment sur Internet.

Protocole : il permet de coder quel protocole de plus haut niveau a servi à créer ce datagramme (6 pour TCP et 17 pour UDP).Ainsi la station destinatrice qui reçoit un datagramme IP pourra diriger les données qu'il contient vers le protocole adéquat.

Total de contrôle d'en tête : il est calculé à partir de l'en tête du datagramme pour en assurer l'intégrité.

Adresse IP source et destination : ils contiennent sur 32 bits les adresses de la machine émettrice et destination finale du datagramme.

Champ options : c'est une liste de longueur variable, mais toujours complétée par des bits de bourrage pour atteindre une taille multiple de 32 bits pour être en conformité avec la convention qui définit le champ longueur de l'en-tête.

4.2. Fragmentation des datagrammes IP

Comme un datagramme IP peut transiter à travers Internet sur ensemble de réseaux aux technologies différentes, il est impossible de définir à priori une taille maximale des datagrammes IP qui permettre de les encapsuler dans une trame quelque soit le réseau (1500 octets pour Ethernet et 4470 pour FDDI par exemple).

La taille maximale d'une trame réseau est appelée MTU et elle va servir à fragmenter les datagrammes trop grands pour le réseau qu'ils traversent. La taille du fragment est choisie la plus grand possible tout en étant un multiple de 8 octets. Un datagramme fragmenté n'est rassemblé que lorsqu'il arrive à la destination finale.

Le processus de fragmentation et réassemblage est rendu possible grâce aux différents champs suivants :

- Le champ déplacement fragment précise la localisation du début du fragment dans le datagramme initial
- Le champ drapeau comporte trois bits dont deux qui contrôlent la fragmentation. Si il est positionné à 1 le premier bit indique que l'on ne doit pas fragmenter un tel datagramme alors il le

rejette et envoi un message d'erreur à l'expédition. Un autre bit appelé fragments à suivre est mis systématiquement à 1 pour tous les fragments sauf le dernier fragment.

5. Protocoles TCP et UDP

On présente ici les deux principaux protocoles de la couche transport d'Internet que sont les protocoles TCP (transmission Control Protocol) et UDP (User Datagram Protocol). Tous les 2 utilisent IP comme couche réseau, mais TCP procure une couche fiable (alors même que IP ne l'est pas), tandis que UDP ne fait que transporter de manière non fiable des datagrammes.

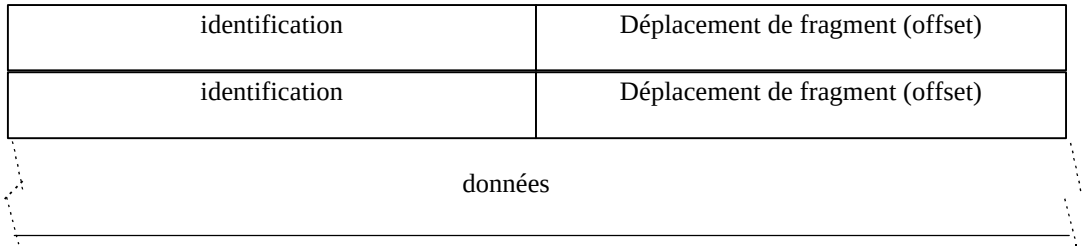
5.1. Protocoles UDP

Le protocole UDP utilise IP pour acheminer, d'une station à une autre, en mode non fiable des datagrammes qui lui sont transmis par une application. UDP n'utilise pas d'accusé de réception et ne peut donc garantir que les données ont bien étés reçues. Il ne réordonne pas les messages si ceux-ci n'arrivent pas dans l'ordre dans lequel ils ont été émis et il n'assure pas non plus le contrôle de flux, c 'est donc l'application qui utilise UDP de gérer tous ces problèmes.

Cependant, UDP fournit un service supplémentaire par rapport à IP, il permet de distinguer plusieurs applications destinatrices sur la même machine par l'intermédiaire des ports un port est une destination abstraite sur une station identifié par un numéro qui sert d'interface d'application pour recevoir et émettre des données.

Chaque datagramme émis par UDP est encapsulé dans un datagramme IP en y fixant à 17 la valeur du protocole.

Le datagramme IP est le suivant :



Numéro de port : chacun sur 16 bits identifiant le processus émetteur et récepteur.

Longueur : sa valeur minimale est de 8 bits

Checksum : C'est un total de contrôle qui est optionnel car il n'est pas indispensable lorsque UDP est utilisé sur un réseau très fiable.

5.2. Protocoles TCP

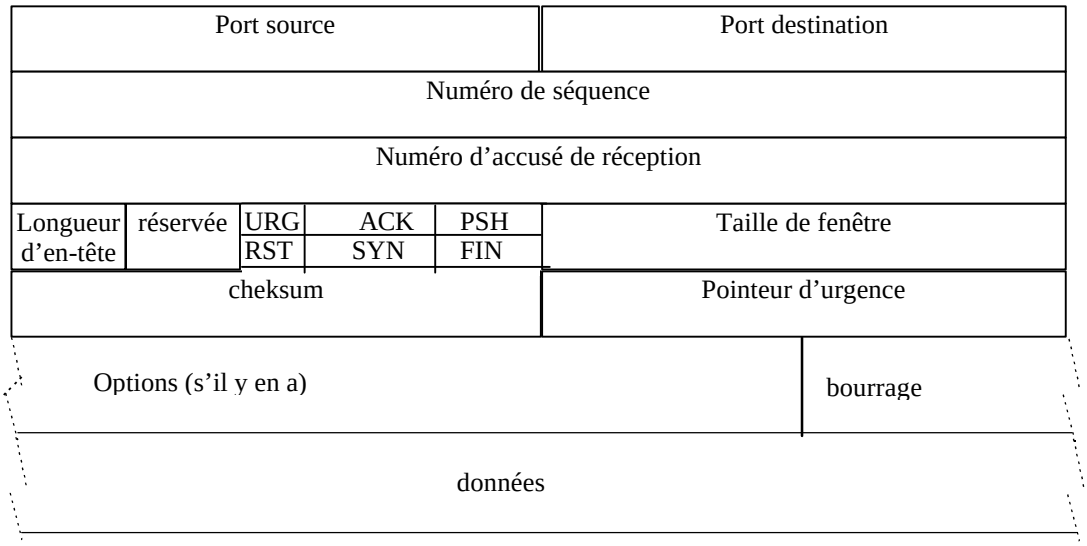
Contrairement à UDP, TCP est un protocole qui procure un service flux orienté connexion et fiable. Les données transmises par TCP sont encapsulées dans les datagrammes IP en y fixant la valeur du protocole à 6.

Le terme orienté connexion signifie que les applications qui dialoguent à travers TCP sont considérés l'un comme serveur et l'autre comme client, et qu'elles doivent établir une connexion avant de pouvoir dialoguer. Les stations vérifient donc préalablement que le transfert est autorisé, que les deux stations sont prêtes en s'échangeant des messages spécifiques. Une fois que tous les détails ont été précisés, les applications sont informées qu'une connexion a été établie et qu'elles peuvent commencer leurs échanges d'information. Cette connexion est Full Duplex.

A l'inverse de TCP peut regrouper des données d'une application pour ne former qu'un seul datagramme de taille convenable de manière à ne pas charger inutilement le réseau.

La fiabilité fournie par TCP consiste à remettre des datagrammes sans pertes, ni duplication, en utilisant la technique de l'accusé de réception.

Le format du segment TCP sert au trois fonctionnalités de TCP, c'est à dire, établir une connexion, transférer des données et libérer la connexion, est le suivant :



TCP à une taille totale de 20 octets et se compose des champs suivants :

Port source et destination : identifient les applications émettrices et réceptrice.

Numéro de séquence : donne la position du flux de segment dans le flux de données envoyées par l'émetteur.

Numéro d'accusé de réception : contient le numéro de séquence suivant le récepteur s'attend à le recevoir.

Longueur d'en tête : contient sur 4 bits la taille de l'en tête. Ainsi une taille peut varier de 20 octets (aucune option) à 60 octets (maximum d'options)

Champs réservé : comporte 6 bits réservés à un usage ultérieur.

La signification des 6 champs bits de code qui permettent de spécifier le rôle et le contenu du segment TCP, est la suivante :

URG=1, le pointeur de données urgente est valide.

ACK=1, le champ d'accusé de réception est valide.

PSH=1, ce segment requiert un PSH.

RST=1, on réinitialise la connexion.

SYN=1, on synchronise les numéros de séquence pour initialiser une connexion.

FIN=1, l'émetteur a atteint la fin de flot de données.

Taille de fenêtre : est un champ de 16 bits qui sert au contrôle de flux selon la méthode de la fenêtre glissante. Il indique le nombre d'octets que le récepteur est prêt à accepter. Ainsi l'émetteur augmente et diminue son flux de données en fonction de la valeur de la fenêtre qu'il reçoit.

Checksum : est un total de contrôle sur 16 bits utilisé pour vérifier la validité de l'en tête et les données transmises.

Pointeur d'urgence : est un offset positif qui, ajouté au numéro de séquence du segment, indique le numéro du dernier octet de donnée urgente. Il faut également que le bit URG soit positionné à 1 pour indiquer les données urgentes que le récepteur TCP doit passer plus rapidement possible à l'application associée à la connexion.

6. Applications

Toutes les applications sont bâties sur le modèle « client serveur » à savoir qu'une des extrémités de la connexion (TCP UDP)/IP rend les services de l'autre extrémité.

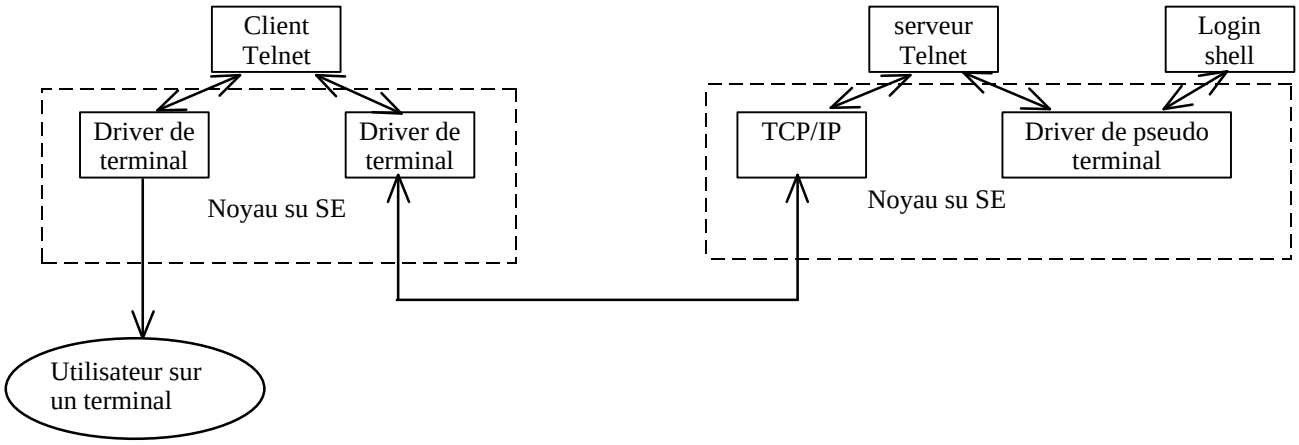
6.1. Protocole de démarrage : BOOTP

BOOTP (BOOTsrop Protocol) est un protocole de démarrage des terminaux X ou stations sans disques qui utilisent UDP comme couche de transport et est généralement associé à TFTP ou NFS. Il sert principalement à fournir son adresse IP à une machine que l'on démarre sur le réseau.

6.2. Connexion à distance : Telnet et Rlogin

Telnet et Rlogin sont deux applications qui permettent à un utilisateur de se connecter à distance sur un ordinateur, pourvu que cet utilisateur y dispose d'un accès autorisé. Ces deux applications permettent toutes les deux de prendre le contrôle (du moins partiellement) d'un ordinateur distant, mais Rlogin ne permet de le faire qu'entre deux machines UNIX, tandis qu'il existe des clients Telnet, pour de nombreuses plates-formes (UNIX, Windows, Mac OS,...). Telnet et Rlogin sont tous les deux bâtis sur TCP.

Le schéma fonctionnel de Telnet et le suivant :



Telnet définit une interface de communication, le terminal virtuel de réseau, pour que clients et serveurs n'aient pas à connaître les détails d'implantation de chaque système d'exploitation.

De cette façon, les échanges se font dans un langage commun compris à la fois par le client et le serveur qui n'ont qu'à assurer de (ou vers) leur propre langage vers (depuis) ce langage cible.

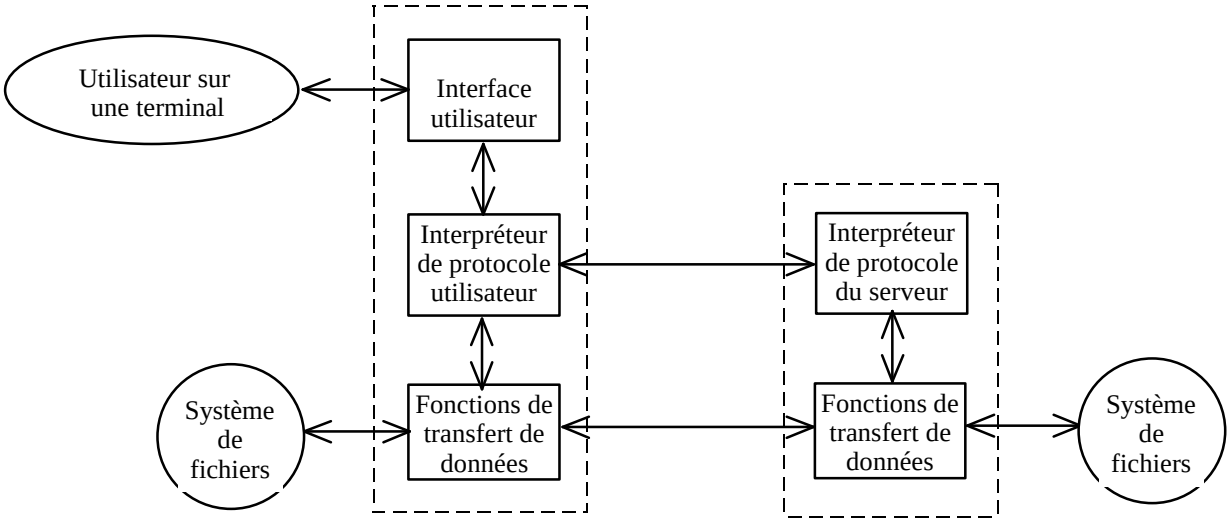
6.3. Système de fichier en réseau : NFS

NFS (Network File System) est un system qui permet de rendre transparente l'utilisation de fichiers réparties sur différentes machines. Lorsqu'un processus utilisateur à besoin de lire, écrire ou accéder à un fichier le système d'exploitation transmet la demande soit au système de fichier local, soit au client NFS. Dans ce dernier cas le client NFS envoie des requêtes au serveur NFS de la machine distante. Ce serveur s'adresse à la routine locale d'accès au fichiers qui lui retourne le résultat retransmis vers le client par la connexion UDP (ou TCP)/IP. Il ne s'agit pas ici de transférer un fichier d'une machine à l'autre mais simplement le rendre disponible de manière totalement transparente.

6.4. Transfert de fichier : TFTP et FTP

TFTP (Trivial File Transport Protocol) et FTP (File Transport Protocol) permettent tous les deux de transférer des fichiers d'une machine à une autre. Cependant TFTP, bâti sur UDP, est plus sommaire que FTP qui utilise TCP.

L'utilisation de FTP depuis un poste client pour aller chercher ou déposer un fichier sur un serveur nécessite de la part de l'utilisateur de se connecter avec nom et mot de passe. Donc si l'utilisateur n 'est pas reconnu la connexion FTP ne sera pas établie. FTP est défini au dessus de TCP et utilisent deux connexions TCP/IP pour fonctionner comme illustré.



Tout d'abord on y voit que le client utilise FTP à travers une interface qui peut être graphique ou texte. La connexion de contrôle est établie de façon normale en mode client serveur sur le port 21 du serveur et sur un port aléatoire du client pour tout ce qui est de type de transfert interactif. Elle sert donc tout le temps de la session de transférer les commandes du client et presque toutes les réponses du serveur.

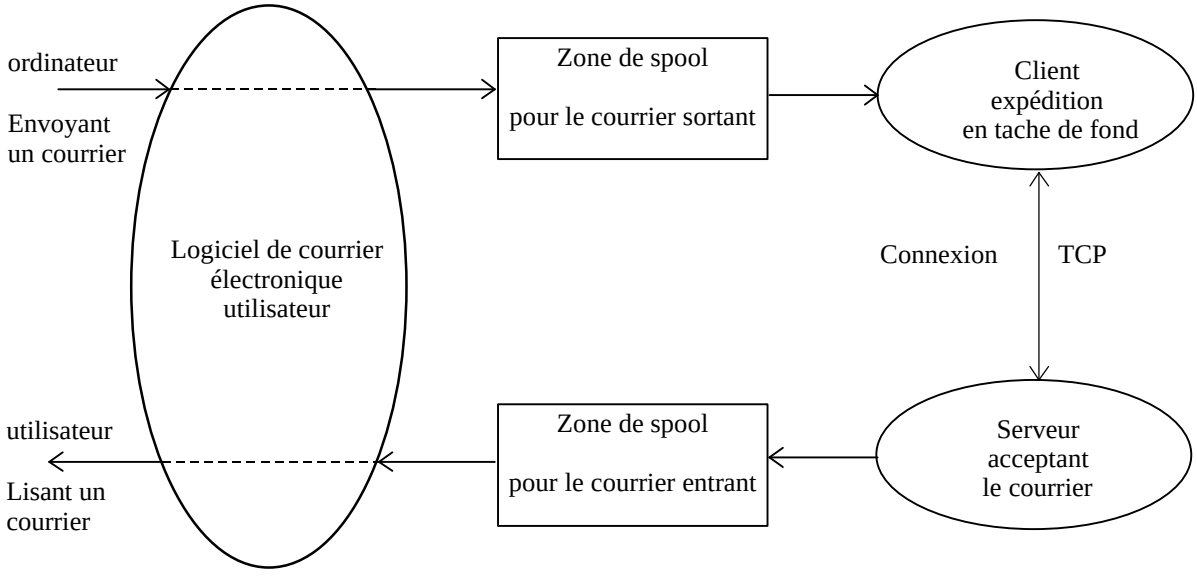
La connexion de données sert à transférer les fichiers et les contenus du répertoire du serveur, c'est à dire les transferts de masse. En effet, lorsqu'un client demande le contenu d'un répertoire la réponse peut être longue et il est préférable de l'envoyer sur cette connexion plutôt que sur celle de transfert interactif.

A chaque fois qu'un fichier doit être transféré, dans un sens ou l'autre le client initie une connexion de données en s'attribuant un port et envoie au serveur une demande de connexion sur la connexion de contrôle. Le serveur se sert du numéro de port reçu pour établir la connexion de données entre son port 20 et le port indiqué par le client.

6.5. Courrier électronique : SMTP

Le courrier électronique au sein d'Internet est géré par le protocole SMTP (Simple Mail Transfert Protocol) bâti sur TCP (port 25). Il permet d'échanger des messages en un expéditeur et un (ou plusieurs) destinataires pourvus que leurs adresses soient connues.

Une des caractéristiques principales du protocole SMTP est d'effectuer une remise différée du courrier qui assure le service sera correctement rendu même si le réseau ou l'ordinateur destinataire sont momentanément en panne ou surchargés. Pour cela le système de messagerie fonctionne de la façon suivante :



Un courrier expédié par un utilisateur est d'abord copié dans une mémoire de spool accompagné des noms de l'expéditeur et le récepteur, de l'ordinateur destinataire et l'heure de dépôt.

Puis le système de messagerie active une tâche de fond le processus de transfert de courrier qui devient client. Il associe le nom de l'ordinateur destinataire à une adresse IP et tente d'établir une connexion TCP avec le serveur SMTP de celui ci. Si cela réussit, le processus de transfert envoie une copie du message au destinataire qu'il enregistre dans une zone spool spécifique. Lorsque le client et le serveur se sont confirmés l'envoi et l'enregistrement du message le client supprime la copie locale. Si le client n'arrive pas établir une connexion TCP ou si celle si est rompue lors du transfert d'un message, il enregistre l'heure de cette tentative et réessaye quelque temps plus tard d'expédier le message.

D'une manière générale un système de messagerie examine régulièrement sa zone spool en envoi et tente d'expédier les messages (nouveau ou en attente à cause d'échec) qui s'y trouvent. Il finira par retourner à son expéditeur un message impossible à expédier après un délai important. Ce mode de fonctionnement (établir une connexion de bout en bout) assure qu'aucun message ne peut se perdre soit il est délivré, soit son expéditeur est prévenu de l'échec.

6.6. World Wide web : HTTP

HTTP (Hyper Text Tansfert Protocol) est le protocole de communication du web permettant d'échanger des documents hypertextes contenant des données sous la forme de texte, d'images fixes ou animés ou de son.

Tout client web communique avec le port 80 du serveur HTTP par l'intermédiaire d'une, ou plusieurs, connexions TCP simultanées chacune des connexions TCP servent à récupérer l'un des composants de la page web :

Geturl : renvoie l'information spécifiée à url.

Headurl : renvoie l'en-tête de l'information demandé et non pas le contenu du document.

Post : pour envoyer du courrier électronique, de messages, des news ou des formulaires interactifs remplis par l'utilisateur.