Systèmes Communicants

(ISC21-2, 32h cours+10h TD+20hTP)
Informatique et **S**ystèmes d'**I**nformation pour la **S**anté (ISIS)
Ali LARAB

Table des matières

Présentation du l'UE « Systèmes Communicants » (ISC21-2)

Objectif: offre un panorama général des différents éléments constituant le nouveau paysage de la communication autour de l'Internet multimédia.

Volume horaire: 62h (32h cours + 10h TD + 20h TP).

Contenu:

Avec Ali Larab :

I – Introduction à Internet: Architecture et organismes de gestion, adressage IP et routage, applications classiques.

II – Systèmes de transmission et Accès au réseau: Systèmes filaires et sans-fil, commutation (circuit et paquet) et Routage, accès filaires (xDSL, câble, CPL), Systèmes sans fil (802.11, GSM, UMTS, satellites).

III – Applications et QOS: Protocoles de transport UDP et TCP, stratégies de QoS,

Avec André Aoun :

Protocoles RTP/RTCP, contrôle du streaming (RTSP), approche INTSERV (RSVP), approche DIFFSERV (Classes de service), signalisation avec H.323, signalisation avec SIP, VoIP et la ToIP, vers le travail collaboratif.

Avec Sylvie Trouilhet :

IV – Interaction Homme-Machine (IHM)

Evaluation:

Contrôle de connaissances en salle,

+

Réalisation d'un projet (étude et/ou développement).

Bibliographie:

- Guy Pujolle, Les réseaux, Edition Eyrolles, ISBN: 2-212-09119-2.
- Douglas Comer, TCP/IP: Architectures, protocoles, applications, Edition Dunod, ISBN: 2-10-008181-0.
- Olaf Kirch & Terry Dawson, Administration réseau sous Linux, Edition Oreilly, ISBN: 2-84177-125-3.
- Chauvin Hameau, Wi-fi - maîtriser le réseau sans fil, Edition:ENI, ISBN : 2746020548
- Jean-François Susbielle, Internet, multimédia et temps réel, Edition: Eyrolles, ISBN : 2212091184
- Guy Cazuguel & Bassel Solaiman & Collectif, Santé et technologies de l'information : Annales des télécommunications Tome 58 N° 5-6 Mai-juin 2003, Edition: Hermes Science Publications, ISBN : 2746207753
- Jean-François Bouchaudy, TCP/IP sous Linux : Administrer réseaux et serveurs Internet/Intranet sous Linux, Edition: Eyrolles, ISBN : 2212113692
- ...

I. Introduction aux réseaux et à Internet

I.1 Architecture et organismes de gestion

I.1.1 Définition d'un réseau?

Le réseau est le résultat de la connexion de plusieurs machines. Il a pour but l'échange d'informations entre utilisateurs ou machines, ainsi que le partage de temps machine, de services ou de matériel.

Selon son utilisation, le terme « réseau » peut avoir plusieurs sens:

- peut désigner l'ensemble des machines ou d'infrastructure informatique d'une organisation. Ex. Internet, réseau LAN...
- peut désigner ou décrire la façon dont les machines d'un site sont interconnectées. Ex. réseau ethernet, réseau en étoile...
- peut désigner le protocole utilisé pour communiquer entre les machines formant le réseau en question. Ex. TCP/IP, NetBeu de Microsoft...

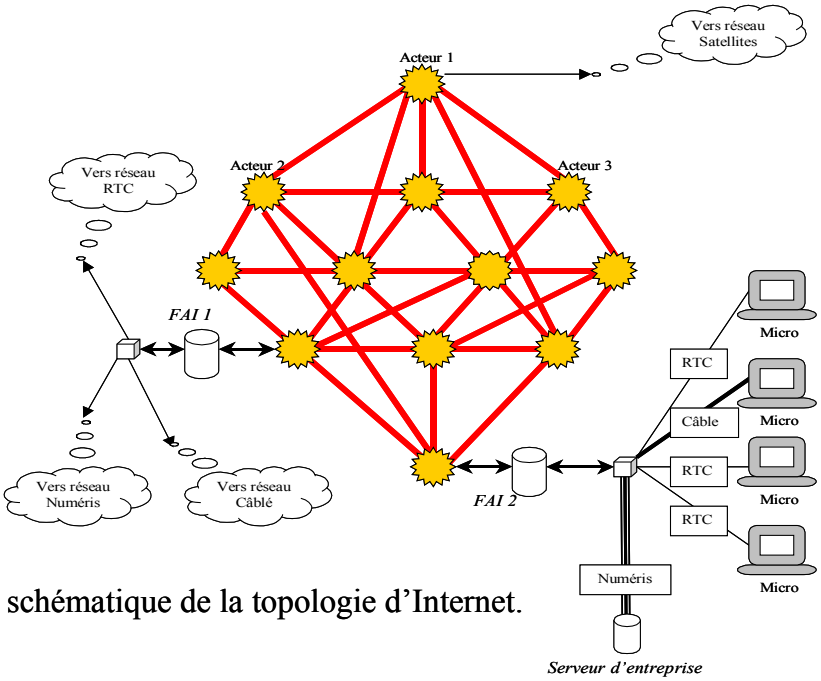
I.1.2 Utilité d'un réseau

Pour des raisons de coûts et de performances, le nombre de machines utilisées dans les entreprises, laboratoires ... a augmenté. Les informations disponibles sur ces différentes machines doivent être cohérentes. Certains services et données ne sont disponibles que sur certaines machines. On est alors amené à échanger des informations ou à exécuter une tâche pour une autre machine puis lui transmettre les résultats à la fin d'exécution. Cet échange ne doit pas se faire manuellement, car il doit être simple, rapide et automatisé. D'où la nécessité de relier les machines entre elles et donc la création d'un « réseau ». C'est ainsi que les réseaux ont commencé à voir le jour.

A l'aide des réseaux, aujourd'hui on arrive à :

- contrôler et améliorer le fonctionnement et la fiabilité de toute un système ou un site (usine, laboratoire...),
- augmenter les ressources matérielles et logicielles d'un site,
- accéder à ses ressources à distance,
- communiquer et échanger des informations (données, fichiers, photo numérisées, musique...) entre utilisateurs et/ou applications sur l'étendu du réseau (dans le monde entier dans le cas d'Internet),
- faire des enseignements ou de la vidéoconférence à distance,
- chercher et mettre à disposition des autres pleins d'informations (Web),
- communiquer facilement et rapidement quelque soit la distance (...)
- rendre divers services: acheter et réserver à distance (téléachat), écouter la radio et voir la télévision sur le réseau, jouer sur le réseau, échanger des messages électroniques et des messages instantané (chat)...

I.1.2.1 Cas particulier: Internet

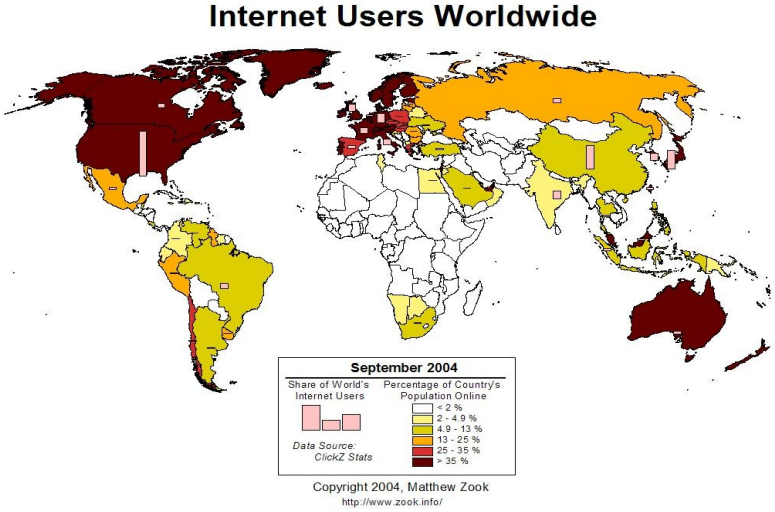


Représentation schématique de la topologie d'Internet.

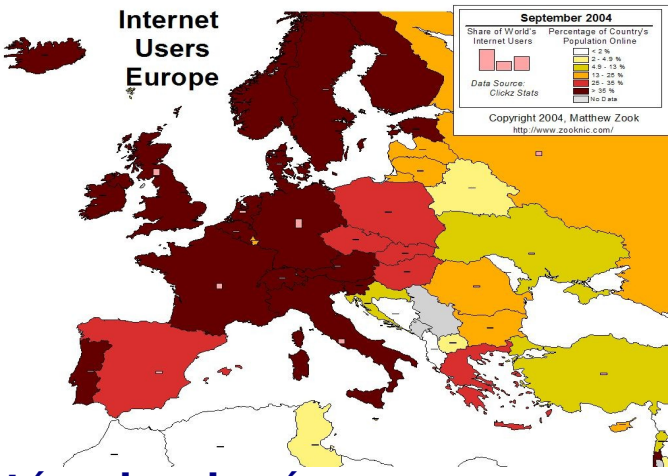
Internet est un réseau de réseaux. C'est un ensemble de réseaux, parmi lesquels Arpanet, NSFnet, des réseaux régionaux comme NYsernet, des réseaux locaux d'universités et centres de recherche ainsi qu'un certain nombre de réseaux militaires.

Internet a aujourd'hui 37 ans (commencé en 1969). Fin 1969 il comptait 4 machines. Aujourd'hui (2006) il compte des millions de machines (dont plus de 6,5 millions de serveurs qui hébergent plus d'un milliard de pages Web.) et 938,710 millions d'utilisateurs (soit 14,6 % de taux de pénétration dans le monde. Source: Nielsen//NetRatings, ITU, InternetWorldStats, juillet 2005).

L'Internet dans le monde (2004): (Source: http://www.zook.info/)



L'internet en Europe (2004): (Source: http://www.zook.info/)



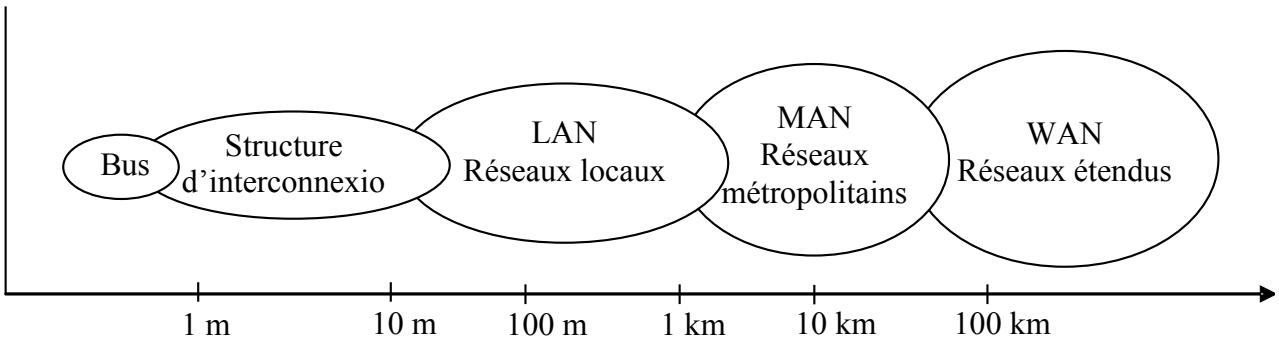
I.1.3 Catégories de réseau

La classification des réseaux en plusieurs catégories se base principalement sur le critère de distance maximale entre les entités communicantes les plus éloignées dans le réseau. De ce critère découlent quatre (voire cinq) catégories.

1. **Structures d'interconnexion**: Cette catégorie de réseaux relie, dans une même pièce ou à des distances faibles, différents calculateurs (pré- et post- processeurs d'ordinateurs vectoriels) entre eux. Le rayon d'un LAN est entre 1 m et 10 m.
2. **Réseaux locaux**: **(LAN, Local Area Network)**: Il correspondent par leur taille aux réseaux intra-entreprise. Ils permettent le transfert de toutes les informations numériques de l'entreprise. Elle est constitué d'un ensemble d'ordinateurs (ou micro-ordinateur) gérés par quelques serveurs. C'est la catégorie de réseau la plus utilisée, car facile à câblée, revient moins cher, la plupart des communications de bureau sont locales à un bâtiment et parce qu'on trouve plus d'applications pour cette catégorie de réseau que pour une autre. Le rayon d'un LAN est entre 10 m et 1 km.

3. **Réseaux métropolitains (MAN, Metropolitan Area Network)**: Cette catégorie correspond à une interconnexion de plusieurs bâtiments situés dans une même ville. Elle peut interconnecter par exemple les différents réseaux locaux de ces bâtiments. Le rayon d'un LAN est entre 1 km et 100 km.

4. **Réseaux étendus (WAN, Wide Area Network)**: Cette catégorie est destinée à l'échange d'informations entre entités sur des distances à l'échelle d'un pays. Le réseau utilisé peut être soit terrestre et utilise des infrastructures au niveau du sol, soit satellite, et demande des engins spatiaux pour mettre en place les répondeurs qui transmettront les signaux vers la terre. Le rayon d'un LAN est entre 100 km et 10 000 km.



Les différentes catégories de réseaux informatiques

Remarques:

a) On peut aussi intégrer dans cette classification la catégorie « **Bus** ». Il s'agit de l'ensemble du matériel constitué des bus qui doivent relier les processeurs, des mémoires, des entrées-sorties d'un calculateur ou d'un multiprocesseur... Dans cette catégorie la distance de connexion maximale entre les deux entités les plus éloignés est très faible. Elle est généralement inférieure à un mètre. Normalement cette catégorie ne doit pas apparaître dans cette classification, car on ne commence à parler de réseau que lorsqu'on relie des systèmes autonomes (ex. calculateurs).

b) Quelques uns des protocoles développés pour les LAN tiennent compte d'une distance maximale entre entités pour fonctionner. Ces protocoles ne sont donc pas adaptés aux MAN ou aux WAN. L'inverse peut aussi être vrai. Certains protocoles dédiés aux WAN sont très complexes et donc non adaptés aux LAN.

c) Pour classifier les réseaux, d'autres critères existent. On peut classifier les réseaux selon:
   - **débit**: réseau bas débit, moyen débit, haut débit, très haut débit,
   - **modèle d'architecture**: réseau OSI, X.25, SNA, DNA, DSA...
   - **gestion**: réseau public ou privé
   - ...

d) On peut faire aussi des classifications au niveau de chaque catégorie. Au niveau du LAN par exemple, on peut par exemple faire les classifications suivantes:
   - réseau **PABX** (Private Automatic Branch eXchange) ou réseau **PBX** (réseau téléphonique d'une entreprise);
   - réseau **bureautique** (partage d'imprimantes, de logiciels...), réseau **industriel** (composé de capteurs et d'actionneurs);
   - ...

I.1.4 Architectures de communication et Normalisation

I.1.4.1 Architectures de communication

Une architecture de communication (voire architecture réseau) est une architecture (structure d'éléments définissant un système complexe) qui définit l'ensemble des entités nécessaires à la communication et les règles régissant les échanges entre ces éléments.

I.1.4.2 Normalisation

Si toutes les machines communicantes utilisent le même langage on aurait pas besoin d'une normalisation. Malheureusement chacun des constructeurs de matériel informatique a défini sa propre architecture de communication permettant l'échange de données entre leurs équipements informatiques. IBM par exemple a défini SNA (*Systems Network Architecture*), DEC a a défini DNA (*Digital Network Architecture*)... Ces architectures de communication sont des architectures propriétaires (architectures constructeurs), car étaient souvent liées au matériel et équipements du constructeur. Ainsi, pour communiquer il faut disposer à chaque fois d'une machine du même constructeur que celui du réseau avec lequel on veut communiquer ou réinventer un moyen (matériel ou logiciel) permettant l'échange d'informations entre des appareils appartenant à des constructeurs différents.

Pour faire face à cette situation, l'organisation ISO (International Standard Organization) s'est occupé de la normalisation des communications dans le domaine des télécommunications et de l'interconnexion de systèmes, et a développé en 1979 le modèle de référence OSI (Open Systems Interconnection). Elle publia en 1984 le document ISO 7498 relatif à ce modèle. Ce dernier est référencé au CCITT (*Comité Consultatif International Téléphonique et Télégraphique*, actuellement UIT (*Union International des Télécommunications*)) sous la norme X.200.

I.1.5 Le modèle OSI

Le modèle OSI de l'ISO est un modèle à 7 couches qui décrit le fonctionnement d'un réseau à commutation de paquets[1]. Chacune de ces couches correspond et résout une catégorie de problèmes rencontrés dans la transmission des informations via un réseau. Il y a exactement 7 couches, car:

- Les fonctions de chaque couche doivent être choisies en pensant à la définition de protocoles normalisés internationaux,
- Il ne faut créer une couche que lorsque le niveau d'abstraction est nécessaire (le nombre de couches doit être réduit pour que l'architecture soit maîtrisable),
- et au même temps ce nombre de couche doit être assez grand pour que des fonctions très différentes ne cohabitent pas dans une même couche.

L'avantage de découper le problème de la trasmission d'information en couches:

- Chaque couche exerce une fonction bien définie. Quand on met en place un réseau, il suffit de trouver une solution pour chacune des couches. Chacune de ces couches de protocoles doit offrir un service à la couche qui lui est supérieure et utiliser les services de sa couche inférieure.
- Interopérabilité: La couche n+1 peut utiliser les services de la couche n qui est très différente à condition que l'interface n/n+1 reste la même.

1 Deux stratégies peuvent être utilisées pour transmettre des informations: la première consiste à envoyer les données en un seul bloc, alors que la deuxième fragmente ces données en plusieurs paquets qui sont envoyés séparément puis reconstitués (rassemblés) sur la machine du destinataire. La 1ère stratégie n'est pas utilisée à cause des erreurs de transmissions qui peuvent survenir sur un réseau.

- Facilité de développement et de modification. On peut modifier la couche n (un protocole) de façon indépendante tant que l'interface avec les deux couches adjacentes (n-1 et n+1) reste inchangée. On peut donc apporter des modifications techniques pour une couche sans être obligé de tout changer.
- Chaque couche n garantit à la couche n+1 que le travail qui lui a été confié est réalisé sans erreur.

La figure ci-dessous représente les différentes couches du modèle OSI :

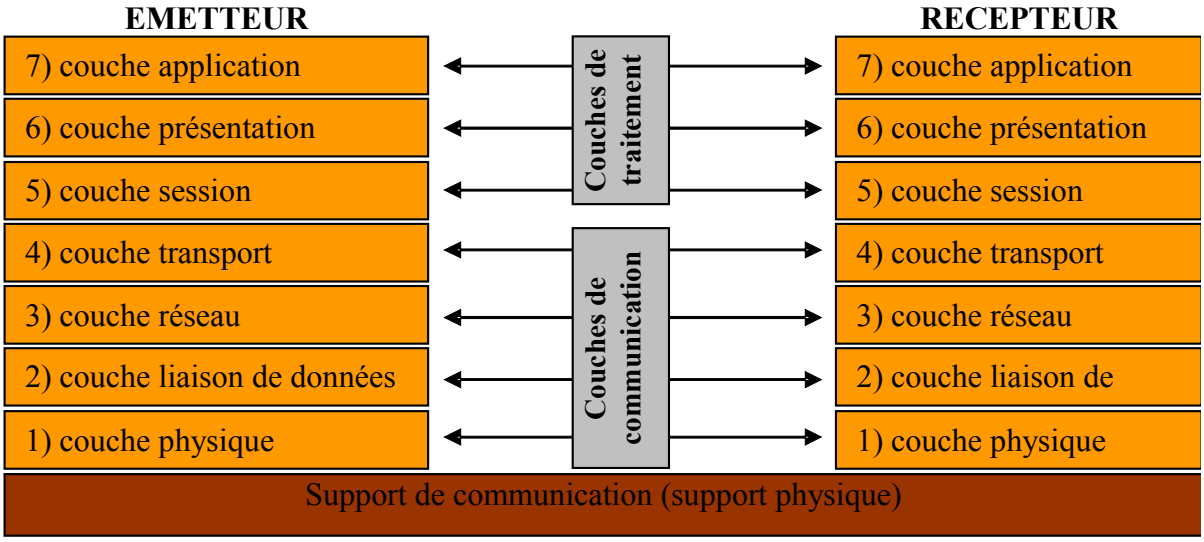


Figure III.7 : L'architecture O.S.I.

Remarque: Le modèle OSI lui-même ne définit pas de service particulier et encore moins de protocole.

Dans ce qui suit nous donnons une breve description de chacune de ces couches. Une étude détaillée sera faite dans le chapitre correspondant.

I.1.5.1 Couche 1: Physique (Matériel)

La couche physique est la première couche du modèle O.S.I. Son service est limité à l'émission et la réception des bits, en les transmettant de façon brute (sans aucune signification) sur un canal de communication. C'est la seule couche effectivement connectée au réseau, car c'est la seule à être concernée par l'interprétation des tensions du câble (les 0 et les 1). Le rôle de cette couche est de garantir la parfaite transmission des données en conduisant les éléments binaires jusqu'à leur destination sur le support physique. Cette couche contient tout le matériel et logiciel nécessaires au transport correct des éléments binaires, en particulier, les interfaces de connexion des équipements informatiques, les modems, les multiplexeurs, les nœuds de commutation formant le matériel intermédiaire entre l'émetteur et le récepteur et enfin divers équipements spécifiques au réseau, mais nécessaires pour assurer la continuité du chemin physique, comme le satellite dans le cas d'une communication par voie hertzienne.

Cette couche s'occupe donc des problèmes strictements matériels.Elle doit ainsi spécifier:

- dans le cas de communications par câble: le type du câble (coaxial, torsadée...), le type du signal électrique envoyé (tension, intensité...), la nature des signaux (carrés, sinusoïdaux...), les limitations (longueur, nombre de stations...), si un blindage est nécessaire ou non...

- dans le cas de communications hertziennes: les fréquences, le type de modulation (phase, amplitude...)...
- dans le cas de communications par fibre optique: le nombre de brins, la couleur du laser, la section du câble...
- ....

Le PDU (*Protocole Data Unit*) ou unité d'information de la couche physique est le « **bit** » (0 ou 1), représenté par une certaine différence de potentiel.

Quelques protocoles (codages) et normes de la couche physique : CSMA/CD, CSMA/CA, Codage NRZ, Codage Miller, RS-232, RS-449, 10Base2, 10BASE5, Paire torsadée, 10BASE-T, 100BASE-TX, ISDN, T-carrier, ADSL, SDSL, VDSL, USB, IEEE 1394, Wireless USB, Bluetooth, ...

I.1.5.2 Couche 2: Liaison de données

La couche liaison de données a pour but la gestion des communications entre deux machines adjacentes (2 machines reliées directement entre elles par un support physique). Nous avons dit dans le point précédent que les données n'ont aucune signification pour la couche physique. C'est donc à la couche liaison de données de leur donner une signification en regroupant (ou fractionnant) la succession de bits (train de bit, données brutes) -reçues de la couches physique- en un ensemble de trames qu'elle doit transmettre après en séquence. Elle gère également les trames d'acquittement renvoyées par le récepteur. Un autre rôle important de cette couche est la détection des erreurs intervenues sur la couche physique (problème sur la ligne de transmission) et parfois la correction de ces erreurs (par exemple dans les réseaux X.25 et GSM), par l'utilisation d'algorithmes de détection et de correction d'erreurs de bas niveau, pour déterminer quand il faut réémettre des informations. Cette couche intègre aussi une fonction de contrôle de flux pour éviter l'engorgement du récepteur.

Cette couche est souvent découpée en 2 sous-couches:

- MAC (Medium Access Control): Elle sert à la synchronisation des accès au support physique. Cette sous-couche est souvent réalisée par du matériel spécialisé comme une carte Ethernet (à l'exception des carte à puce par exemple).
- LLC (Logical Link Control): elle se situe au-dessus de la sous-couche MAC. Elle sert principalement à la gestion des erreurs. Contrairement à la sous-couche MAC, LLC est une réalisation logicielle.

Le PDU de la couche liaison de données est appelé « **trame** ». Une trame est composées de quelques centaines à quelques milliers d'octets maximum.

Quelques protocoles de la couche liaison de données: Ethernet, Anneau à jeton, ARCnet, Econet, CAN (Controller Area Network), FDDI (Fiber Distributed Data Interface), LocalTalk,X.21, X.25, Frame Relay, BitNet,Wi-Fi, PPP(Point-to-point protocol), HDLC, MPLS (Multiprotocol Label Switching), SLIP (Serial Line Internet Protocol), Token Ring...

I.1.5.3 Couche 3: Réseau

La couche réseau a pour but de construire une voie de communication de bout en bout à partir de voies de communication avec ses voisins directs. En effet, pour aller d'un émetteur à un récepteur, il faut passer par des nœuds de commutations intermédiaires ou par des passerelles qui interconnectent deux ou plusieurs réseaux (ou sous-réseaux) entre eux. La couche réseau a donc pour rôle l'acheminement correct des paquets d'informations de l'émetteur jusqu'au récepteur, à travers cette succession de connexions physiques, en utilisant les services offerts par la couche liaison de chacune de ces connexions. Cette couche est donc la seule à être directement concernée par la topologie du réseau. Elle est aussi la dernière à être supportée par toutes les machines du réseau pour le transport des données; Le déroulement des couches supérieures se fait uniquement sur les deux machines communicantes (émetteur et récepteur).

Les principales fonctions de cette couche sont :

- Le « **routage** »: sert à déterminer le chemin permettant de relier les deux machines distantes, à travers un maillage de nœuds de commutation. Un algorithme de routage est utilisé pour optimiser le temps de réponse. D'autres informations sont nécessaires à cela telles que le mécanisme de routage et de calcul des tables de routage (tables statiques ou dynamiques...).
- Le « **contrôle de flux** »: sert à éviter les embouteillages des paquets dans le réseau (congestion des nœuds, engorgement du sous-réseau).
- L' « **adressage** »: c'est au niveau de cette couche qu'il faut ajouter des adresses complètes dans les différents paquets, pour qu'ils atteignent leur destinataire.

Le PDU de la couche réseau est appelé « **paquet** ».

Remarques: La couche réseau est la plus qui caractérise l'architecture réseau utilisée. C'est la raison pour laquelle l'architecture en question prend souvent le nom du protocole principal de cette couche (on parle par exemple d'un réseau IP, d'un réseau NetBeu ou d'un réseau ATM).

Quelques protocoles de la couche réseau : NetBEUI, IPv4, IPv6, ARP, IPX, BGP, ICMP, OSPF, RIP, IGMP, IS-IS, CLNP, WDS, ATM, ...

I.1.5.4 Couche 4: Transport

La couche transport est l'une des couches les plus importantes. Elle est responsable du bon acheminement des messages complets au destinataire. Elle gère les communications de bout en bout entre les processus (émetteur et récepteur). Elle gère l'ensemble du processus de connexion, avec toutes les contraintes qui y sont liées et d'une manière transparente pour la couche session.

Le principal rôle de cette couche est de découper les messages de la couche session (quand ils sont trop grands) en unités plus petites, puis les passer à la couche inférieure (couche réseau), tout en s'assurant que les messages arrivent correctement au récepteur. Au niveau de ce dernier (récepteur), son rôle consiste à rassembler les paquets reçus de la couche inférieure (réseau) pour former le message à transmettre à la couche supérieure (session).

Un autre rôle attribué à cette couche est l'optimisation des ressources du réseau; Elle crée une connexion réseau pour chaque connexion de transport requise par la couche session. Afin, par exemple, d'améliorer le débit (QoS), cette couche est aussi capable de créer plusieurs connexions réseaux par processus de la couche session pour répartir les données. Elle peut aussi utiliser une seule connexion réseau pour transporter plusieurs messages à la fois (grâce au multiplexage). Elle est donc responsable du type de service à fournir à la couche session et donc aux utilisateurs du réseau (service en mode connecté ou non, avec ou sans garantie d'ordre de délivrance, diffusion du message à plusieurs destinataires à la fois).

Cette couche s'occupe de l'établissement et du relâchement des connexions sur le réseau. C'est elle aussi qui est la dernière couche où on se préocupe de la correction des erreurs (exception faite pour le service DNS sur UDP dans la pile TCP/IP).

Le PDU de la couche transport est appelé « **message** » ou « **segment** ».

Quelques protocoles de la couche transport : TCP, UDP, ICMP, SCTP, RTP, SPX, TCAP, DCCP, ...

I.1.5.5 Couche 5: Session

Le rôle de cette couche est la gestion (organisation et synchronisation) des échanges entre tâches distantes. Elle établit une liaison entre les deux programmes d'application et commande leur dialogue (déterminer qui doit émettre à l'instant 't' (gestion du jeton)). Comme son nom l'indique, cette couche a pour but d'ouvrir et de fermer des sessions entre les utilisateurs distants. Avant de communiquer, elle s'assure que l'utilisateur que l'on veut atteindre -ou du moins son représentant, qui peut être une boite aux lettres électronique- est bien présent. En effet, il est inutile d'émettre de l'information s'il

n'y a personne à l'autre extrémité pour récupérer ce qui a été envoyé. C'est elle aussi qui détermine si toutes les données pertinentes ont été reçues pour la session afin d' interrompre la réception et la transmission de données.

La couche session permet aussi d'insérer des points de reprise dans le flot de données pour pouvoir reprendre le dialogue après une panne. Elle réalise aussi le lien entre les adresses logiques et les adresses physiques des tâches réparties. Elle a aussi pour rôle de faire de telle sorte à pouvoir transmettre des informations en multipoints (étoile ou diffusion), car les services transport sont des service de communication point à point.

Quelques protocoles de la couche session : RPC, Netbios, ASP

I.1.5.6 Couche 6: Présentation

Les couches au-dessous de la couche présentation transortent des octets bruts sans se préocuper de leur signification (les textes, nombres... n'ont aucune signification pour elles). Cette tâche est confiée à la couche présentation; Cette couche a donc pour rôle de coder les données applicatives et de rendre l'information compatible entre les tâches communicantes. Elle convertit les données applicatives manipulées par les programmes en un ensemble d'octets transportés par le réseau.

Cette couche peut aussi convertir les données, les reformater, les crypter et les compresser.

Quelques protocoles de la couche présentation : XDR, ASN.1, SMB, AFP

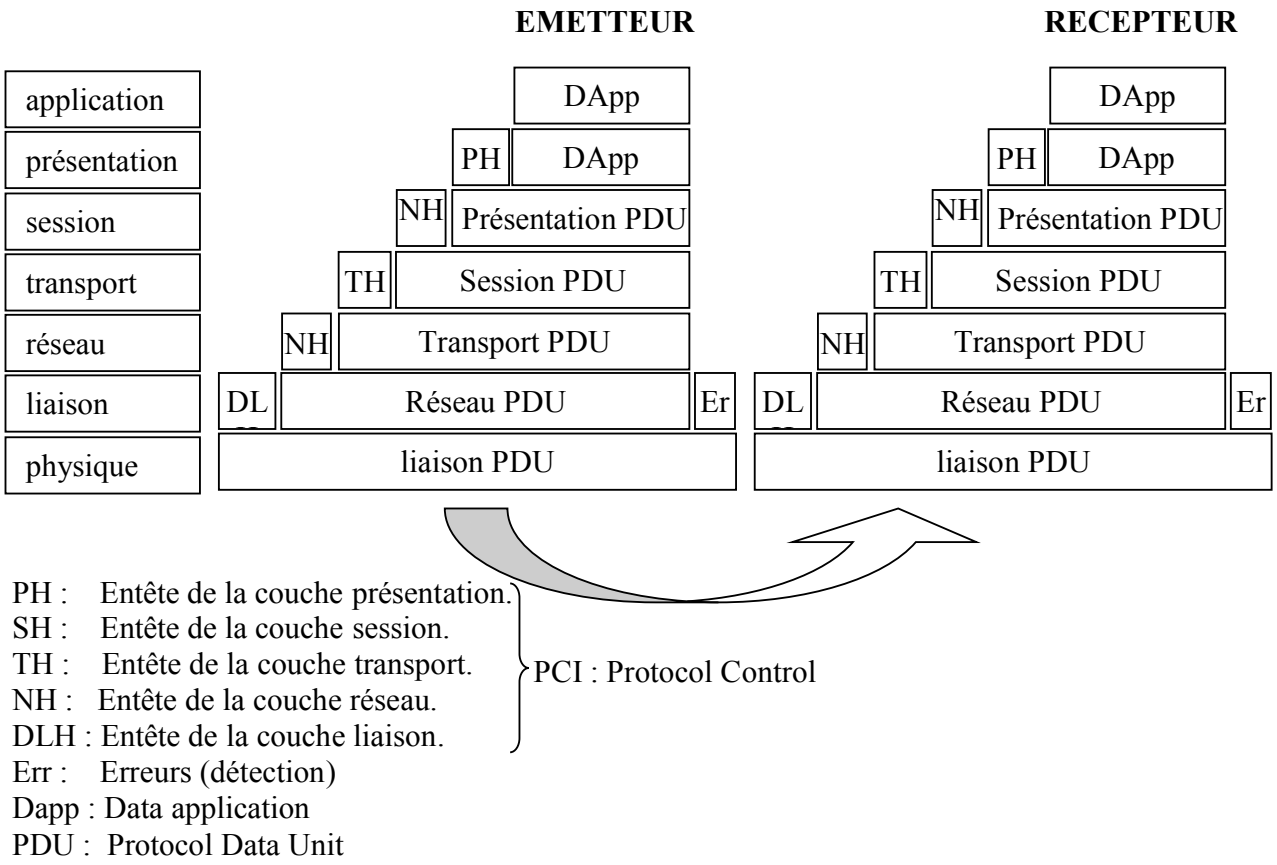
I.1.5.7 Couche 7: Application

C'est la dernière couche du modèle OSI. Les couches de 2 à 5 transportent des octets bruts, la couche présentation s'intéresse à leur syntaxe et la couche application s'occupe de leur sémantique. Cette couche est le point de contact entre l'utilisateur et le réseau. C'est elle qui contient donc l'ensemble des applications qui apportent à l'utilisateur les services de base offerts par le réseau (transfert de fichier, messagerie, transfert de la voix, telnet...).

Quelques protocoles de la couche application : Puisqu'il n'y a pas beaucoup de méthodes qui permettent d'assurer les fonctions des couches 2 à 6, ces couches ont un nombre de protocoles assez restreint. Par contre au niveau de la couche application c'est tout à fait le contraire. C'est pour cela que cette couche contient un grand nombre de protocoles (HTTP, SMTP, SNMP, FTP, Telnet, NFS, Gopher, SSH, NNTP, DNS, XMPP, POP3, IMAP, IRC, VoIP, WebDAV, SIMPLE, ...)

I.1.6 Relation entre couches du modèle OSI

Chacune des couches du modèle OSI a des frontières communes avec ses deux couches voisines. Par exemple entre la couche réseau et la couche transport, il y a une jonction (frontière) que les deux doivent reconnaître. C'est à travers ces frontières que communiquent ces différentes couches. A chque fois qu'une couche 'n' transmet des données à la couche 'n-1' (vers le bas) elle leur ajoute des informations (cf. figure ci-dessous), et à chaque fois qu'elle reçoit des données de la couche 'n-1' elle lui ôte ses propres informations à elle et transmet le reste à la couche 'n+1' (vers le haut).



Transmission des données vers le haut et vers le bas dans l'architecture O.S.I.

En réalité, le modèle OSI est rarement implémenté. Pour diverses raisons, d'autres architectures sont implémentées à sa place (les modèles OSI réduits (EHS, FIP...), l'architecture TCP/IP....). Dans ce qui suit nous verrons le modèle OSI réduit ainsi que l'architecture TCP/IP.

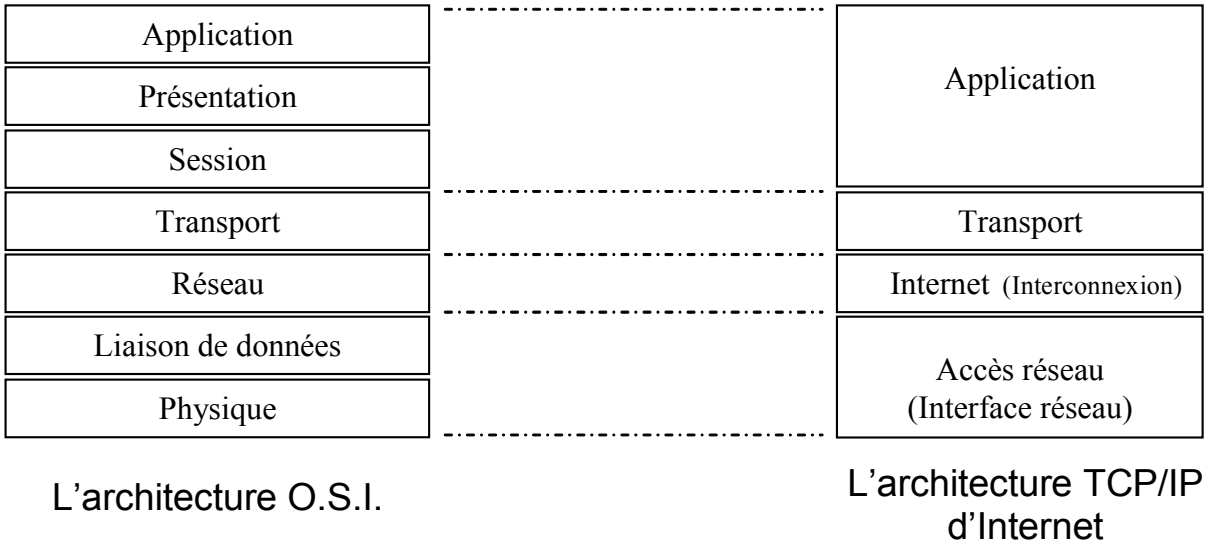
I.1.7 Le modèle OSI réduit

Certains réseaux ont des contraintes très fortes telles que les contraintes de sûreté ou du temps réel. Il doivent alors réduire le nombre de couches à parcourir pour gagner du temps et/ou renforcer d'autres couches pour améliorer et garantir la qualité des communications. On parle dans ce cas alors du modèle « OSI réduit ». Ce modèle contient généralement 3 ou 4 couches qui sont des couches inévitables. Ces couches sont la couche **physique** (nécessaire pour l'envoi de données sur le support de communication), la couche **liaison de données** (gère et contrôle l'accès au médium. Elle est nécessaire pour transformer la couche physique en une liaison exempte d'erreurs. Elle est plus réduite que celle du modèle OSI, car elle n'offre pas de communication en mode connexion), la couche **réseau** (permet à des unités localisées sur des sous-réseaux distants de créer des liens et de communiquer) et la couche **application** (héberge les applications. Elle englobe parfois les couches 5,6 et 7 du modèle OSI).

I.1.8 L'architecture TCP/IP

TCP/IP est appelé par abus de langage modèle. Elle n'est qu'une architecture. Elle prend son nom des deux principaux protocoles qui la constituent, TCP (*Transmission Control Protocol*) de la couche transport qui est utilisé au-dessus du protocole IP (*Internet Protocol*) de la couche réseau. Cette architecture s'est imposée comme modèle de référence au lieu du modèle OSI, car elle est née d'une implémentation, et la normalisation OSI est venue ensuite. C'est son adoption quasi universelle qui a fait son principal intérêt.

TCP/IP est une architecture réseau en 4 couches: couche accès réseau, couche interconnexion (ou Internet), couche transport et couche application. La figure ci-dessous illustre la correspondance entre l'architecture OSI et celle de TCP/IP.



Correspondance entre l'architecture O.S.I et l'architecture TCP/IP d'Internet.

I.1.8.1 Couche 1: Accès réseau

Cette couche regroupe les fonctions des deux couches les plus basses du modèle O.S.I (physique et liaison de données). Elle fournit le moyen de délivrer des données aux systèmes rattachés au réseau.

I.1.8.2 Couche 2: internet (IP ou Interconnexion)

Elle correspond à la couche 3 (réseau) du modèle OSI. C'est la principale couche de cette architecture. Elle réalise 'interconnexion des réseau distant en mode non connecté. Elle se base sur le protocole IP (*Internet Protocol*). Ce protocole a pour but d'acheminer les paquets (datagrammes) indépendamment les uns des autres jusqu'à leur destination. Ces paquets sont routés individuellement dans le réseau. Et puisqu'il n'y a pas de connexion établie, ces paquets peuvent arriver dans le désordre. Les ordonner est la tâche de la couche supérieure. Le protocole IP ne prend en charge ni la détection de paquets perdus ni la possibilité de reprise sur erreur.

I.1.8.3 Couche 4: Transport (TCP)

Cette couche est équivalente à celle du modèle OSI. Elle assure l'acheminement des données, ainsi que les

mécanismes permettant de connaître l'état de la transmission. Elle assure la fiabilité des échanges, veille à ce que les données arrivent dans l'ordre correct, et détermine à quelle application les paquets doivent être délivrés.

Cette couche a deux protocoles distincts, TCP et UDP.

- ***User Datagramme Protocol (U.D.P)***: c'est un protocole particulièrement simple, son seul avantage est un temps d'exécution court qui permet de tenir compte des contraintes de « temps réel » ou de limitation de place sur un processeur. Mais du point de vue sécurité, ce protocole est non fiable, il fournit un service sans reprise sur erreur. Il n'utilise aucun acquittement, ne reséquence pas les messages et ne met en place aucun contrôle de flux. Les messages UDP peuvent être perdus, dupliqués, remis hors séquence ou arrivés trop tôt pour être traités lors de leur réception.
- ***Transmission Control Protocol (TCP)***: Ce protocole a en charge le découpage du message en datagrammes, le réassemblage à l'arrivée avec remise dans le bon ordre, ainsi que la réémission de ce qui a été perdu. A l'inverse du protocole précèdent (UDP), ce protocole fournit une transmission (plus ou moins) fiable, il spécifie comment distinguer plusieurs connexions sur une même machine, et comment détecter et corriger une perte ou une duplication de paquets. Il définit comment établir une connexion et comment la terminer.

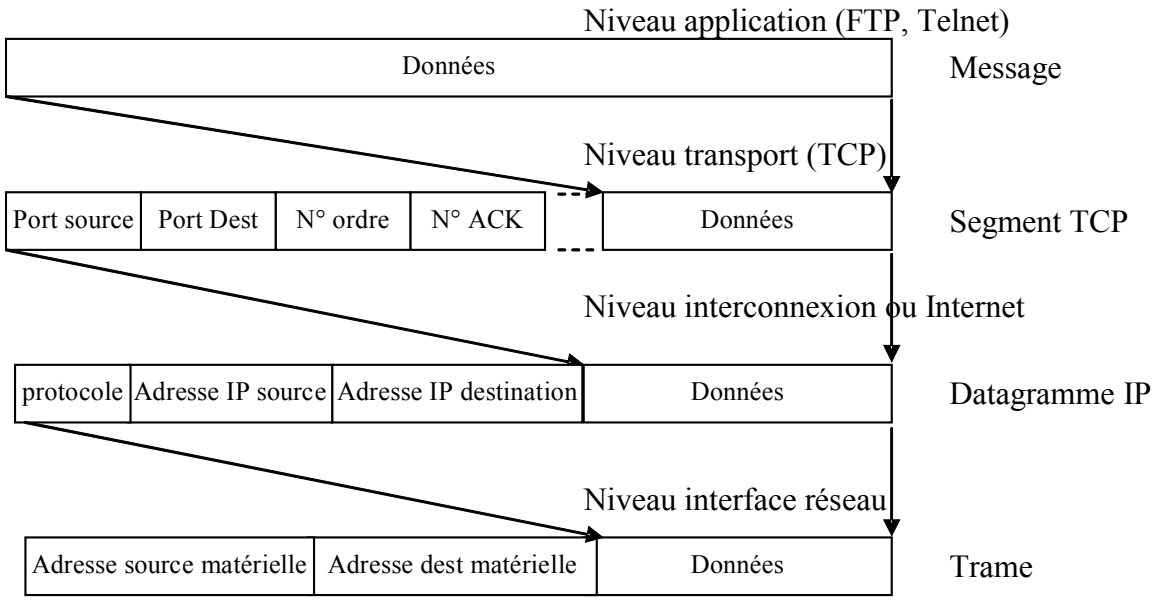
I.1.8.4 Couche 7: Application

C'est l'application la plus riche du point de vue nombre d'applications réseaux et services associés. Elle englobe l'ensemble des couches session, présentation et application du modèle OSI. C'est dans cette couche que se situent la plupart des programmes et protocoles réseaux. Ces programmes fonctionnent généralement juste au-dessus des protocoles TCP et UDP et sont souvent associés à des ports bien définis (par défaut!).
La figure ci-dessous montre les différentes couches de l'architecture TCP/IP et les protocoles qui leurs sont associés.

<table>
<tr><td rowspan="2">N°</td><td rowspan="2">Nom de la couche</td><td colspan="7">protocoles</td><td rowspan="2">Terminologie</td></tr>
<tr><td colspan="5">utilitaires</td><td colspan="2"></td></tr>
<tr><td>4</td><td>Application</td><td>Telnet</td><td>HTTP</td><td>FTP</td><td>PING</td><td>FINGER</td><td>SMTP</td><td>SMNP…</td><td>Messages</td></tr>
<tr><td>3</td><td>Transport</td><td colspan="5">TCP</td><td colspan="2">UDP</td><td>Paquets</td></tr>
<tr><td>2</td><td>Internet (Interconnexion)</td><td colspan="7">IP , ARP</td><td>Datagrammes</td></tr>
<tr><td>1</td><td>Interface du réseau</td><td colspan="7">Ethernet, Token Ring, X25, …</td><td>Trames</td></tr>
</table>

Protocoles de l'architecture TCP/IP

A chaque fois q'un protocole de la couche 'n' reçoit des données de la couche 'n+1', il les encapsule et leur ajoute ses propres données de contrôle tel qu'il est spécifié dans la figure ci-dessous.

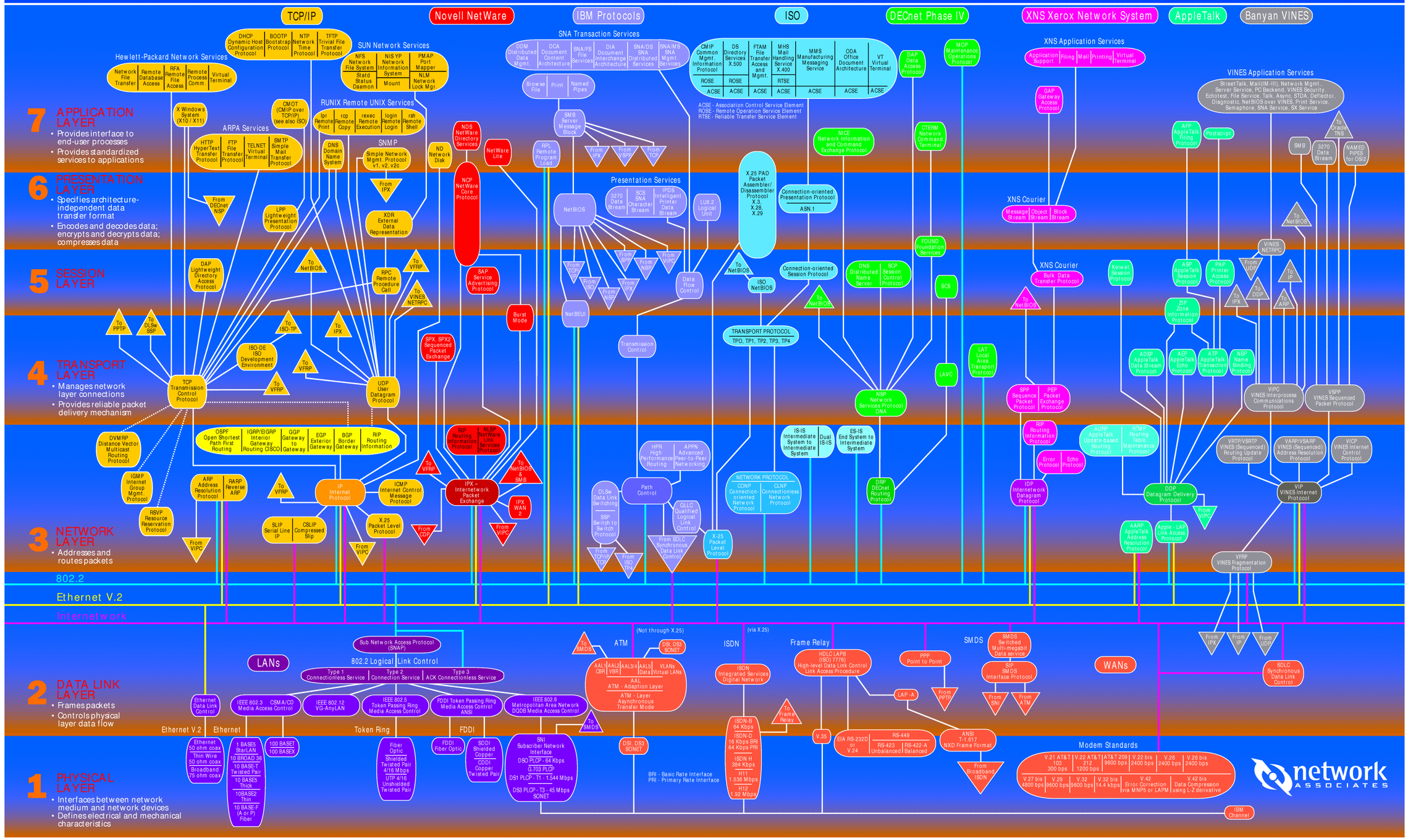


Principe d'encapsulation dans l'architecture TCP/IP

Remarque: La séparation des couches dans la pile IP est nettement plus approximative. Alors que pour être conforme au modèle OSI, un protocole d'une pile ne doit pas dépendre des protocoles des autres couches, mais uniquement du service fourni. Ce point n'est pas respecté dans l'architecture TCP/IP. Il suffit par exemple d'observer son mécanisme de détection d'erreurs. Les deux protocoles TCP et UDP ont dans leur en-tête une somme de contrôle pour la détection des erreurs. Le calcul de cette somme fait intervenir une partie de l'en-tête IP. Les protocoles TCP et UDP ne sont donc pas indépendants de IP. Cela se remarque notamment au fait que lors du passage de IPv4 à IPv6, il a fallu redéfinir la façon de calculer ces sommes de contrôle alors que les protocoles eux-mêmes n'ont pas réellement changés.

- L'architecture TCP/IP nécessite la coopération des Systèmes d'Exploitation des machines distantes dans pratiquement toutes les couches, alors que dans le modèle OSI le Système d'Exploitation n'intervient qu'à partir de la couche 4 (jusqu'à la couche 7).

NETWORK ASSOCIATES GUIDE TO COMMUNICATIONS PROTOCOLS



I.1.9 Architecture des réseaux

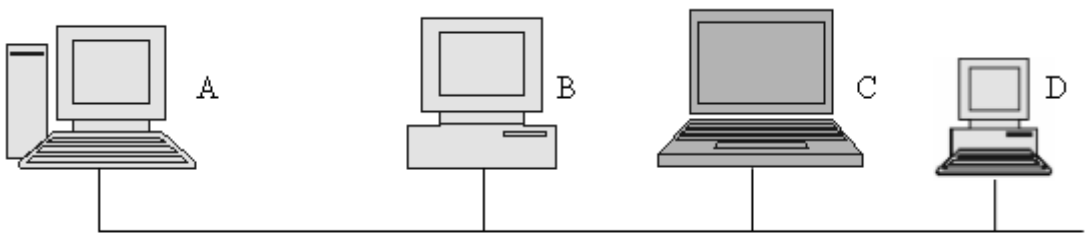
La topologie du système de câblage est souvent influencée par la configuration du site auquel le réseau est destiné; l'étoile est particulièrement bien adaptée à la distribution capillaire. Cependant, on rencontre également des topologies en arbre et en bus dans les réseaux de bâtiment. Les liaisons interbatiment sont généralement des liaisons point-à-point ou des topologies en anneau.

I.1.9.1 Bus

Dans cette configuration (appelée aussi réseau multipoint), toutes les stations sont reliées à un seul câble (généralement coaxial, Ethernet) connecté au serveur. (cf. figure ci-dessous.). Aux extrémités il faut mettre un bouchon (terminateur). C'est une configuration facile à mettre en oeuvre, mais extrêmement fragile, car si un problème survient sur un point (ou une station) du réseau, c'est toute la suite du câble qui sera hors service.

Dans une configuration en bus on utilise souvent le système CSMA/CD (Carrier sens Multiple Access /colision Detection, Accès multiple avec détection de porteuse et de collision). Quand une entité A veut émettre elle se met à écouter le bus (CS). Si une porteuse est détectée (ce qui veut dire que le bus est utilisé), elle attend la fin de la communication, sinon elle émet ses données sur le câble. Durant cette émission A reste en écoute du câble pour détecter une éventuelle collision (CD). Si une collision est détectée, chacune des deux machines concernées suspend immédiatement son émission et attend un certain temps aléatoire avant de réécouter le câble et de réémttre ses données.

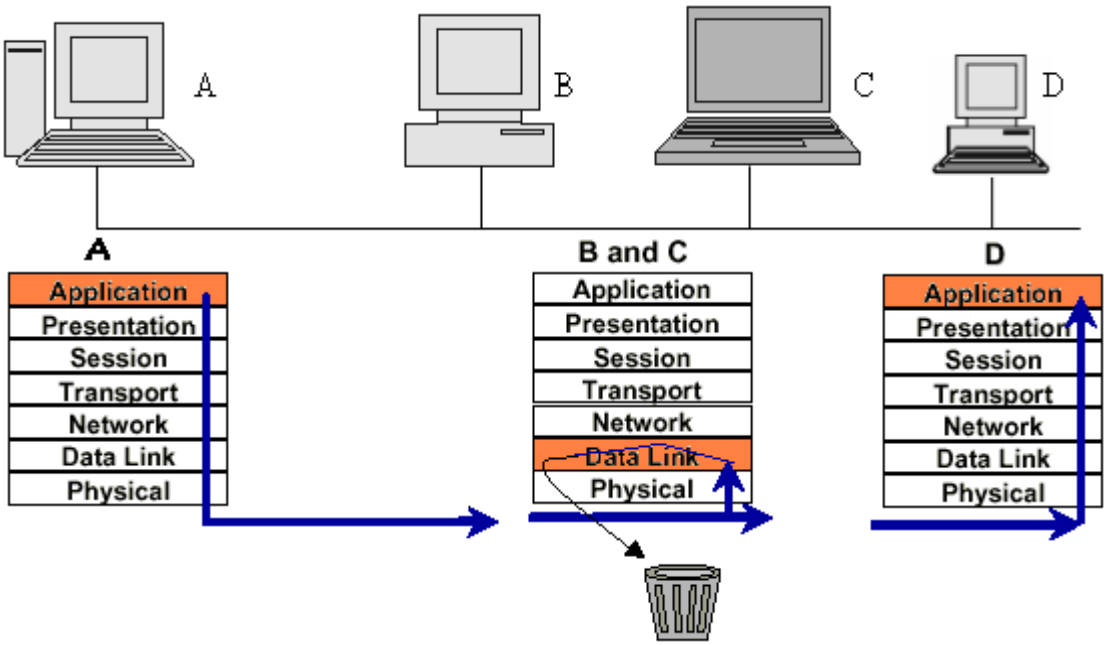
Du point de vue risque, les données envoyées du point A vers le point C peuvent être accessibles au nœud B, et potentiellement altérées ou même déroutées.



Configuration en bus

Quand l'entité A envoie un message à l'entité D, il n'y a que D qui prélève ce message. Les autres le jettent (voir figure ci-dessous). C'est au niveau de la couche 2 que ces machines décident de garder ou de rejeter le message.

Cette topologie était très utilisée auparavant. Aujourd'hui, elle n'est presque plus d'actualité.

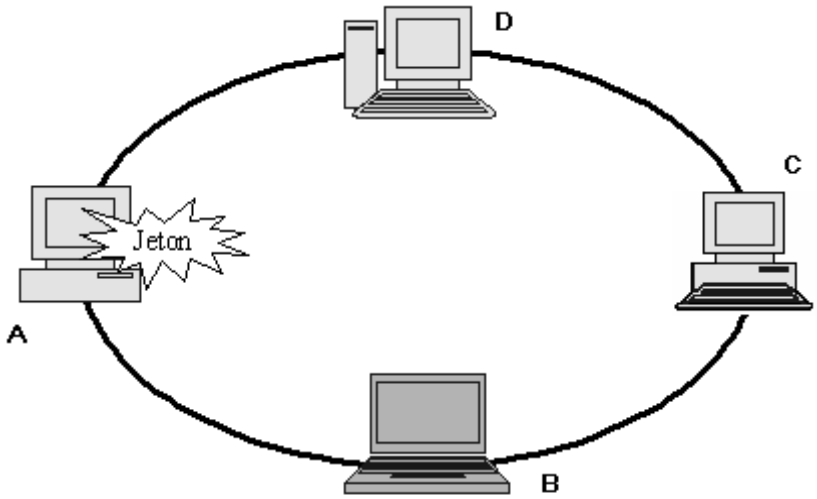


I.1.9.2 Anneau

Dans un réseau en anneau, un seul chemin (double ou simple) relie les nœuds, et le circuit est fermé (cf. figure ci-dessous). Ce réseau est moins cher à câbler qu'un réseau en étoile, car il utilise moins de câble.

Dans une telle topologie l'information circule toujours dans le même sens. Toutes les machines (sauf A) qui reçoivent le message émis par A le recopie immédiatement sur l'autre partie du câble et elle remonte au même temps cette information jusqu'à sa couche 2 pour voir si le message lui est destiné. Si ce n'est pas le cas, elle détruit ces informations. A force de suivre le message et puisque l'anneau est fermé, le message reviendra à la machine qui l'a émis. Celle-ci le compare avec celui qui a été envoyé pour détecter si une erreur est survenue lors de sa transmission. Si aucune erreur n'est détectée, le message est détruit.

Dans cette topologie, chacune des machines doit attendre son tour pour émettre sur le réseau. Pour émettre une machine doit être en possession d'un jeton. Ce jeton est un message particulier que les machines se font passer les une aux autres. Une fois une machine a envoyé ses données, elle rend le jeton disponible et le transfert à la machine suivante. Si la machine qui a le jeton n'a rien à émettre, elle le transmet directement à la suivante. Ce jeton est donc envoyé d'une machine à l'autre dans une boucle circulaire.

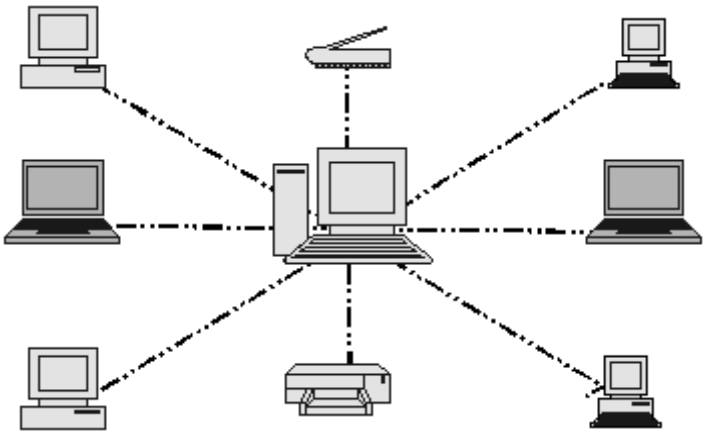


Configuration en anneau

Du point de vue sécurité, contrairement au réseau en bus, une rupture du câble dans un réseau en anneau est facilement contournée dans le cas d'un signal qui voyage dans les deux sens. Mais il présente le risque d'analyseur de protocole. Si une machine envoie le jeton vers une autre bloquée ou éteinte, le réseau sera arrêté. Si, pour une raison ou une autre, ce jeton est perdu, des algorithmes spécifique existent pour sa régénération.

I.1.9.3 Etoile

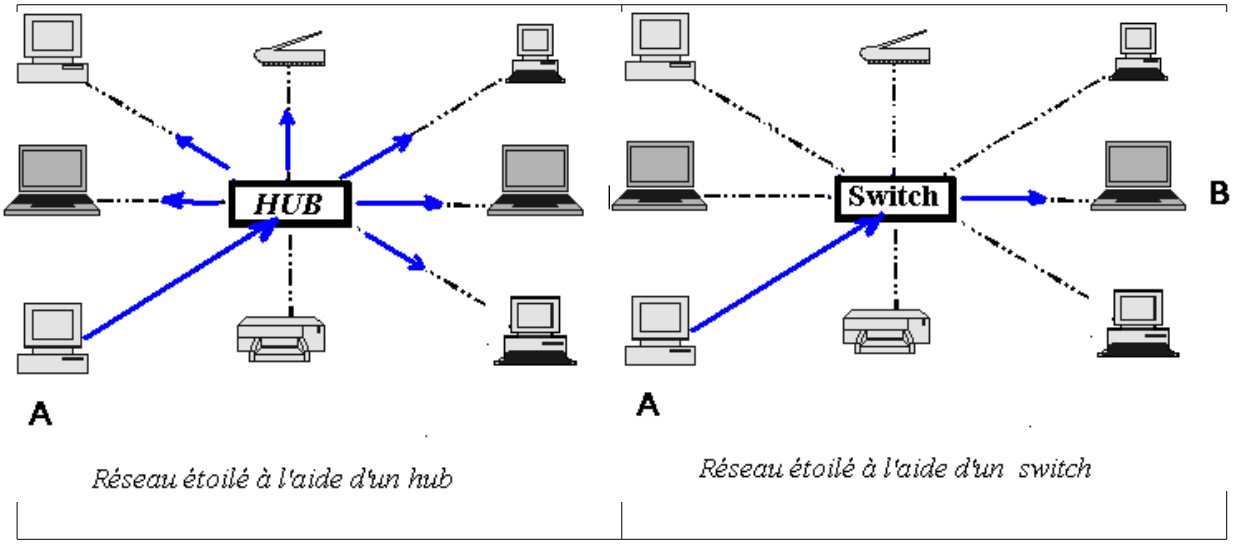
Cette configuration est appelée parfois réseau centralisé, car toutes les communications rayonnent d'un moyeu central (cf. figure ci-dessous). Dans cette architecture, chaque nœud est connecté à ce moyeu et isolé des autres nœuds.



Configuration en étoile

Cette configuration est conçue essentiellement pour réduire le trafic que doivent affronter les machines pour communiquer. Dans les deux configurations précédentes, un message de A vers D doit affronter deux machines mais dans la figure ci-dessous, aucune machine ne doit s'affronter. Le moyeu central, soit il retransmet les messages reçus vers toutes les machines

auxquelles il est connecté (cas de hub (concentrateur)), soit seulement vers la (ou les) machine(s) qui est (sont) destinataire(s) du message (cas d'un switch (commutateur)).



Réseau étoilé à l'aide d'un hub

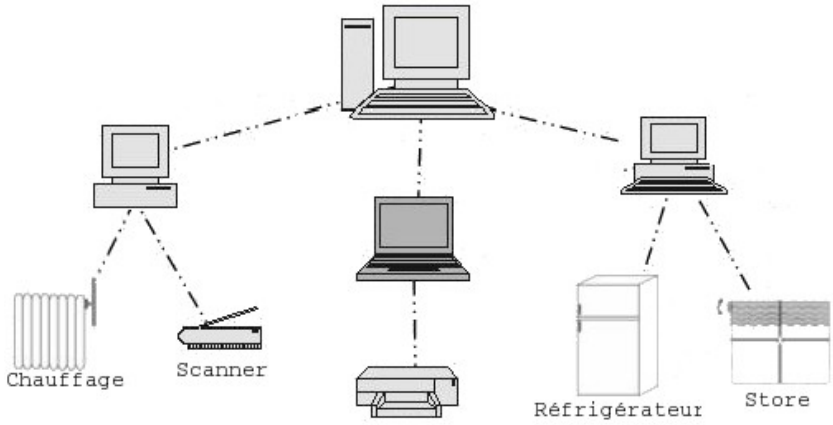
Réseau étoilé à l'aide d'un switch

Du point de vue sécurité, puisque toutes les communications entre deux points sont confinées à un seul chemin et si ce chemin est sûr, les communications sont sûres. Il y a donc moins de risque d'exposition aux attaques par analyseur de réseau. Et de plus, puisque l'étendu du réseau est normalement limité géographiquement, assurer la sécurité physique et empêcher les accès non autorisés est plus facile qu'avec les autres types de réseau. Par contre, on doit contrôler l'accès physique au câblage, ainsi que l'accès physique et logique au serveur et au HUB (points vulnérables du réseau).

A part ces configurations de base, d'autres sont utilisées, mais elles ne sont utilisées que dans des réseaux conçus pour des tâches particulières. Parmi ces topologies:

I.1.9.4 Arbre

C'est une architecture hiérarchisée où les données remontent l'arborescence puis redescendent. Une panne sur une partie du réseau ne touche que les nœuds en dessous (cf. figure ci-dessous).



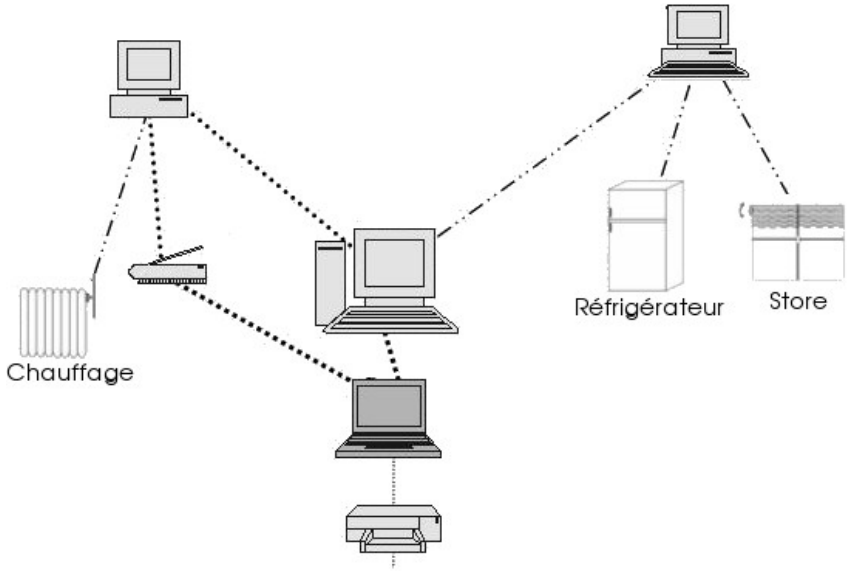
Configuration en arbre

I.1.9.5 Maille

Cette topologie est rarement utilisée (utilisée seulement des laboratoires ou réseaux particuliers), car elle nécessite beaucoup de câblage (n*(n-1)/2 câbles où n est le nombre de machines du réseau).

I.1.9.6 Libre

C'est une combinaison des topologies précédentes (cf. figure ci-dessous).



Configuration libre

I.1.10 Synthèse

Le modèle OSI est la structure réseau la plus étudiée (unanimement reconnue), mais paradoxalement la moins utilisée. Les raisons pour cela sont principalement :

- Apparue trop tard. Dans l'art de la publication des normes, une norme doit être publiée dans le creux qui apparaît entre les recherches effectuées dans ce domaine et les investissements des industriels. OSI est apparue trop tard par rapport à ce moment là; L'architecture TCP/IP est déjà mise en place et largement diffusée.
- Les industriels des réseaux trouvent que le modèle OSI est trop complexe (trop complet). Par exemple les couches 5 (session) et 6 (présentation) sont très rarement utilisées dans les architectures existantes. On peut citer aussi le contrôle d'erreur et de flux qui sont faits par plusieurs couches du modèle OSI, alors que ces services sont optimisés dans l'architecture TCP/IP.
- Les premières implémentations du modèle OSI étaient lourdes et lentes (à cause de la complexité de ce modèle), alors que les premières implémentations de TC/IP dans l'UNIX de l'université Berkeley étaient gratuites, efficaces et rapides. Les gens ont alors opté pour l'architecture TCP/IP.

La synthèse sur les topologies réseaux peut être résumée dans les points suivants:

- La configuration maillée n'est pas utilisé car trop coûteuse,
- Dans une configuration en étoile il faut prendre soin de l'élément central, car s'il tombe en panne (ou on arrive à le corrompre) alors le résultat se répércute sur tout le réseau.

- La configuration en bus n'est plus utilisée dans les réseaux locaux car très fragile,
- La configuration en bus (avec le protocole CSMA/CD) ne convient pas à l'environement temps réél, car deux machines peuvent monopoliser le bus,
- La configuration en anneau (avec jeton) convient à l'environement temps réél, car on peut calculer grace au jeton le délai maximum pour transmettre une information entre deux entités. Cette configuration nécessite plus de câble, car il faut reboucler la dernière machine sur la première.

I.2 Adressage IP et routage

Nous avons dit précédement que la couche réseau a pour rôle de trouver un chemin sur le réseau pour faire communiquer deux machines distantes. Ce chemin est constitué d'une succession de connexions physiques. L'idée se base sur un ensemble de noeuds intermédiaires (routeurs) qui aiguillent les paquets selon leurs adresses sources et destinations. C'est donc au niveau de cette couche qu'il faut ajouter des adresses complètes dans les différents paquets pour qu'ils atteignent leur destinataire.

L'idée initiale de cette couche était de trouver un moyen de communiquer entre les machines du DoD Américains même si un des plusieurs chemins (câbles) qui relient ces machines est coupé (détruit par l'ennemi). Il a fallut alors trouver un moyen qui permet de trouver d'une manière automatique des chemins (routes) entre les différentes machines du DoD, même si certains liens sont coupés. Ce moyen s'appelle le « routage ». Le routage sert donc à trouver, à travers un maillage de noeuds de commutation, le meilleurs chemin qui permet de relier les deux machines communicantes. Les algorithmes utilisés au niveau de cette couche permettent aussi de controler le flux en évitant les embouteillage des paquets (congestion des noeuds, engorgement des sous-réseaux).

Pour acheminer les messages entre les deux machines distantes à travers une succession de connexions physiques, la couche réseau utilse sur chacune de ces connexions les services offerts par sa couche liaison (couche inférieure). Afin de permettre à sa couche supérieure (couche transport) d'assurer de bonnes connexions, la couche réseau lui offre les services suivants:

- Service sans connexion et sans acquittement : On fait appel à ce service quand le support de communication est fiable et/ou le contrôle des erreurs et de flux est assuré par la couche supérieure (transport). Le protocole IP est un exemple de standards offrant ce service.
- Service avec connexion: On fait appel à ce service si par exemple le canal de transmission est moins fiable ou dans le cas où on ne tolère aucune perte de données. A l'aide de ce service on offre aux deux machines communicantes un canal fiable, on garantie la livraison des données transmises, la non duplication des trames et le respect de l'ordre de ces trames. Le protocole X25 est un exemple de standards offrant ce service.

Dans ce qui suit nous verrons les principaux services offerts par la couche réseau, qui sont l'adressage, le routage et le contrôle de la congestion.

I.2.1 Adressage (identification des machines)

I.2.1.1 Adressage (IPv4)

Toute entité (machine, application, page Web) sur le réseau est identifiée par une adresse:

- Les cartes réseaux sont identifiées par une adresse physique (ex. adresse MAC),
- Les machines sont identifiées par une adresse logique (ex. adresse IP),
- Les applications sont identifiées par un numéro de canal (ex. numéro de port),
- Les pages HTML sur le web sont identifiées par un URL (Uniform Resource Locator).

Ces adresses peuvent être:

- « Physiques » ou « logiques »:
  - l'adresse physique identifie de façon unique la connexion d'une machine à un type de réseau (ex. adresse Ethernet),
  - l'adresse logique est indépendante de la machine support et identifie un programme ou un utilisateur (ex. adresse IP)
- « Plates » ou « structurées »:
  - l'adresse structurée peut être décomposée en champs à signification individuelle bien définie éventuellement en rapport avec la localisation géographique (ex. numéro de téléphone, adresse IP...)
  - l'adresse plate n'est pas décomposable en champs significatifs (ex. adresse Ethernet).

Les adresses utilisées dans la couche réseau sont structurées, mais peuvent être physiques (en général ça concerne les adresses X.21[2]) ou logiques (ex. adresses IP). Dans ce qui suit nous nous intéresseront aux adresses IP (adresses logiques structurées).

Comme Internet est un réseau de réseaux (un ensemble de réseaux interconnectés par des routeurs.), l'adressage a une très grande importance. L'adressage dans l'architecture TCP/IP permet de masquer les détails physiques des réseaux et de faire apparaître le réseau Internet comme un réseau unique et uniforme (réseau virtuel). Dans un réseau basé sur cette architecture, chacune des machines est identifiée par une adresse appelée « adresse IP ». Cette adresse doit être unique sur tout le réseau et chaque machine doit avoir une seule adresse IP, sauf si elle dispose de plusieurs interfaces réseaux.

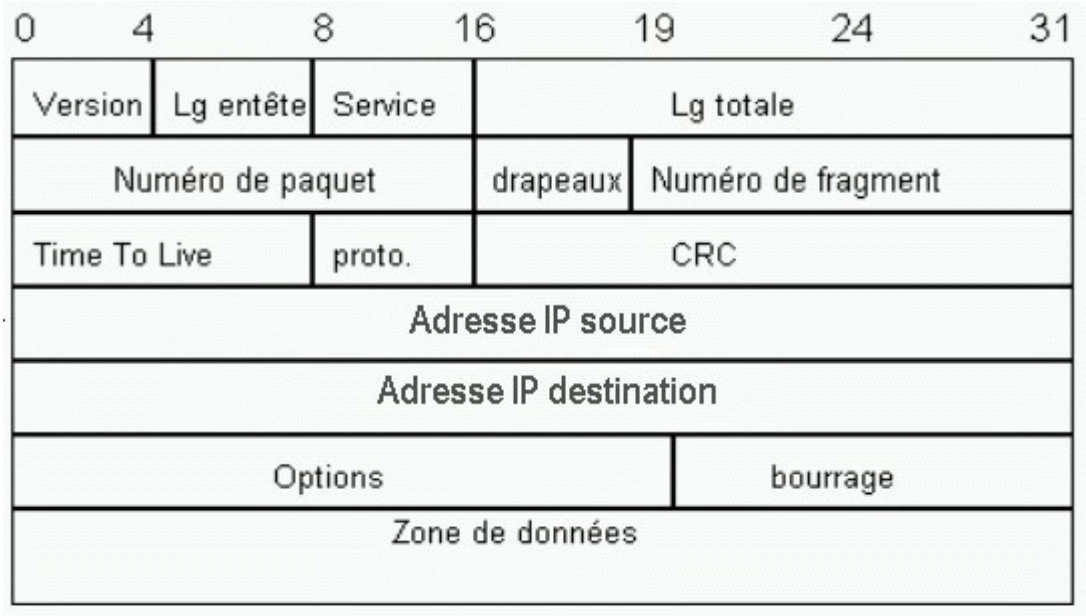
Cette adresse IP permet d'identifier une connexion d'une machine à un réseau indépendamment de l'adresse physique de cette machine. C'est pour cela que Internet apparraît comme un réseau virtuel uniforme.

Même si le support matériel (donc l'adresse matérielle) change, l'adresse IP reste inchangée, car elle détermine de façon permanente un service. Contrairement à l'adresse matérielle[3], l'adresse IP peut être changée et réaffectée à une autre machine.

Le format du paquet IP (datagramme) que nous avons donné précédemment est un format simplifié. La figure ci-dessous est une représentation plus détaillée du paquet (datagramme) IP transmis à la couche interconnexion (dite aussi physique) pour être envoyé sur le réseau.

2 Les adresses X.21 (norme CCITT pour les réseaux publics de transmission de données) sont structurées sur 14 chiffres décimaux, dont 3 pour le pays, 1 pour le réseau à l'intérieur du pays, 10 chiffres pour l'adresse de l'hôte (7 pour la région et 3 pour le numéro local).

3 Certaine carte réseau permettent de changer leurs adresses MAC !



Format détaillé d'un datagramme IP

Ce datagramme contient un en-tête qui a une taille de 5 mots (words). C'est sa taille par défaut, mais il peut aller jusqu'à 15 mots. Cet en-tête stocke notamment l'adresse IP source et destination, le protocole de la couche supérieure, la version IP utilisée. Sur la figure, le champ:

- Version (4 bits) : sert à indiquer la version du protocole IP qui est utilisée (IPv4 ou IPv6).
- Lg entête (4 bits): Le champ Lg entête ou Internet header longth indique la longueur de l'en-tête en mots de 32 bits. S'il n'y a pas d'option utilisée, cette longueur est de 5 mots (longueur par défaut).
- Service (8 bits): Utilisé pour indiquer le type de service. Il définit la priorité du paquet et le type de routage souhaité. Cela permet à un logiciel de réclamer différents types de performance pour un datagramme : délai court, haut débit, haute fiabilité ou bas prix.
- Lg totale (16 bits): Il définit le nombre d'octets contenus dans le paquet (en-tête compris). Le fait que ce champ est codé sur 16 bits, limite la taille d'un paquet IP à une longueur maximale de 65535 octets.
- Numéros de paquet ou identification (16 bits): sert à identifier les fragments d'un datagramme. Il doit donc être unique pour chaque nouveau datagramme.
- Drapeau ou flags (3 bits): sert à contrôler la fragmentation des paquets. Le 1er bt est actuellement inutilisé. Le 2ème bit indique si le paquet peut être fragmenté (bit à 1) ou non (bit à 0). Lorsqu'il est à 1, cela veut dire que le paquet ne peut pas être fragmenté. Dans ce cas, si le routeur ne peut pas acheminer de teks paquets (taille du paquet supérieure à la MTU), il le rejète. Quand le 3ème bits est à 1 cela veut dire que le paquet est un fragment de données et que d'autres doivent suivre. Quand il est à 0, cela veut dire que soit le fragment est le dernier, soit le paquet n'a pas été fragmenté.
- Numéro de fragment ou offset (13 bits): set à indiquer la position qu'occupent les données de ce fragment dans le message original.
- Time to live ou TTL (8 bits): sert à indiquer la durée de vie d'un paquet sur le

réseau. A chaque fois que ce paquet traverse un routeur, son champ TTL est généralement décrémenté de 1. Si ce champ devient nul, le paquet n'est plus relayé et jeté. Ce problème arrive généralement quand le paquet boucle sur le réseau. La durée de vie d'un paquet est donc de 2^8 secondes (un peut plus de 4 minutes).

- Protocole (8 bits): sert à indiquer le protocole de la couche supérieur (transport) auquel il faut transmettre le paquet. Les valeur de ce champ sont: 17 pour UDP (User Datagramme Protocol), 6 pour TCP (Transmission Control Protocol), 1 pour ICMP (Internet Control Message Protocol), 8 pour EGP (Exterior Gateway Protocol), 89 pour OSPF (Open Shortest Path First Routing).
- CRC ou checksum (16 bits): C'est le champ de contrôle de l'en-tête IP.
- Adresse IP source (32 bits): contient l'adresse de l'émetteur.
- Adresse IP destination (32 bits): contient l'adresse du destinataire.
- Options (entre 0 et 40 bits): Ce champ est facultatif. Il sert à stocker des demandes spéciale pour requérir un routage particulier pour certains paquets.
- Bourage (12 bits): Ce champ est facultatif tout comme le champ « option ». Il sert à cacher le contenu du paquet en lui ajoutant des bits qui alignent le flux.
- Zonne de données: Contient les donnée à transmettre.

I.2.1.2 Format et classes d'adresses IP

I.2.1.2.1 Format d'une adresse IP (IPv4):

Les machines d'un réseau TCP/IP ont une adresse IP (version actuelle, IPv4) représentée sur un entier de 32 bits (4 octets). Cette adresse est souvent donnée sous forme de quatre chiffres décimaux séparés par des points (w.x.y.z). Chacun de ces chiffre est compris entre 0 et 255 (ex. 200.113.23.254).

I.2.1.2.2 Adresses IP particulières (conventionnelles):

Il y a quelques adresses qui sont particulières. Elle ne sont donc pas affectées aux machines du réseau. Elle sont utilisées pour signifier ou remplir des tâches particulières:

- Par convention le numéro 0 de la partie hôte (machine) n'est pas attribué. Il est utilisé pour adresser le réseau lui-même et aucun hôte en particulier. Par exemple 192.168.11.0 désigne l'adresse du réseau.
- À l'inverse, une adresse qui a tous les bits de sa partie hôte à 1, désigne l'adresse de toutes les machines du réseaux. Cette adresse est appellée « adresse de broadcast ». Qand on envoie un message à cette adresse, c'est toutes les machines du réseau qui le recevront. Il faut éviter l'utilisation d'une telle adresse, car elle encombre le réseau.
- L'adresse 255.255.255.255 est l'adresse de broadcast général. Elle est utilisée, par exemple, par une machine au moment du boot pour s'autoconfigurer via le DHCP.
- L'adresse 127.0.0.1 (boucle locale) designe la machine elle-même (localhost, c'est l'adresse de la machine qui envoie la requête). Elle ne change pas d'un réseau à un autre.
- L'adresse 0.0.0.0 est utilisée par les machines pendant leur démarrage (procédure BOOT).
- Les adresses qui commencent par 192.168 n'existent par sur Internet. Ce sont des adresse IP privées. Elle sont réservées pour les réseaux locaux (fonctionnant sous TCP/IP).

I.2.1.2.3 Structure d'une adresse IP:

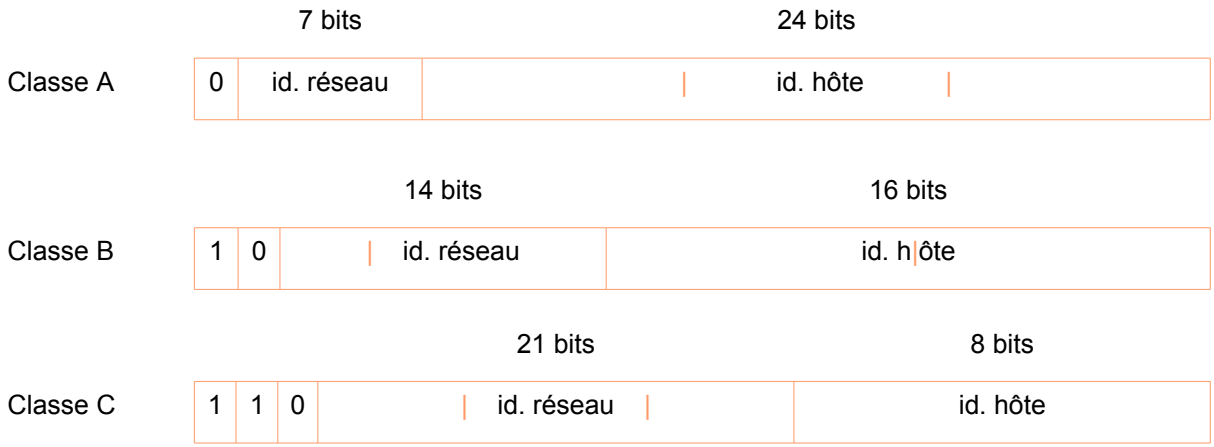
Une adresse IP est constituée de deux parties:

- un identificateur de réseau (Network) qui désigne un réseau parmi tous les réseaux qui sont connectés à Internet.
- un identificateur de la machine (Host) qui désigne une machine parmi toutes les machines du réseau désigné par l'indentificateur réseau.

Le nombre de bits qu'occupe les deux identificateurs (machine et réseau) diffère selon la classe d'adresse utilisée.

Il existe 4 classes d'adresses IP. Chacune permet de coder un nombre différent de réseaux et de machines. Pour cela les identifiants réseau et machine sont codés différement. Pour voir si l'adresse du réseau est codé sur 1, 2 ou 3 octets, il suffit de regarder la valeur du premier octet (w dans la forme w.x.y.z).

1. **Classe A** : w ∈ [0, 127]. Les paquets de cette classe commencent par 0. La longueur du champ adresse réseau est codé sur 1 octet (exactement 7 bits) et celui des hôtes (machines) sur 3 octets (24 bits, cf. figure ci-dessous). Donc on peut avoir dans cette classe 128 (2^7) réseaux et 16 millions (exactement 16 777 216) de machines.
2. **Classe B** : w ∈ [128, 191]. Les paquets de cette classe commencent par 10. La longueur du champ adresse réseau est codé sur 2 octets (moins 2 bits) et celui des hôtes (machines) sur 2 octets. Donc on peut avoir dans cette classe 16 384 (2^{14}) réseaux et 65 534 (2^{16}-2) machines.
3. **Classe C** : w ∈ [192, 223]. Les paquets de cette classe commencent par 110. La longueur du champ adresse réseau est codé sur 3 octets (moins 3 bits) et celui des hôtes (machines) sur 1 octets. Donc on peut avoir dans cette classe 2 millions (exactement 2 097 152=2^{21})) de réseaux et 256 (254=2^8 - 2) machines.
4. **Classe D** : w ∈ [224, 239]. Les paquets de cette classe commencent par 1110. La longueur du champ adresse réseau est codé sur toute la longeure de l'adresse: 4 octets (moins 4 bits = 28 bits). Il n'y a donc pas de place pour le champ hôtes. Cette classe d'adresse s'appelle adresse de groupe. Contrairement aux 3 premières classes qui sont dédiées à l'unicast (point-à-point), cette classe est dédiée pour faire du multicast.
5. **Classe E** : w ∈ [240, 255 (ou 247)]. Les paquets de cette classe commencent par 1111.C'est une classe expérimentale. Elle est réservée pour une utilisation future.



					28 bits
Classe D	1	1	1	0	id. réseau (adresse multidestinataire)

					28 bits
Classe E	1	1	1	1	id. réseau (réservé pour un usage ultérieur)

A partir de la figure précédente on déduit que pour savoir à quelle classe appartient une adresse IP (A, B, C, D ou E), il suffit d'examiner les bits de poids fort de l'octet de poids fort de cette adresse.

I.2.1.3 Masque de réseau:

Nous avons dit qu' une adresse IP est subdivisée en deux, une pour le réseau et l'autre pour la machine dans ce réseau. La partie occupée par l'un et par l'autre est variable selon la classe d'adresse IP utilisée. Le masque de sous-réseau est là pour répondre à cette question (joue le rôle de séparateur entre ces deux parties d'une adresse IP).

Même si on peut choisir une adresse IP indépendament d'un masque, ce couple reste indissociables. Une adresse IP toute seule ne veut rien dire puisqu'on ne saura pas quelle est sa partie réseau et quelle est sa partie machine, et un masque tout seul ne veut rien dire aussi puisqu'il n'y a pas d'adresse sur laquelle l'appliquer.

Avantages et inconvénients du masque:

Avantages :

- Permet de connaître, de déterminer et de limiter les machines qui appartiennent au réseau,
- Permet d'optimiser le fonctionnement du réseau, en segmentant de la façon la plus correcte l'adressage du réseau et de séparer les machines les plus sensibles des autres,
- Permet de limiter la congestion du réseau,
- Permet de prévoir l'évolution du réseau.

Inconvénients:

- Augmente la complexité des tables de routage dans le cas où il y a beaucoup de sous-réseau à router.

Le masque a la même forme qu'une adresse IP. C'est un ensemble de 32 bits où les bits à 1 représentent l'identifiant réseau (la partie réseau de l'adresse), et les bits à 0 représentent l'identifiant machine (la partie machine de l'adresse). Les bits à 1 sont généralement regroupés à part et ceux à 0 régroupés à part. C'est pour cette raison que le masque n'est (généralement) constitué que des octets 255 et 0 (ex. de masque 255.255.0.0).

Ainsi il suffit de faire un ET logique entre l'adresse IP et son masque pour connaître l'identifiant réseau et machine de cette adresse, et donc la classe à la quelle elle appartient.

ex.

Soit la machine M1 qui a pour adresse IP 192.168.34.22 à laquelle est on associe le masque 255.255.255.0

Un ET logique entre ces deux valeur donne 192.168.34.0. Il s'agit de l'adresse réseau auquel appartient la machine M1.

Il n'y a que le premier octet qui est nul. Donc ce réseau ne peut avoir que 255 machines. Les adresses disponibles pour ces machines sont :

```
192.168.34.1
```

```
192.168.34. 2
...
192.168.34.253
192.168.34.254
```

Nota Ben: Les adresses 192.168.34.0 et 192.168.34.255 ne sont pas utilisées (interdites).

- L'adresse 192.168.34.0 (il n'y a que des 0 dans la partie réservée aux machines) est réservée au réseau,
- L'adresse 192.168.34.255 (il n'y a que des 1 dans la partie réservée aux machines) est réservée au broadcast.

Remarque:

Les bits à 1 et à 0 du masque ne sont pas obligés d'être regroupés, mais il conseillé qu'il le soient, car les adresse des machines se suivent, ce qui facilite énormément l'exploitation du réseau.

Par exemple, dans l'exemple précédent si le masque était de 255.255.254.1 (`1111 1111.1111 1111.1111 1110.0000 0001`), nous aurions le même nombre de machines mais les adresses disponibles seraient les suivantes:

- Aux machines spécifiées dans l'exemple précédent, il faut ajouter celles qui ont l'adresse 192.168.35.xxx. (car le 9ème bit peut changer de valeur).
- A cet ensemble de machines ainsi formé, il faut enlever celles qui ont leur dernier octet impair (car le dernier bit doit être à 1). Ce qui revient à dire que ce réseau ne comporte que des machines ayant un nombre pair dans leur dernier octet.

Dans cet exemple nous avons modifié juste un seul bit. Si les bits étaient tous mélangés, la gestion du réseau serait beaucoup plius compliqué.

A partir de cet exemple nous remarqons que la gestion du réseau avec un masque dont les bits ne sont pas contigus est compliquée.

Les octets à utiliser dans un masque (masque valide et masque invalide):

Tous les octets sont autorisés, mais pour éviter la complication, nous avons dit que les bits doivent être contigus. Ainsi les valeurs possibles pour chaque octet sont 11111111, 11111110... 10000000, 00000000. Ce qui donne les octets suivants 255, 254, 252, 248, 240, 224, 192, 128, et 0.

Ainsi le masque 255.255.240.0 est valide alors que le masque 255.255.241.0 ne l'est pas.

Notation CIDR des masques:

Il est simple de déterminer les machines appartenant à un réseau quand le masque ne contient que des 0 et des 255, par contre la tâche devient plus difficile quand l'octet est différent de ces deux nombres (ex. 248, 224, 128...).

Un autre système de notation des masques résout ce problème. Il s'agit de la notation CIDR (Classless InterDomain Routing, ou routage Internet sans classe). Elle note, en décimal, le nombre de bits significatifs (qui sont à 1) de la partie réseau de l'adresse (ie. de l'identifiant réseau). Cette notation suppose elle aussi que la contiguïté des bits est respectée.

Ex.

La notation 192.168.34.0/255.255.255.0 peut être représentée sous cette manière 192.168.34.0/24 (3 fois 255, donc 3 * 8 bits= 24).

La notation 192.168.34.0/255.255.255.248 peut être représentée sous cette manière 192.168.34.0/29

Comment choisir le masque de son réseau?

Le masque doit:

- Prendre en considération le nombre n_1 de machines existantes. Y ajouter 2 (celle du broadcast et celle du réseau).
- Laisser une marge de sécurité pour ajouter d'autres machines (extention du réseau).

  $N= n_1 + n_2 + 2$

Le masque sera ainsi calculé de cette manière

- Prendre la puissance de 2 qui est supérieure à ce nombre afin de pouvoir adresser toutes les machines: $2^y > (n_1 + n_2 + 2)$.
- Retirer le nombre y de 32 (32 est la plus grande valeur dans le masque CIDR): x= 32- y
- Affecter la valeur x au masque CIDR: masque=adresse/x.

Remarque:

- Le masque est aussi en relation avec les limites de ce que l'on a le droit de faire. Par exemple on ne peut pas y aller au delà de la plage qui nous est allouée (par notre fournisseur d'accès par exemple).
- On peut calculer le masque qui convient à un réseau, mais le choix de ce masque reste l'initiative de l'administrateur de ce réseau

Ex.

Si nous disposons dans notre réseau de n_1 machines (n_1 = 60), et de l'adresse réseau suivante 192.168.34.0

- Il faut trouver y tel que $2^y > (60 + n_2 + 2)$: y= 6 si n_2 =2 . Deux (2) machines supplémentaires est très peu. Il faut laisser au moin un intervalle d'une dizaine de machines suplémentaires. On a donc $(n_1 + n_2 + 2)>72$
- Il faut trouver y tel que « $2^y > 72$ ». La résolution de cette équation donne y=7.
- X=32-7=25. C'est à dire nous avons 7 bits à 0 (pour identifier les machines), les 25 autres bits resteront à 1 (pour identifier le réseau).
- Le masque est donc de 192.168.34.0/25, ou 192.168.34.0/255.255.255.128 (11111111.11111111.11111111.10000000).
- La plage d'adresse à définir dans ce cas sera soit [192.168.34.1 – 192.168.34.127], soit [192.168.34.128 - 192.168.34.254]. Elle ne doit pas être par exemple [192.168.34.31 - 192.168.34.158], car les adresses de 192.168.34.1 à 192.168.34.127 auront leur 7ème bit à 0 alors que celles de 192.168.34.128 à 192.168.34.158 auront leur 7ème bit à 1. Selon le masque que nous avons défini, ellles n'appartiendront pas au même réseau. Elles ne pouront alors pas communiquer!!!

Masque des classes d'adresse:

- **Classe A**: le masque associé à la classe A est **255.0.0.0**. Nous avons alors une plage d'adressage allant de 1.0.0.1 jusqu'à 126.255.255.254 soit 16 777 214 adresses possibles.
- **Classe B**: le masque associé à la classe B est **255.255.0.0**. Nous avons alors une plage d'adressage allant de 128.0.0.1 jusqu'à 191.255.255.254 soit 65 534 adresses possibles.
- **Classe C**: le masque associé à la classe C est **255.255.255.0**. Nous avons alors une plage d'adressage allant de 192.0.0.1 jusqu'à 223.255.255.254 soit 255 adresses possibles.
- **Classe D**: le masque associé à la classe D est **255.255.255.240**. Nous avons alors une plage d'adressage allant de 224.0.0.1 jusqu'à 239.255.255.254 soit 255 adresses possibles.

– **Classe E**: le masque associé à la classe E est **255.255.255.240**. Nous avons alors une plage d'adressage allant de 240.0.0.1 jusqu'à 255.255.255.254. Cette classe ne contient que des sous-réseaux, car le champ machine a longueur de 0 bit (n'existe pas).

Le tableau ci-dessous est un résumé sur les classes d'adresses IP:

Classe	Valeur de w (adresse IP=w.x.y.z)	Longueur id réseau	Nbr de réseaux	Nbr max de machines	masque
A	de 0 à 127	1 octet	127	16 777 214	255.0.0.0
B	de 128 à 191	2 octets	16 384	65 534	255.255.0.0
C	de 192 à 223	3 octets	2 097 152	255	255.255.255.0
D	de 224 à 239	4 octets	tout	255, dédiée au multicast	255.255.255.240
E	de 240 à 255	4 octets	tout	Réservée pour une utilisation future	255.255.255.240

Adresses IP privées des classes:

Parfois une entreprise n'a qu'un seul ordinateur connecté à Internet. Les autres ordinateurs de cette entreprise se connectent à celui-ci pour profiter des services rendus par Internet. Dans ce cas seul cet ordinateur dispose d'une adresse IP public, tous les autres auront une adresse IP privée. Les adresses de ces ordinateurs diffèrent selon la classe d'adressage utilisée au sein de l'entreprise:

- Adresses IP privées de classe A : 10.0.0.1 à 10.255.255.254. Elle permet de créer de vastes réseaux privés comprenant des milliers d'ordinateurs.
- Adresses IP privées de classe B : 172.16.0.1 à 172.31.255.254. Elle permet de créer des réseaux privés de taille moyenne.
- Adresses IP privées de classe C : 192.168.0.1 à 192.168.0.254. Elle permet de créer des réseaux privés de petite taille.

I.2.1.4 Evolution de IP (IPv6)

Quand le protocole IPv4 a été développé le nombre d'équipements connecté au réseau Internet était relativement faible, il n'y avait pas beaucoup d'équipements mobiles et les délais de transmission n'avaient pas une grande importance, car il n'y avait pas d'urgence dans la transmission des données. Aujourd'hui les données ont changées et le protocole IPv4 rencontre certains problèmes, notament:

- Manque d'adresses: La version 4 du protocole IP ne s'attendait pas au succès actuel d'Internet. Elle n'a donc pas prévu assez d'adresses pour gérer toutes les machines qui se connecte à Internet. Aujourd'hui, il n'y a pas encore de pénurie d'adresses IP. Cependant, il est certain qu'étant donné le développement rapide d'Internet, on va vite arriver à une situation critique.
- Algorithmes trop complexes: Pour faire face à l'épuisement des adresses IP, les classes d'adresses ont été remplacées par l'utilisation des masques. Les algorithmes et tables de routage sont alors devenus plus complexes que prévus. Actuellement, il n'y a aucune technique de configuration automatique d'espace d'adressage n'a été définie.
- Problème de gestion des appareils mobiles. Le nombre des appareil mobiles est en plein croissance. Chacun de ces appareils doit avoir une adresse IP différente selon l'endroit où il se trouve. Ce qui pose ainsi d'autres problèmes de gestion d'adresses.
- Absence de classes de service: Il n'y a pas de classes de service qui correspondent aux exigences imposées par les nouveaux flots de données tels que la téléphonie ou la vidéo sur IP (ToIP, VoIP), les sessions interactives...
- Augmentation des délais d'acheminement: certaines opérations effectuées dans les routeurs (recalcul du code de contrôle après la modification du champ durée de vie,

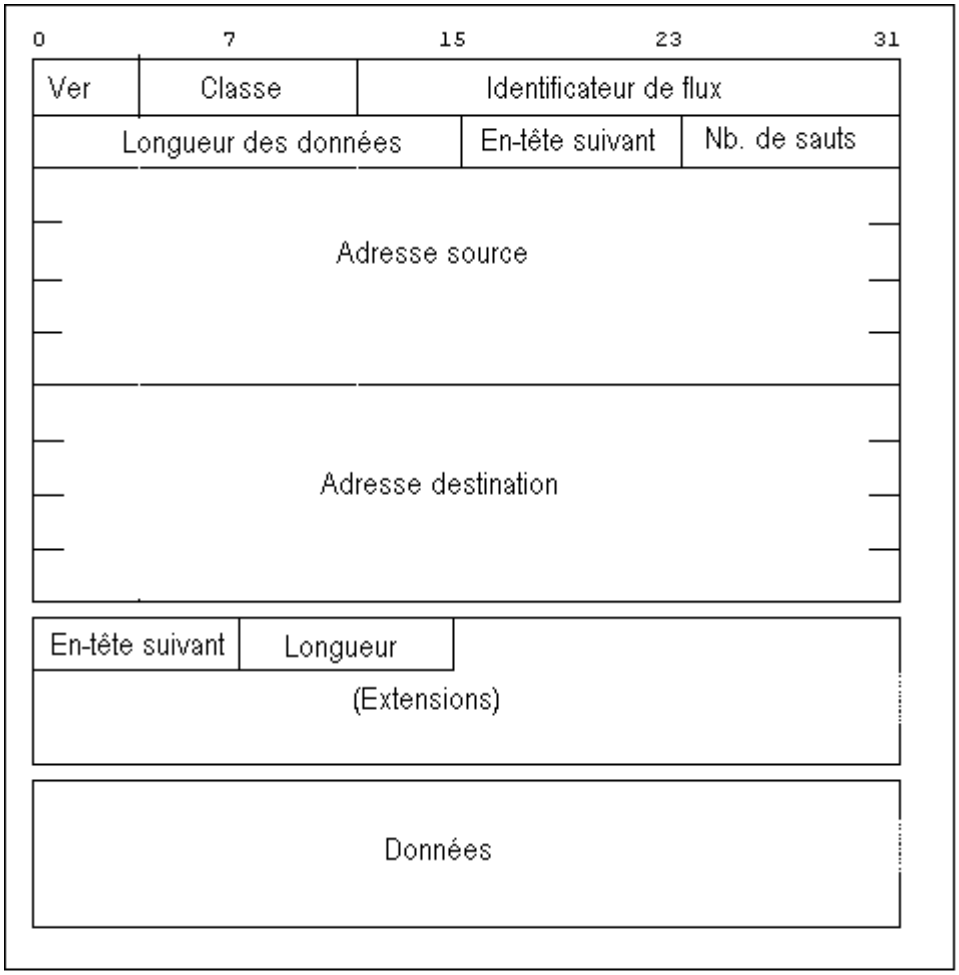
fragmentation/réassemblage des messages) sont très couteuses en temps de calcul et augmentent le délais d'acheminement.

C'est pour toutes ces raisons qu'une nouvelle version d'IP a été créée (en 1995) et commence à être déployée. Cette nouvelle version est baptisée IPv6 (IP version 6) ou IPng (IP nouvelle génération). Parmi les nouveautés essentielles de cette version, on trouve :

- Un plus grand espace d'adressage: 2^{128} au lieu des 2^{32} d'IPv4,
- Automatisation des mécanismes de configuration et de renumérotation, notamment pour les systèmes mobiles. Cette autoconfiguration permet à un équipement de devenir "plug and play". Il suffit de le connecter physiquement pour qu'il acquière une adresse IPv6 et une route par défaut. Si un appareil mobile se déplace, il se reconnecte automatiquement sans interruption de service, tout en gardant la même adresse.
- Un entête simplifié et efficace: Elle est de taille fixe et n'a pas d'options comme c'est le cas dans IPv4. Contrairement à IPv4, il n'y a pas d'options qui sont examinées au niveau des routeur, ce qui facilite et rend efficace le routage.
- Séparation des données privées des informations de gestion des droits numériques[4]. Cette séparation devient de plus en plus essentielle avec l'arrivée de la haute définition,
- Amélioration de la QoS (Qualité de Service, intégration de flux avec des priorités) et du multicast, et intégration du protocole de sécurité IPsec.

La figure ci-dessous représente le paquet IPv6.

4 Ex de gestion des droits numériques: rendre impossible la consultation d'une œuvre (un DVD par exemple) hors de la zone géographique pour laquelle est prévue , rendre impossible la consultation d'une œuvre selon ses préférences (désactivation de l'avance rapide sur certains passages publicitaires de DVD), interdiction ou limitation de la copie ou du transfert des oeuvres...



Format d'un datagramme IPv6

Signification de chaque champ: --> exercice

I.2.1.4.1 Adressage IPv6

Une adresse IPv6 est codée sur 16 octets (128 bits). Ce qui permet d'avoir environs $3,4 \times 10^{38}$ adresses, soit 340 282 366 920 938 463 463 374 607 431 768 211 456 (plus de 42,5 millions de milliards d'adresses par millimètre carré de la surface terrestre).

Avec IPv6 on n'utilise plus la notation décimale. On utilise une notation hexadécimale. Les 16 octets sont regroupés en 8 goupes (de 2 octets) séparés par « : »

Exemple: 1fff:0000:0a88:85a3:0000:0000:ac1f:8001

Cette adresse peut être écrite sous la forme:

1fff:0:a88:85a3:0:0:ac1f:8001

ou: 1fff:0:a88:85a3::ac1f:8001

ou: 1fff::a88:85a3:0:0:ac1f:8001,

Par contre cette écriture n'est pas valide, car elle contient plusieurs substitution (il ne peut y avoir qu'une seule occurrence de la sequence « :: » dans la notation d'une adresse IPv6):

1fff::a88:85a3::ac1f:8001

Remarque:

1. L'adresse IPv6 nulle peut être abrégée en **::0.0.0.0** ou en **::**

2. Pour permettre le déploiement d'IPv6 de la manière la plus flexible possible (transition douce prévue sur 20 ans), la compatibilité avec IPv4 est garantie. Ainsi l'adresse IPv6 peut contenir une adresse IPv4 : on place les 32 bits de IPv4 dans les bits de poids faibles et on ajoute un préfixe de 96 bits (80 bits à 0 suivis de 16 bits à 0 ou 1).

   ex. 0000:0000:0000:0000:0000:ffff:192.168.32.32 = ::ffff:c0a8:2020

   ou 0000:0000:0000:0000:0000:0000:192.168.32.32 = ::c0a8:2020

3. L'adresse IPv6 locale est obtenue automatiquement à partir de l'adresse MAC de l'interface réseau. Par exemple: à l'adresse MAC « 00:02:3F:12:19:B8 » est affectée l'adresse IPv6 suivante: « fe80::2**02:3f**ff:fe**12:19b8** ».
4. Pour utiliser l'adresse IPv6 comme nom de hôte (dans un URL par exemple), il faut l'encadrer avec des crochets (ex. http://**[**1fff:0:a88:85a3:0:0:ac1f:8001**]**/index.html)
5. La manière d'utiliser le masque est la même que dans IPv4 (ex. 2001:6b0:1:1a0::/55)

Quelques adresses particulières sous IPv6:

- Localhost: ::1
- Adresse inconnue: ::
- All-nodes multicast address (adresse de lien s'adressant à tous les noeuds): ff02::1
- All-routers multicast address (adresse de lien s'adressant à tous les routeurs): ff02::2

- http://www.google.fr= http://209.85.135.103/ = http://[::d155:8767]/
- - IPv6 utilise un **adressage hiérarchique** (identification des différents réseaux de chaque niveau) ce qui permet un routage plus efficace.
- http://yahoo.fr = http://217.12.3.11/ = http://[::d90c:03b]/
- http://redhat.com= http://209.132.177.50/= http://[::d184:b132]/

- IPv6 prend mieux en charge le trafic en temps réel (garantie sur le délai maximal de transmission de datagrammes sur le réseau).

I.2.1.4.2 IPv6 et la mobilité

Mobile = station pour laquelle le point de rattachement à Internet change le plus souvent.

....P30Crucianu

--> Exo.

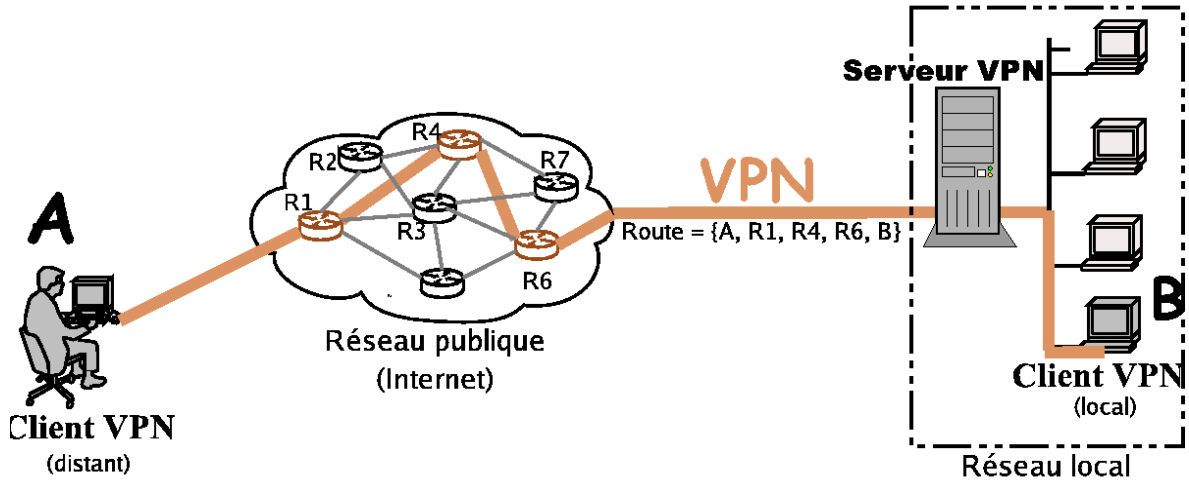
I.2.1.4.3 IPv6 et la sécurité

Contrairement à IPv4 qui fait l'hypothèse que les mécanismes de sécurité sont présents à un niveau supérieur (couche application dans TCP/IP ou couche présentation dans le modèle OSI), IPv6 inclut des mécanismes de sécurité à son niveau.

Interêt: fournir des mécanismes de sécurité supplémentaires au niveau réseau qui seront transparents aux utilisateurs (protection des communications inter-site) en plus des mécanismes offerts par les couches présentation et application (protection des communication inter ou intra-site).

--> IPsec

--> VPN



Donner aux étudiant ma vidéo sur IPv6

I.2.1.5 DNS

I.2.2 Passage des adresses IP aux adresses physiques

Dans l'architecture TCP/IP chaque machine est identifiée sur le réseau par une adresse IP. Il s'agit d'une adresse logique qui ne dépend pas du matériel utilisé pour relier la machine au réseau. Cette machine est aussi identifiée par une adresse physique qui quant à elle dépend du matériel utilisé. Avant de transmettre le paquet IP à la couche physique il faut trouver un moyen de convertir l'adresse IP en l'adresse physique correspondante. Parmi les méthodes qui répondent à ces besoins, il y a :

- la table,
- la conversion directe,
- la conversion dynamique.

I.2.2.1 Table

Avec cette méthode toutes les machines doivent disposer d'une table qui fait la conversion entre adresse logique (ex. adresse IP) et adresse physique (ex. adresse MAC).

Cette méthode est efficace, mais lourde à gérer, car à chaque ajout, suppression ou modification d'une adresse IP pour une machine, il faut remettre à jour cette table sur toutes les machines.

I.2.2.2 Conversion directe

Elle consiste à coïncider tout ou partie de l'adresse physique à l'adresse IP (ex. mettre l'adresse physique de la machine à la place du dernier octet de l'adresse IP). Elle est très facile à mettre en oeuvre certains réseau tels que Pronet[5], mais elle ne se fait pas avec les réseaux Ethernet.

5 Pronet est un réseau basé sur une topologie de type anneau à jeton, généralement câblé sur de la

I.2.2.3 Conversion dynamique (ARP)

Dans cette méthode si une machine connaît l'adresse physique de son destinataire elle lui envoie directement le message, sinon, elle envoie sur le réseau (en broadcast) une demande de résolution d'adresse. La machine qui a l'adresse IP correspondante va lui envoyer une réponse dans laquelle il y a son adresse physique. Ce mécanisme est connu sous le nom d'**ARP** (*Address Resolution Protocol*).

Chacune des machines connectées au réseau local posséde une table de correspondance entre les adresses IP et adresse matérielles (MAC) des machines du réseau local. Cette table doit être souvent mise à jour (car il y a de nouvelles machines qui se connectent au réseau, d'autres qui changent d'adresse IP...). Le protocole ARP (*Address Resolution Protocol*) sert à cela.Si une machine se rend compte que sa table n'est pas à jour (ne connaît pas l'adresse MAC correspondante à une adresse IP) elle envoie (en boadcast pour atteindre toutes les machines y compri les nouvelles) une requête ARP avec l'adresse IP dont on ne connaît pas l'adresse MAC correspondante. La machine qui a cette adresse IP répend en indiquant sont adresse MAC.

I.2.3 Passage des adresses physiques aux adresses IP (résolution inverse, RARP)

Si une machine connaît son adresse physique (ou celle d'une autre machine) et veut obtenir son adresse IP, elle fait appel au protocole RARP (Reverse Address Resolution Protocol). Pour cela, il doit y avoir sur le réseau une ou plusieurs machines (serveur RARP) qui contient des tables (mises à jour à la main) associant des adresses physiques aux adresses IP. Si une machine veut connaître son adresse IP (lors du boot avec un OS sur le réseau par exemple) elle envoie en diffusion sur le réseau une demande RARP. Les serveurs RARP répondent à cette demande en envoyant à la machine son addresse IP. Ce genre de scénario arrive par exemple pour les machines qui démarrent avec un OS situé le réseau. Pour qu'elle puissent obtenir le fichier image de leur boot elles doivent utiliser des protocoles de transfert de fichiers qui sont souvent basés sur TCP/IP. Elle doit donc au préalable connaître son adresse IP.

I.2.4 Routage

Dans cette partie nous verrons comment la couche interconnexion arrive à envoyer ces datagrammes sur le réseau.

I.2.4.1 Introduction au routage

La notion de classe d'adresse n'a pas intervenu jusqu'ici dans le fonctionnement du réseau. Celle-ci intervient au niveau du routage pour voir si deux machines se situent sur le même sous-réseau ou non.

Quand la couche IP d'une machine reçoit des données à envoyer dans un paquet IP, le numéro de réseau du destinataire est comparé au numéro de réseau de la machine locale. Deux cas sont possibles:

1. Les deux machines se situent sur le même sous-réseau local. La couche physique de cet

paire torsadée. Les adresses assignées aux adaptateurs (grâce à de petits interrupteurs) sont comprises entre 1 et 254.

émetteur associe à l'adresse IP destination une adresse physique. A l'aide de cette adresse physique les machines connectées au réseau peuvent rejeter les messages qui ne leur sont pas adressés dès que le message arrive au niveau de la couche physique (sans être obligé de le remonter à la couche internet).

2. Les deux machines ne se situent pas sur le même réseau (sous-réseau) local. Dans ce cas le message est envoyé à la passerelle qui le transmet via le réseau vers la machine destinatrice. A chaque transition, la machine intermédiaire, qui retransmet le message, modifie l'adresse physique du destinataire en mettant celle du noeud suivant (car la couche physique ne voit que les deux machines adjacentes qui sont reliées entre elles par un support physique).

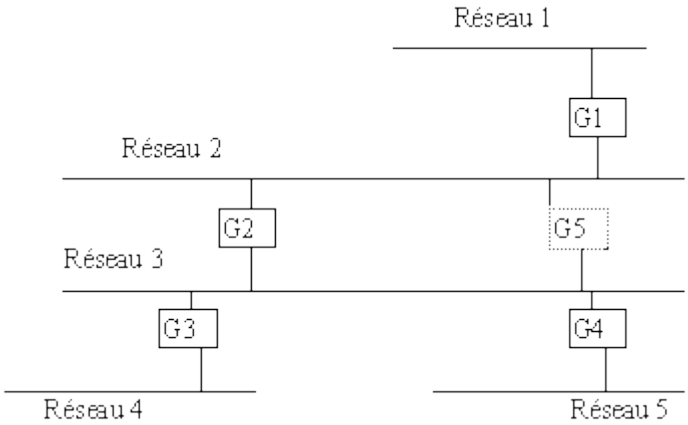
Dans le deuxième cas, la decision d'emprunter telle ou telle passerelle est appelé « routage ».

I.2.4.2 Principe d'un algorithme de routage

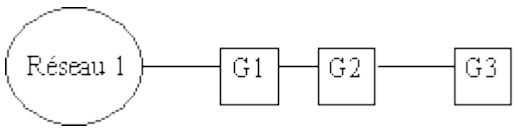
Protocole RIP (Routing Information Protocol). Implémentation logiciel de RIP= « routed ». Ce logiciel est connu car distribué avec l'OS UNIX 4 BSD.

Les passerelles utilise RIP en mode actif et les machines l'utilisent en mode passif.

Pour comprendre le fonctionnement de RIP prenons la figure ci-dessous:



- La passerelle Gl diffusera sur le réseau un message qui contient la paire (1, 1), ce qui signifie que G1 peut atteindre le réseau 1 pour un coût de 1.
- Les passerelles G2 et G5 reçoivent ces informations et mettent leurs tables à jour en créant une route qui passe par la passerelle G1 pour atteindre le réseau 1 (pour un coût de 2).
- Ultérieurement, les passerelles G2 et G5 propagent la paire (1, 2) lorsqu'elles diffusent leur message RIP sur le réseau 3.
- Le cas échéant, toutes les passerelles et les machines créeront une route vers le réseau 1.
  --> la figure suivante représente les routes vers le réseau 1.



Si l'accès de G1 au réseau 1 est coupé (panne) --> G1 affecte une distance infinie (16) à la route correspondante, puis diffuse l'information. Les autres mettent à jour leurs chemins. G1 cherche un autre chemin. ....

Algorithme de routage (simplifié)

Machine émettrice: adresse IP= « @IP1 » , adresse réseau= « IPN1 ».

Machine destinatrice: adresse IP= « @IP2 » , adresse réseau= « IPN2 ».

- Si M1 et M1 appartiennent au même sous-réseau (IPN2 = IPN1) :
  - obtenir l'adresse physique de la machine destinatrice et lui envoyer directement le message,
- Sinon, si une route spéciale est spécifiée dans la table vers @IP2 :
  - router le datagramme vers cette route,
- Sinon, si IPN2 est dans la table de routage :
  - router le datagramme vers cette adresse,
- Sinon, s'il existe une route par défaut :
  - router le datagramme vers la passerelle par défaut,
- Sinon:
  - Déclarer une erreur de routage (ICMP).

I.2.4.3 Protocoles utilisés pour les grands réseaux

RIP = simple à mettre en oeuvre mais ne résout pas tous le problèmes.

De plus, vue le nombre de réseaux, les tables de routage de RIP peuvent devenir énormes.

- Quand une passerelle reçoit un datagramme, elle regarde dans sa table de routage si elle connait la prochaine passerelle pour atteindre le réseau.
  - Si oui, elle lui remet le datagramme (Généralement les passerelles RIP connaissent toutes les routes du réseau local et ignorent les routes vers les réseaux extérieurs.),
  - sinon, elle le remet à une passerele dite « extérieur »
- Les passerelles extérieurs utilisent d'autres algorithmes tels que SPF (Short Path First) et EGP (Exterior Gateway Protocol).

  EGP est souvent mis en place sur des passerelles qui font l'interconnexion de sites avec des réseaux fédérateurs. Principe d'EGP: chaque passerelle ne connait que ses voisins immédiats et met en place une route par défaut sur l'un de ses voisins pour pouvoir router les paquets vers des réseaux qu'elle ne connaît pas.

I.2.4.4 Routage et fragmentation de paquets

Le réseau1 dispose d'un MTU M1, il est connecté au réseau 2, via G1, qui dispose d'un MTU M2, qui ... via Gn-1, qui dispose d'un MTU Mn.
Supposons qu'une machine du réseau 1 envoie un datagramme IP de longueur L à destination d'une machine sur le réseau N, alors 5 cas de figures peuvent se présenter:

- 1°) L < min(M1,M2,...Mn)

alors, le datagramme est émis de passerelles en passerelles jusqu'à ce qu'il atteigne sa destination sur le réseau N.

- 2°) L > min(M1,M2,...Mn)
  alors si le datagramme ne doit pas être fragmenté, un message ICMP d'erreur est émis vers la machine source et le datagramme est détruit par la passerelle qui ne peut pas le faire transiter sur l'autre réseau.
- 3°) L > min(M1,M2,Mn)
  alors si le datagramme peut être fragmenté, la passerelle qui ne peut émettre directement ce datagramme va le couper en autant de petits datagrammes que nécessaire et émettre tous les fragments sur l'autre réseau.
  Lorsque les fragments arrivent sur la passerelle suivante, cette dernière ignore que ce sont des fragments, et les traite comme des datagrammes normaux.
- 4°) le datagramme arrive sur une passerelle qui ne peut le traiter faute de ressources suffisantes, alors ce dernier est détruit sans autre forme de procès.
- 5°) le datagramme arrive sur la passerelle Gi qui ne dispose pas d'information pour router ce datagramme, alors elle le détruit et émet un message ICMP qui signale une erreur de routage.

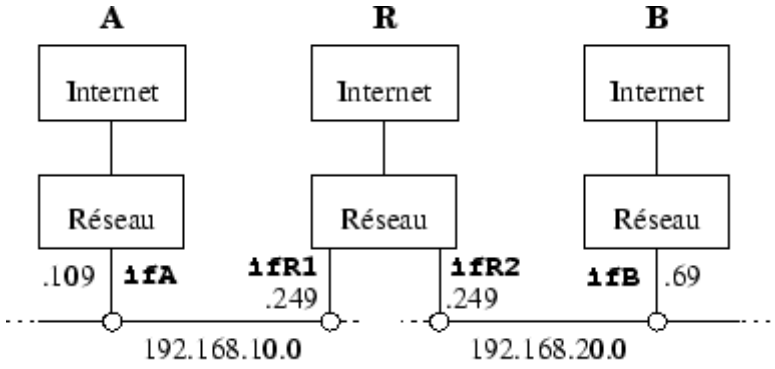
I.2.4.5 Résumé: Comment ça fonctionne?

Nota: Cet exemple est extrait de la documentation de François Laissus "Cours d'introduction à TCP/IP".

Deux réseaux privés sont utilisés dans cet exemple: le réseau 1 a pour adresse 192.168.10.0 et le réseau 2 a pour adresse 192.168.20.0.

Nous faisons l'hypothèse que la passerelle fonctionne comme une machine Unix qui ferait du routage entre deux de ses interfaces !

Ce réseau est illustré par la figure ci-dessous:



Ce tableau résume l'adressage physique et logique de la situation :

Interface	Adresse MAC	Adresse IP
ifA	08:00:20:20:cf:af	192.168.10.109
ifB	00:01:e6:a1:07:64	192.168.20.69
ifR1	00:06:5b:0f:5a:1f	192.168.10.249
ifR2	00:06:5b:0f:5a:20	192.168.20.249

Nous faisons en outre les hypothèses suivantes :

1. Les caches `` arp '' des machines A, B et R sont vides
2. La machine A a connaissance d'une route vers le réseau 192.168.20 passant par

192.168.10.249 et réciproquement la machine B voit le réseau 192.168.10.0 via le 192.168.20.249

3. La machine A a connaissance de l'adresse IP de la machine B

La machine A envoie un datagramme à la machine B, que se passe t-il sur le réseau ?

Étape 1:

La machine A applique l'algorithme de routage et s'apperçoit que la partie réseau de l'adresse de B n'est pas dans le même LAN (192.168.10/24 et 192.168.20/20 diffèrent).

L'hypothèse 2 entraîne qu'une route existe pour atteindre ce réseau, passant par R. L'adresse IP de R est dans le même LAN, A peut donc atteindre R par un routage direct. La conséquence de l'hypothèse 1 implique que pour atteindre R directement il nous faut d'abord déterminer son adresse physique. Le protocole ARP doit être utiliser.

- A envoie en conséquence une trame ARP comportant les éléments suivants :

```
•     SENDER HA   08:00:20:20:cf:af
•     SENDER ADR  192.168.10.109
•     TARGET HA   ff:ff:ff:ff:ff:ff
•     TARGET ADR  192.168.10.249
```

Avec un champ OPERATION qui contient la valeur 1, comme `` question ARP ".

Remarquez qu'ici l'adresse IP destination est celle de R !

Étape 2

R répond à la `` question ARP " par une `` réponse ARP " (OPERATION contient 2) et un champ complété :

```
•     SENDER HA   00:06:5b:0f:5a:1f
•     SENDER ADR  192.168.10.249
•     TARGET HA   08:00:20:20:cf:af
•     TARGET ADR  192.168.10.109
```

Étape 3

A est en mesure d'envoyer son datagramme à B en passant par R. Il s'agit de routage indirect puisque l'adresse de B n'est pas sur le même LAN. Les adresses physiques et logiques se répartissent maintenant comme ceci :

```
•     IP SOURCE   192.168.10.109
•     IP TARGET   192.168.20.69
•     MAC SOURCE  08:00:20:20:cf:af
•     MAC TARGET  00:06:5b:0f:5a:1f
```

Remarquez qu'ici l'adresse IP destination est celle de B !

Étape 4

R a reçu le datagramme depuis A et à destination de B. Celle-ci est sur un LAN dans lequel R se trouve également, un routage direct est donc le moyen de transférer le datagramme. Pour la même raison qu'à l'étape 1 R n'a pas l'adresse MAC de B et doit utiliser ARP pour obtenir cette adresse. Voici les éléments de cette `` question ARP " :

```
•     SENDER HA   00:06:5b:0f:5a:20
•     SENDER ADR  192.168.20.249
•     TARGET HA   ff:ff:ff:ff:ff:ff
•     TARGET ADR  192.168.20.69
```

Étape 5

Et la `` réponse ARP " :

```
•      SENDER HA   00:01:e6:a1:07:64
•      SENDER ADR  192.168.20.69
•      TARGET HA   00:06:5b:0f:5a:20
•      TARGET ADR  192.168.20.249
```

Étape 6

Enfin, dans cette dernière étape, R envoie le datagramme en provenance de A, à B :

```
•      IP SOURCE   192.168.10.109
•      IP TARGET   192.168.20.69
•      MAC SOURCE  00:06:5b:0f:5a:20
•      MAC TARGET  00:01:e6:a1:07:64
```

Remarque, comparons avec le datagramme de l'étape 3. Si les adresses IP n'ont pas changé, les adresses MAC, diffèrent complètement !

Remarque : Si A envoie un deuxième datagramme, les caches ARP ont les adresses MAC utiles et donc les étape **1, 2, 4 et 5** deviennent inutiles. Il n'y aura que l'étape **3** qui sera exécutée..

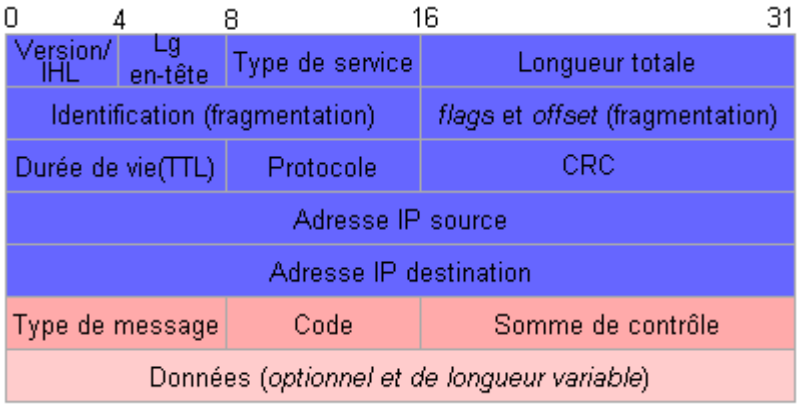
I.2.5 ICMP et contrôle d'erreur

ICMP (*Internet Control Message Protocol, RFC 792*) est un protocole de la couche réseau, qui accompagne le fonctionnement du protocole IP. C'est grâce à lui qu'une machine émettrice peut savoir qu'il y avait eu un problème de transmission (réseau, protocole ou port inaccessible, réseau ou machine inconnue, arrosage de la source (volume trop important), pas d'information de routage, durée de vie dépassée...). ICMP permet de contrôler les erreurs de transmission, car IP ne gère que le transport des datagrammes et ne permet pas l'envoi de messages d'erreurs.

ICMP est donc utilisé principalement pour créer des rapport pour les erreurs et informer la machine concernée. Il permet donc:

- d'informer l'émetteur de l'impossibilité d'atteindre la destination, et donc de livrer le datagramme (paquet),
- d'informer l'émetteur que la durée de vie du datagramme envoyé est expirée,
- d'informer l'émetteur qu'il y a eu des erreurs dans le format du datagramme transmis (en-tête erroné),
- d'informer un noeud que le noeud suivant a épuisé sa capacité de stockage de datagrammes,
- d'aider à la configuration automatique des équipements (redirection ICMP, découverte des routeurs...),
- de répondre avec une réponse d'echo (par exemple pour la commande ping),
- d'envoyer ou de demander l'heure et/ou la date système de la machine,
- de demander/donner une adresse IP ou un masque de sous-réseau,
- ...

Mais il pourait y avoir une perte de paquets sans se rendre compte, donc sans la réception de message ICMP.
Le message ICMP est encapsulé dans un datagramme IP. Le format d'un datagramme ICMP ainsi formé est représenté par la figure ci-dessous:



Format d'un datagramme ICMP

Note: Une nouvelle version d'ICMP est associée à la nouvelle version d'IP, IPv6. Il s'agit de ICMPv6. Ce dernier est identifié dans l'en-tête IPv6 par la valeur 58 dans le champ « En-tête Suivant ». ICMPv6 est un protocole générique; il est utilisé, par exemple, pour rapporter des erreurs trouvées dans le traitement de paquets, pour effectuer des diagnostics ou une découverte de voisinage, et pour rapporter l'appartenance à un multicast. Il intégre donc en plus des fonctions d'IMPv4 les fonctions de gestion des groupes du multicast du protocole IGMP (*Internet Group Message Protocol*) et les fonctions du protocole ARP (*Address Resolution Protocol*) d'IPv4

I.3 Protocoles de la couche transport

Dans la partie précédente nous avons vu comment fonctionne une machine pour envoyer des informations à une autre machine distante (routage). Ces informations en réalité ne sont pas destinées à une machine tout court, mais plus exactement à une application s'exécutant sur cette machine. Le but de cette section (des protocoles de la couche transport de la pile TCP/IP!) est de déterminer cette application.

Dans l'architecture TCP/IP, la couche transport est composée de deux protocoles: TCP et UDP. On va donc décrire dans ce qui suit chacun de ces protocoles.

I.2.2 Protocole TCP

I.1.1.1 Protocole TCP dans l'architecture TCP/IP

TCP (*Transport Control Protocol*, RFC 793) est un protocole de la couche transport. Il est fiable (plus ou moins fiable) et orienté connexion. Il permet l'acheminement sans erreur des paquets. Il est dit "orienté connexion", car il ouvre une session entre deux ordinateur distants, la maintient et effectue lui même le control d'erreur.

Au niveau de l'émetteur, il fragmente le message à transmettre de façon à pouvoir le faire passer sur la couche internet. Au niveau du récepteur, il rassemble ces fragments et les remet dans l'ordre afin de reconstruire le message initial. C'est le champ "protocol" du paquet IP qui indique au protocole IP à quel protocole de la couche transport le message doit être transmis. Si ce champ contient la valeur 6, le message sera transmis au protocole TCP, et si cette valeur est 17, le message sera transmis au protocole UDP.

I.1.1.2 Principe de fonctionnement de TCP

Une connexion TCP est identifiée par la paire de **sockets (source, destination)**. Une socket se compose d'une adresse IP et d'un numéro de port. C'est à l'aide de ce dernier (numéro de port) qu'on identifie le processus de niveau supérieur (service de la couche application) qui demande un service

TCP. L'ouverture d'un port peut être passive (pour le serveur qui va attendre une demande de connexion) ou active (pour le client qui éffectue une demande de connexion sur le port d'un serveur qui généralement a déjà ouvert une connexion passive). Un port est vu comme un lieu de rendez-vous. Ainsi le programme serveur demande à son Système d'Exploitation de lui donner toutes les informations qui arrivent sur ce port. Le client qui veut dialoguer avec ce serveur demande aussi à son Système d'Exploitation de lui donner un numéro de port. Il va envoyer ses données à partir du port qui lui a été réservé (port source) vers le port qui écoute sur la machine du serveur (port destination). Les numéros de ports vont de 1 à 65535. Pour que le client sache le numéro de port qui est attribué à un service au niveau du serveur, la norme l'autorité IANA (*Internet Assigned Numbers Authority*) a réservé des numéros de port par défaut à ces services. Ce sont des numéro inférieurs à 1024 (On peut toujours modifier cette association, mais cela nécessite les droits root). Le tableau ci-dessous donne quelque services avec les numéros de port qui leur sont attribués:

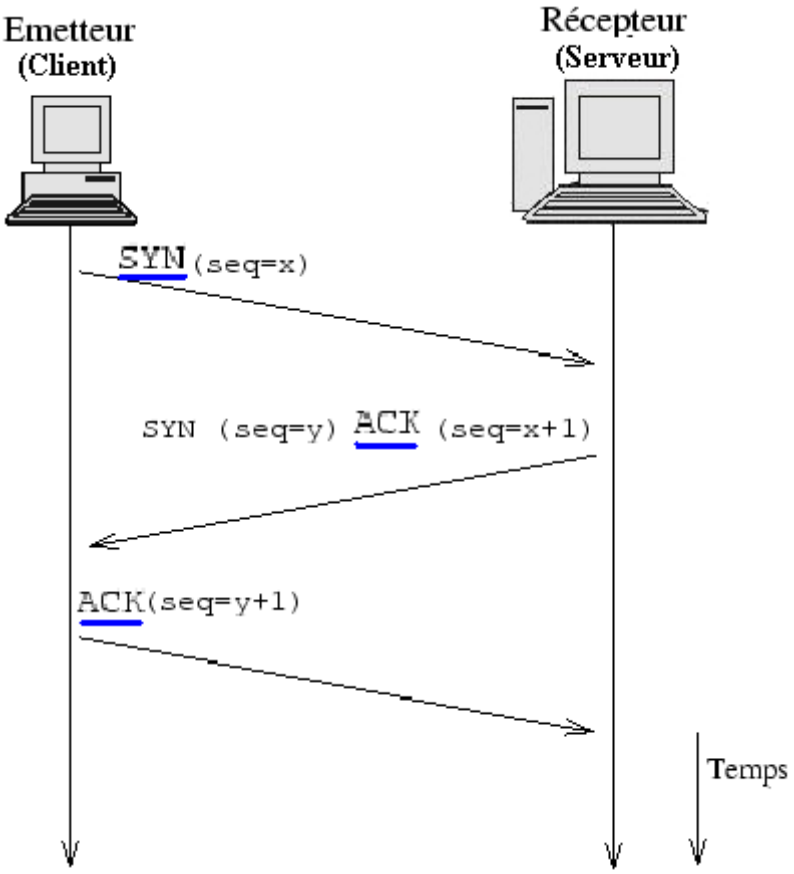
Service	Numéro de port qui lui a été réservé	Service	Numéro de port qui lui a été réservé
HTTP	80	FTP	20 et 21 (20 pour les données et 21 pour les commandes)
HTTPS	443	RPC	111
SMTP	25	Finger	79
POP3	110	DNS	53
Telnet	23	DHCP (BOOTP)	68 (BOOTP 67 et 68)
SSH	22	DayTime	13

Une communication TCP s'effectue en 3 phases: l'établissement de la connexion, le transfert de données et la fermeture de la connexion.

1. **Etablissement de la connexion**:

   L'établissement de la connexion se fait par une **poignée de main** en trois temps:

   - L'émetteur (appelé client) envoie au récepteur (appelé dans ce cas serveur) un segment comportant le drapeau SYN (voir figure ci-dessous), avec sa séquence initiale (ISN=x). Dans ce cas, on dit que l'émetteur éffectue une « ouverture active » (active open).
   - Le serveur qui est déjà en écoute (ouverture passive, ouverture d'une socket serveur) répond avec sa propre séquence (ISN=y) tout en acquittant le paquet précédent (ACK, seq=x+1),
   - Le client acquitte le deuxième segment (seq=y+1).



Ouverture d'une connexion TCP

2. **Transfert de données**:

   Le transfert de données commence par un échange de quelques octets qui justifie l'établissement de la connexion. C'est après cela que commencent les vrais échanges de données. Pendant la phase de transferts de données, certains mécanismes permettent d'assurer la robustesse et la fiabilité de TCP. Parmi eux les checksums, le numéro de séquence, le numéro d'acquittement et les temporisations.

   - Le checksum permet la détection des erreurs dans un segment,
   - l'acquittement, ainsi que la temporisation, permettent la détection des segments perdus ou retardés. L'émetteur utilise des timers afin de réémettre automatiquement les segments pour lesquels l'acquittement s'est fait attendre trop longtemps.
   - Le numéro de séquence est utilisé afin d'ordonner les segments TCP reçus et de détecter les données perdues. Un seul exemplaire est gardé d'un segment reçus correctement plusieurs fois. Le numéros de séquence représente le propre numéro de séquence de l'émetteur TCP, alors que le numéro d'acquittement représente le numéro de séquence du destinataire.

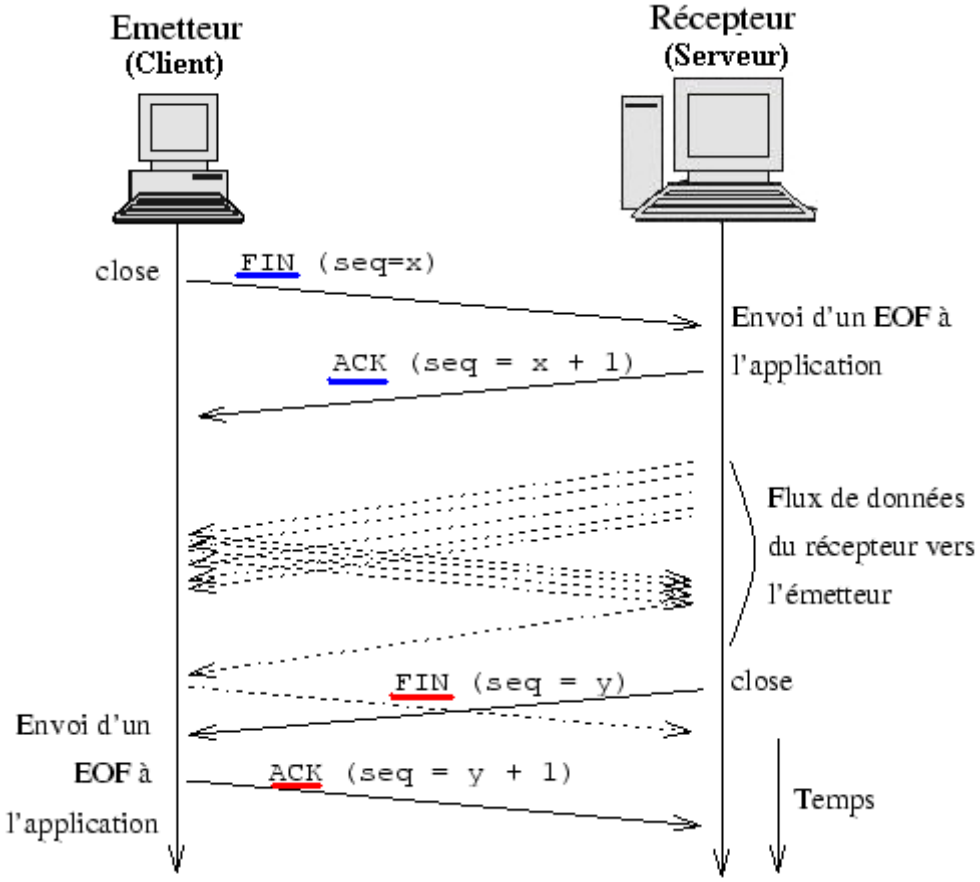
   Afin d'assurer la fiabilité de TCP, le destinataire doit acquitter les segments reçus en indiquant qu'il a reçu toutes les données du flux d'octets jusqu'à un certain numéro de séquence. Ainsi grâce à cet acquittement et aux numéros de séquences, le système distant (destinataire) peut ordonner les segments, reconstituer les données originales et les remettre à l'application destinatrice.

Un système de « fenêtres glissantes » (voir plus loin) sur le bloc de données à envoyer est aussi utilisé pour optimiser le transfert de données.

Des recherches sont actuellement menées pour d'améliorer le protocole TCP. Leur but est un traitement plus efficacement des pertes, une minimisation des erreurs, une meilleur gestion de la congestion et être rapide dans des environnements très haut débit.

3. **Ferméture de la connexion**:

La phase de terminaison d'une connexion (cloture canonique (*orderly release*) et non pas abrute (*abortive release*)) utilise une **poignée de main en 4 temps**. Chaque extrémité de la connexion effectue sa terminaison indépendamment de l'autre. La raison pour cela est que la connexion TCP est « full duplex ». Les données circulent alors dans les deux sens d'une manière indépendante l'une de



Ferméture d'une connexion TCP

l'autre. La fin d'une connexion nécessite alors une paire de segments FIN et ACK pour chaque extrémité. La machine (ou plutôt l'application) qui envoie un paquet avec le drapeau FIN indique à la couche TCP de la machine distante qu'elle n'enverra plus de données. La machine distante doit alors acquitter ce segment en incrémentant d'une unité le "séquence number" (voir figure ci-dessous). Quand le client envoie FIN, le serveur répond par ACK, mais il va continuer à envoyer les données restantes. A la fin de l'envoi de données, il envoie lui aussi au client un segment avec le drapeau FIN (avec un seq=y). Le client acquitte ce segment en envoyant un ACK avec un numéro de séquence=y+1. C'est à ce moment là que la connexion est considérée définitivement close.

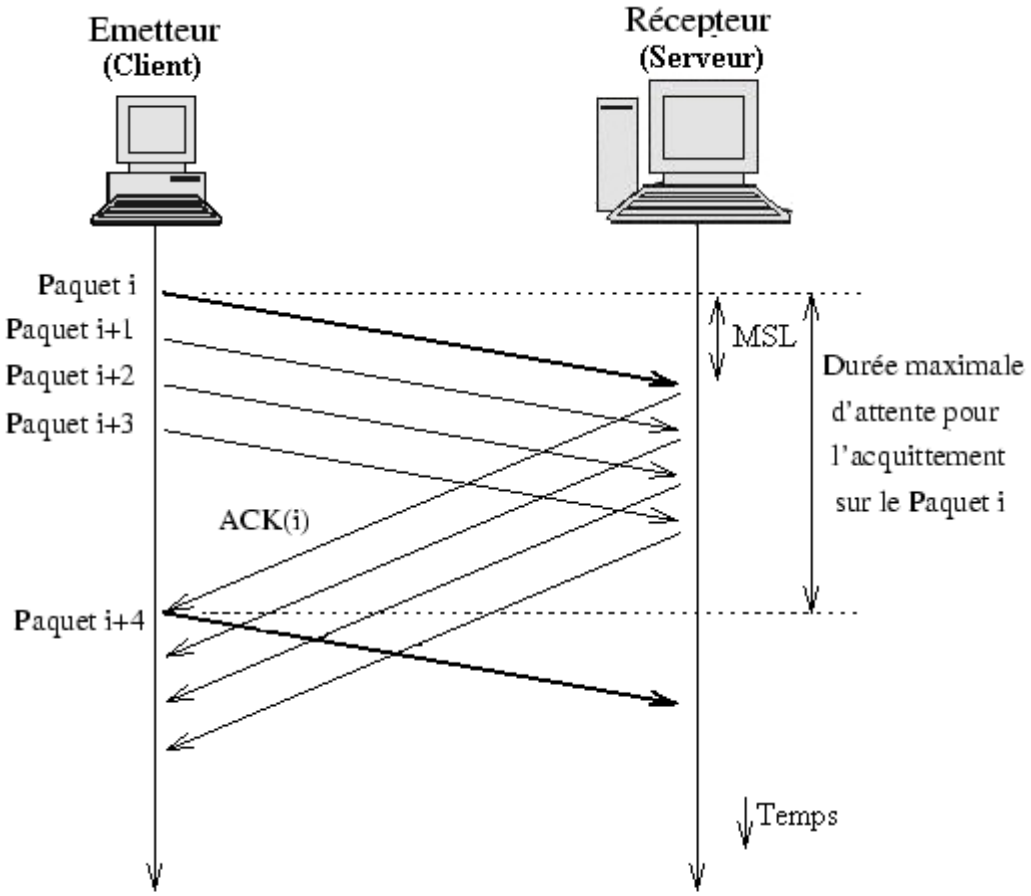
Remarque: Parfois des fermétures brutales de connexion se produisent (application brutalement

interrompue sans avoir fait appel à la primitive 'close'). A ce moment là, au lieu d'avoir un échange de 4 paquets comme précédement, c'est un mécanisme de reset qui termine rapidement cette connexion (*abortive release*).

I.1.1.3 Système de fenêtres glissantes

Le mécanisme d'acquittement devrait obliger l'émetteur d'attendre la réception de l'acquittement ACK avant d'envoyer le prochain segment. Cette attente est appelée RTT[6] (RTT=2*MSL[7]). L'utilisation de l'acquittement de cette manière va pénaliser la bande passante (sera toujours vide). Il faut à chaque fois attendre un temps RTT avant d'envoyer la prochaine information. Le système de « fenêtres glissantes » ("*Sliding Windows*") est là pour contourner ce problème et améliorer ainsi l'utilisation de la bande passante.

Il permet à l'émetteur d'envoyer en continu des données sans attendre les précédents accusés de réception. Il permet aussi au récepteur de recevoir les paquets dans le désordre et de profiter de ces délais d'attente pour réorganiser ces paquets.



Principe de la fenête coulissantes

La fenêtre définit le temps ou le volume de données susceptibles d'être passées via une connexion TCP avant que l'émetteur ne reçoive un accusé de reception. Si l'émetteur ne reçoit pas d'accusé de

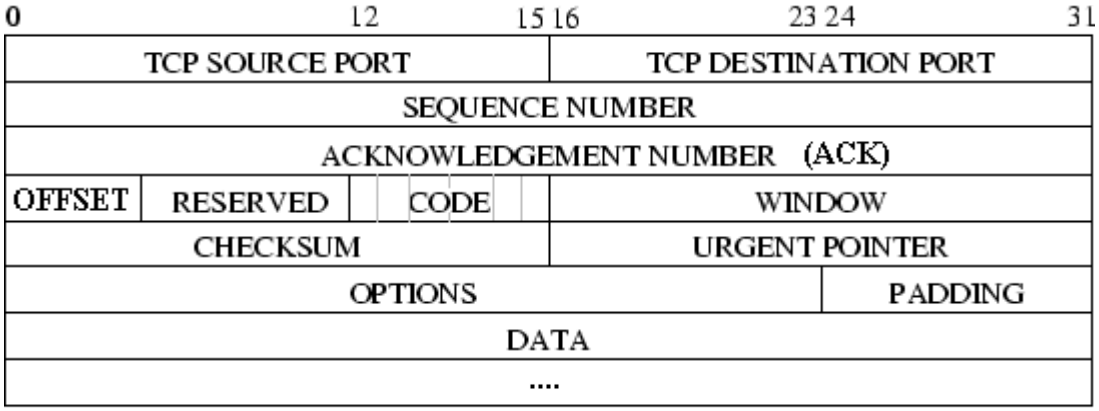
6 RTT (*Round Trip Time*).

7 MSL (*Maximum Segment Lifetime*), c'est le temps maximum que prend un segment pour traverser le réseau. MSL = somme du temps de transit entre chaque routeur et du temps passé dans les diverses files d'attente sur les routeurs.

réception au bout de du temps défini par la longueur de la fenêtre il réemet le paquet concerné. Cette longueur est précisée par le champ WINDOW de l'en-tête du segment TCP (voir ci-dessous).

I.1.1.4 Structure d'un segment TCP

La structure d'un segment TCP est donnée par la figure ci-dessous:



Structure d'un segment TCP

Signification des champs de l'en-tête d'un segment TCP:

- « **TCP SOURCE PORT** »: correspond au numéro de port de l'application locale.
- « **TCP DESTINATION PORT** »: correspond au numéro de port de l'application distante.
- « **SEQUENCE NUMBER** »: Ce champ contient un nombre qui identifie la position des données à transmettre par rapport au segment original. Il correspond au numéro du paquet.
- « **ACKNOWLEDGEMENT NUMBER** »: Ce champ contient un nombre qui identifie la position du dernier paquet (octet!) reçu dans le flux entrant. Il accuse la réception du paquet dont il porte le numéro. Si sa valeur est par exemple x, cela veut dire que tous les paquets inférieurs à x ont été bien reçus. Ce champ doit être accompagné du drapeau ACK.
- « **OFFSET** »: Ce champ définit le nombre de mots de 32 bits dans l'en-tête TCP. Il indique donc dans le segment TCP le lieu où commencent les données.
- « **RESERVED** »: Ce champ est réservé pour un usage ultérieur.
- « **CODE** »: C'est un champ de 6 bits qui sert à influer sur le comportement de TCP en caractérisant l'usage du segment. Généralement il n'y a qu'un seul bit qui est activé à la fois. Chacun de ces bits a une signification particulière:
  1. **URG**: indique que le champ `` URGENT POINTER '' doit être exploité,
  2. **ACK**: indique que le champ `` ACNOWLEDGMENT NUMBER '' doit être exploité,
  3. **PSH**: indique au récepteur que toutes les données collectées doivent être transmises à l'application sans attendre les éventuelles données qui suivent,

4. **RST**: demande la réinitialisation de la connexion,
5. **SYN**: indique la synchronisation des numéros de séquence et que le champs `` SEQUENCE NUMBER '' contient la valeur de début de connexion.
6. **FIN**: indique que l'émetteur du segment a fini d'émettre.

- « **WINDOW** »: définit la longueur de la fenêtre coulissante. Le contenu de ce champs correspond donc au nombre d'octets à partir de la postion marquée dans accusé de réception que le récepteur est capable de recevoir. Le déstinataire ne donc pas envoyer les paquets qui viennent après le nombre x tel que x = numéro de séquence + window.
- « **CHECKSUM** »: sert à valider le paquet en vérifiant qu'il n'a pas été altéré lors de sa transmission. Ce checksum est obtenu en calculant le complément à 1 sur 16 bits de la somme des compléments à 1 des octets de l'en-tête et des données pris deux par deux (mots de 16 bits). Si le message entier contient un nombre impair d'octets, un 0 est ajouté à la fin du message pour terminer le calcul du Checksum.
- « **URGENT POINTER** »: Ce champs communique la position d'une donnée urgente en donnant son décalage par rapport au numéro de séquence. Ce champ n'est interprété que lorsque le Flag URG est marqué à 1. Dès que cet octet est reçu, la pile TCP doit envoyer les données à l'application.
- « **OPTIONS** »: C'est un ensemble de champs optionnels qui peuvent occuper un espace de taille variable à la fin de l'en-tête TCP. La présence d'options est détectée dès que le champs OFFSET est supérieur à 5.
- « **PADDING** »: Ce champs a un nombre de bits variable (entre 0 et 7), car il sert à combler le vide laissé par le champ "OPTIONS" afin d'obtenir un en-tête d'une taille multiple de 32 bits. La valeur des bits de bourrage est 0.
- « **DATA** »: Ce champ contient les données transportées. Durant la phase d'établissement de la connexion, ce champs a une longueur nulle.

I.1.1.5 Contrôle de la congestion et fiabilité de bout-en-bout

La congestion se produit quand la charge est plus importante que ce que l'on peut traiter ou gérer. Même si la couche réseau essaye de gérer la congestion, la majeure partie du travail est éffectué par le protocole TCP, car la solution convenable consiste à diminuer le débit des données émises. Pour cela TCP essaye de manipuler dynamiquement la taille de la fenêtre.

L'émetteur se rend compte d'une congestion quand plusieurs paquets restent sans aquittement (pas d'ACK alors que le timer de transmission est expiré). L'expiration du timer de transmission aujourd'hui est rarement due à une erreur de transmission, car les principaux supports de transmission sont en fibre optique.

La congestion peut se produire soit au niveau du récepteur, soit au niveau du réseau. Pour que l'émetteur évite cette congestion il doit donc gérer deux fenêtres: celle que le récepteur a accordé (relative à la taille de son tampon mémoire) et celle de la congestion. Le nombre d'octets que l'émetteur peut envoyer est donc le minimum entre ces deux fenêtres.

I.1.2 Protocole UDP

I.1.2.1 Protocole UDP dans l'architecture TCP/IP

Même si le protocole TCP gère bien les communications, il n'est pas adapté à certaines applications. Les applications temps-réel par exemple n'ont pas besoin des mécanismes de transport fiable de TCP. Elles peuvent même en souffrir. Pour ce genre d'applications, il est souvent préférable de gérer la perte et la congestion que d'essayer de les éviter. Ces applications choisissent alors à la place de TCP un autre protocole appelé UDP (*User Datagram Protocol*, RFC 768). Si champ "protocol" du datagrame IP est égal à 17, alors le protocole de la couche supérieure à laquelle il est transmis est UDP.

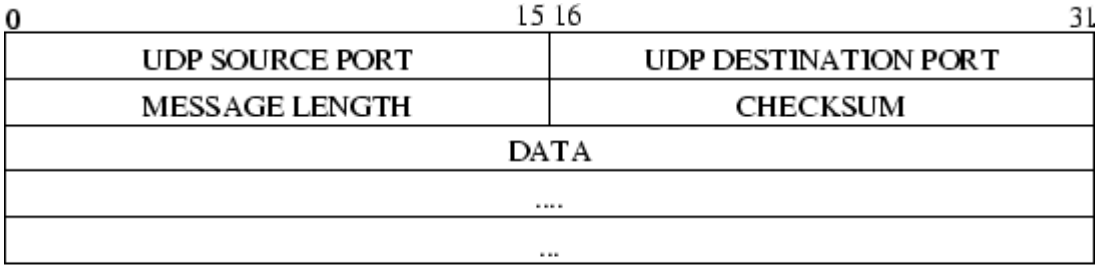
UDP est un protocole plus simple que TCP. Il n'est pas fiable, il travaille sans connexion, ne demande pas la retransmission des paquets perdus et ne remet pas dans l'ordre les paquets reçus.

UDP est adapté:

- dans le cas où les couches supérieures (application) s'occupent de la remise en ordre des messages,
- pour la transmission de la voix. La perte d'un paquet n'influence pas sur le message vocal transmis. De même, l'inversion de deux phonèmes ne gêne pas la compréhension du message vocal.
- pour la transmission de la vidéo. Pour les mêmes raisons que la voix, UDP est bien adapté à la transmission de la vidéo temps réel (TV, conférence...).

I.1.2.2 Structure d'un datagramme UDP

La structure du datagramme UDP est donnée par la figure ci-dessous:



Structure d'un datagramme UDP

Signification des champs du datagramme UDP:

- « **UDP SOURCE PORT** »: correspond au numéro de port de l'émetteur. C'est un champ optionnel utilisé dans le cas où le destinataire doit rendre une réponse à l'émetteur. S'il n'est pas utile sa valeur est mise à zéro (0).
- « **UDP DESTINATION PORT** »: correspond au numéro de port du destinataire.
- « **MESSAGE LENGTH** »: correspond à la longueur du datagramme, en-tête compris.
- « **CHECKSUM** »: c'est un champs optionnel, car UDP travaille sans reprise sur erreur,
- « **Data** »: contient les données transmises.

I.1.2.3 Influence d'IPv6 sur les protocoles de la couche transport:

Les modifications apportées aux protocoles de la couche transport (UDP et TCP) suite à

l'évolution du protocole IP (de la version 4 à la version 6) sont minimes. D'ailleurs l'un des pré-requis à la mise en oeuvre d'IPv6 était de laisser les deux protocoles TCP et UDP tels qu'ils sont, car ils sont utilisés par la majorité des applications réseaux. Le fait de ne pas modifier ces deux protocoles facilitera grandement le passage d'IPv4 à IPv6. Néanmoins certaines modifications se sont imposées, principalement le checksum qui a été adapté au format du paquet IPv6 et qui est devenu obligatoire pour UDP après avoir été facultatif. Le deuxième changement concerne la prise en compte de l'option jumbogramme de l'extension proche-en-proche.

I.3 Applications classiques

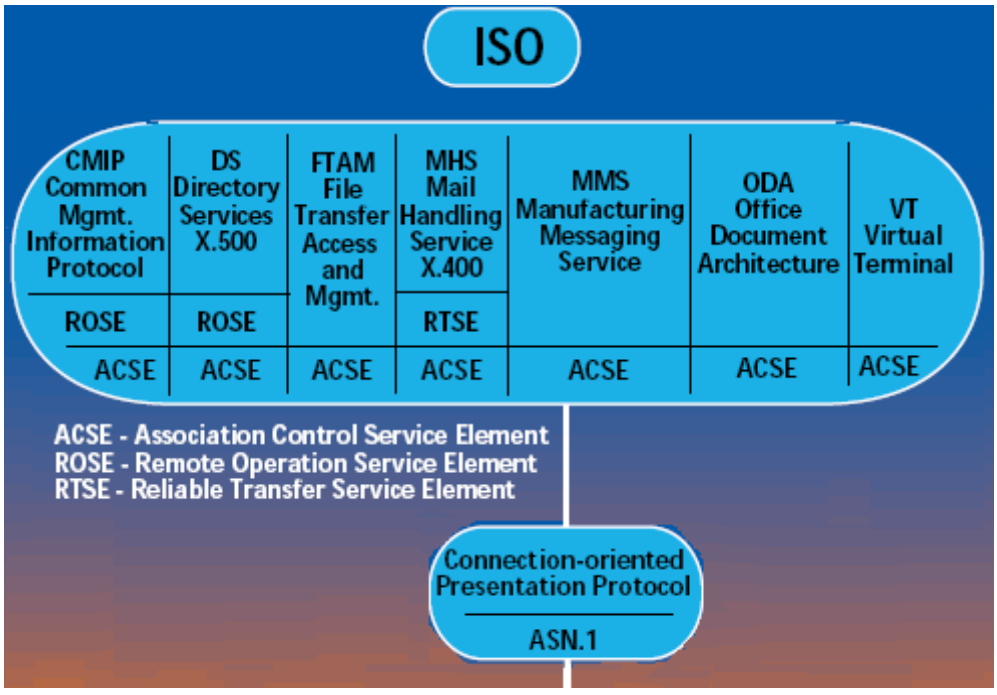
I.3.1 Couche application

La couche application est la couche la plus riche du modèle OSI et de l'architecture TCP/IP.

Elle fournit des services aux applications et aux utilisateurs qui sont situés juste au dessus d'elle.

I.3.2 Composantes application dans le modèle OSI

Parmi les services de la couche application du modèle OSI nous pouvons citer (voir poster des protocoles): ACSE, CCRSE, FTAM et MHS.



I.3.2.1 ACSE

L'ACSE (*Association Control Service Element*) permet d'établir des associations de niveau application avec authentification des accès. Elle permet de faire des appels entre deux

programmes applicatifs. ACSE vérifie l'identité et les contextes des entités applicatives (applications).

Nota: Au niveau application, les connexions sont appelées « associations ».

I.3.2.2 CCRSE

CCRSE (*Commitment, Concurrency and Recovery Service Element*) permet de coordonner de façon fiable (même en cas de panne de certains systèmes) des interactions entre des sites multiples, dont un qui est maître et les autres sont esclaves.

I.3.2.3 FTAM

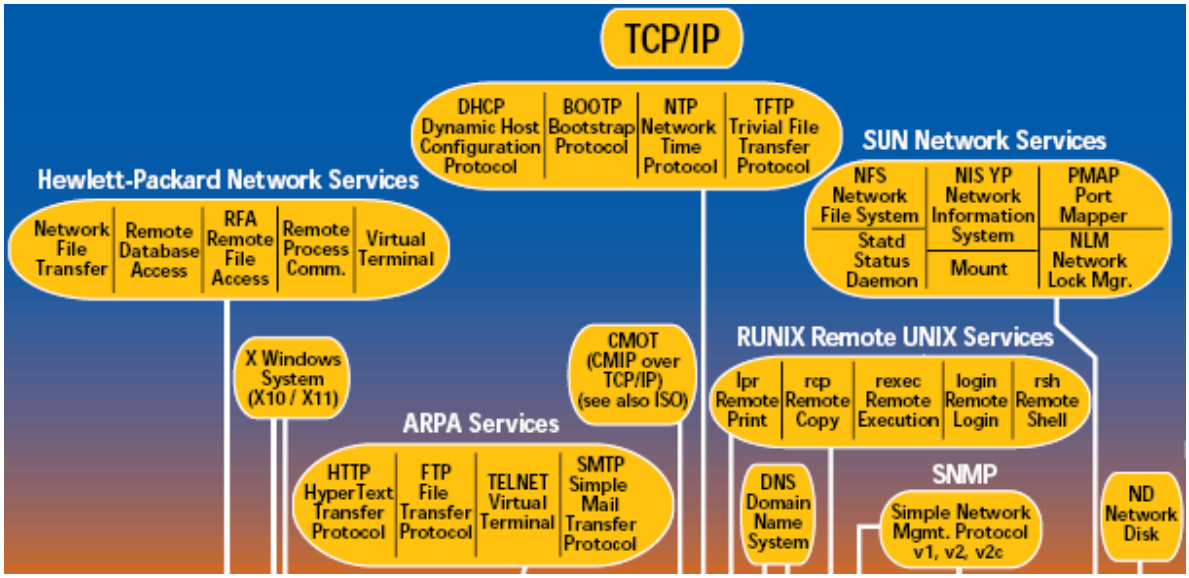
FTAM (*File Transfert, Access and Management*) est un protocole de transfert, d'accès et de gestion de fichiers distants.

I.3.2.4 MHS

Le service de messagerie MHS (*Message Handling System*) permet de transférer de façon fiable du courrier entre utilisateurs, sans nécessiter la présence simultannée des utilisateurs devant leur terminal.

I.3.3 Composantes application dans le modèle TCP/IP

La couche aplication de l'architecture TCP/IP posséde un grand nombre de protocoles (services de couche application). Ces services sont décrits dans des documents appelés RFC (Request For Comments), mais implémentés parfois différement par les différents constructeurs. Parmi ces services, on cite: TELNET, FTP, RPC, NFS, SMTP, FINGER, PING, SNMP, HTTP...



I.3.3.1 TELNET

TELNET (*TELecommunication NETwork*, RFC 854) est un service qui permet de se

connecter à une machine distante en émulation de terminal. Un terminal virtuel (NVT, *Network Virtuel Terminal*) est défini dans le but de résoudre les problème d'hétérogénéité des équipements utilisés; une traduction entre les caractéristiques locales et NVT est donc faîte à chacune des extrêmités. Le but du protocole Telnet est de fournir un moyen de communication très généraliste, bi-directionnel et orienté octet.

Par défaut, un client Telnet établit une connexion avec un serveur Telnetd, sur le port 23 (ceci est le fonctionnement par défaut, car le client Telnet peut par exemple se connecter au serveur httpd).

Lors d'une connexion au serveur, le client est authentifié à l'aide d'un login et d'un mot de passe. Lors de l'établissement de la connexion, une négociation d'options telnet (transmission binaire, écho...) a lieu entre le client et le serveur.

Commande : `telnet adresse-machine-distante`

Ex.

```
• ali@localhost:~$ telnet quantum-informatique.selfip.com
• Trying...
• Connected to quantum-informatique.selfip.com.
• Escape character is '^]'.
• login : ali
• Password:****
•
```

Exemple de client telnet : puTTY :

I.3.3.2 SSH

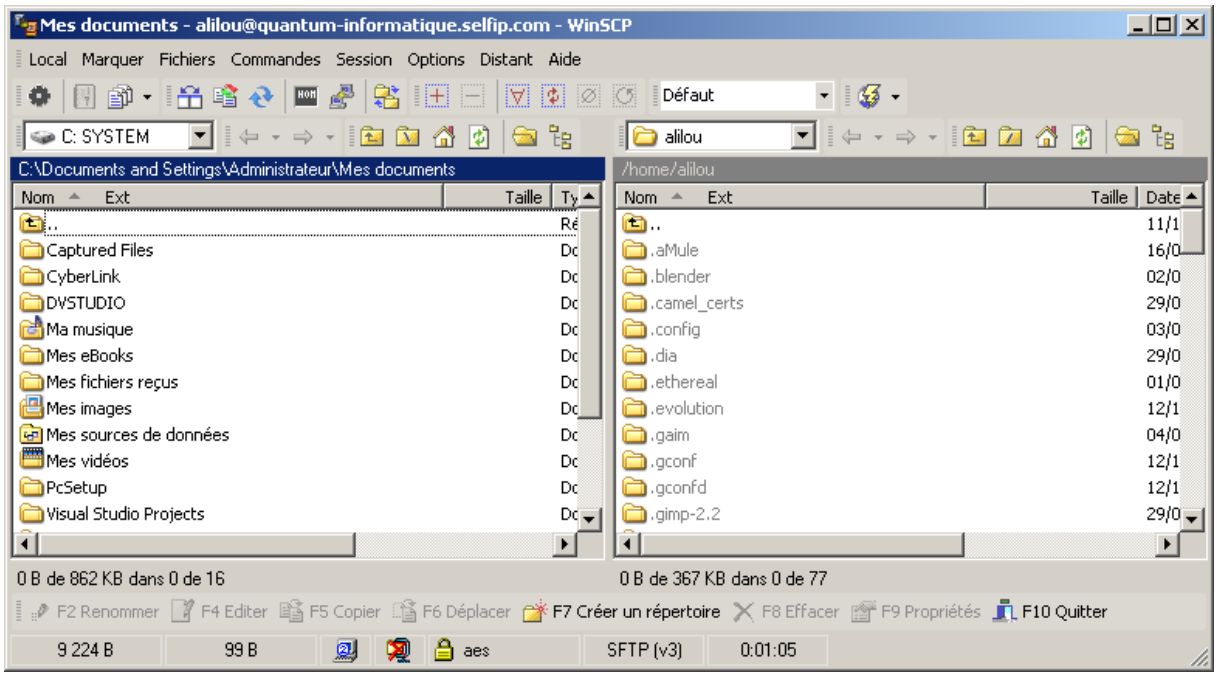
La simplicité de Telnet implique que toutes les communications sont transmises en clair sur le réseau, mots de passe compris. Des sniffers comme tcpdump ou Wireshark permettent d'intercepter les communications de la commande telnet. Des protocoles chiffrés comme SSH ont été développés pour fournir un accès distant remplaçant Telnet et dont l'interception ne fournit aucune donnée utilisable à un éventuel espion.

SSH (Secure Shell) est donc un protocole de communication sécurisé, qui impose un échange de clés de chiffrement en début de connexion. Par la suite toutes les trames sont chiffrées, ce qui rend impossible d'utiliser un sniffer pour voir ce que fait l'utilisateur. Le protocole SSH a été conçu avec l'objectif de remplacer les différents programmes rlogin, telnet et rsh.

Ex:

```
• ssh quantum-informatique.selfip.com
```

Exemple de client SSH sous Windows: WinSCP:



Sous Linux: Il suffit d'utiliser par exemple konqueror avec le protocole fish. En tapant l'adresse de la manière suivante: `fish://adresse-machine-distante`

I.3.3.3 FTP

FTP (*File Transfer Protocol*, RFC 959) permet de transférer et/ou de gérer des fichiers entre des machines distantes.

Le client FTP établit une connexion sur le port 21 du serveur ftpd après avoir été authentifié (une connexion sans authentification est possible). Cette connexion permet d'échanger des commandes et réponses concernant les transferts. Pour transférer les données, une deuxième connexion sur le port 20 (TCP) du serveur est établie. Cette deuxième connexion est contrôlée par des commandes qui transitent sur la première connexion.

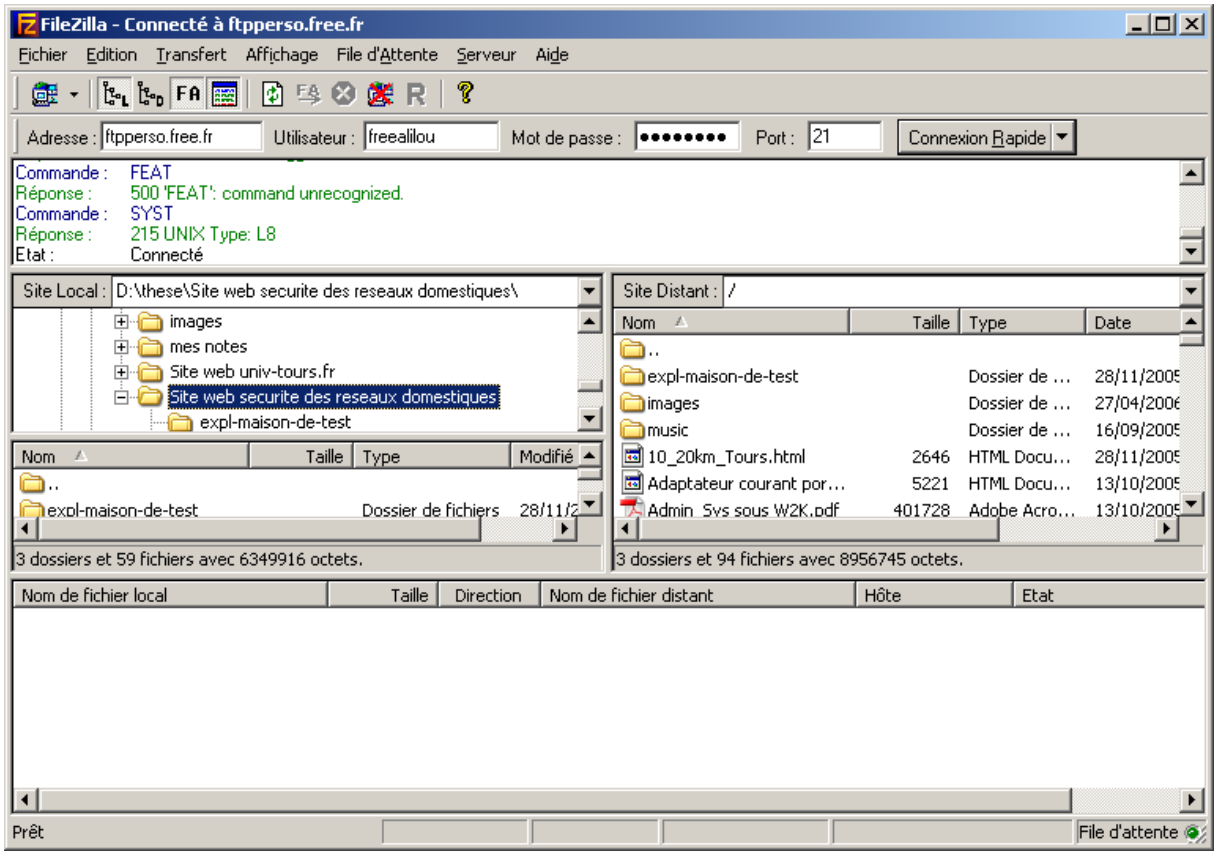
Exemple:

```
C:\Documents and Settings\Administrateur>  ftp quantum-informatique.selfip.com
Connecté à Quantum-informatique.selfip.com.
220 ******* Azul: Welcome to Ali's FTP Server *******
Utilisateur (Quantum-informatique.selfip.com:(none)) : ali
331 Password required for ali.
Mot de passe : *******
230 ali, nous vous souhaitons la bienvenue

ftp> ls
200 PORT command successful
150 Opening ASCII mode data connection for file list
Doc
Depot
.....
ISIS
226 Transfer complete.
ftp : 58 octets reçus en 0,00 secondes à 58000,00 Ko/sec.

ftp> !ls
'ls' n'est pas reconnu en tant que commande interne ou externe, un programme
```

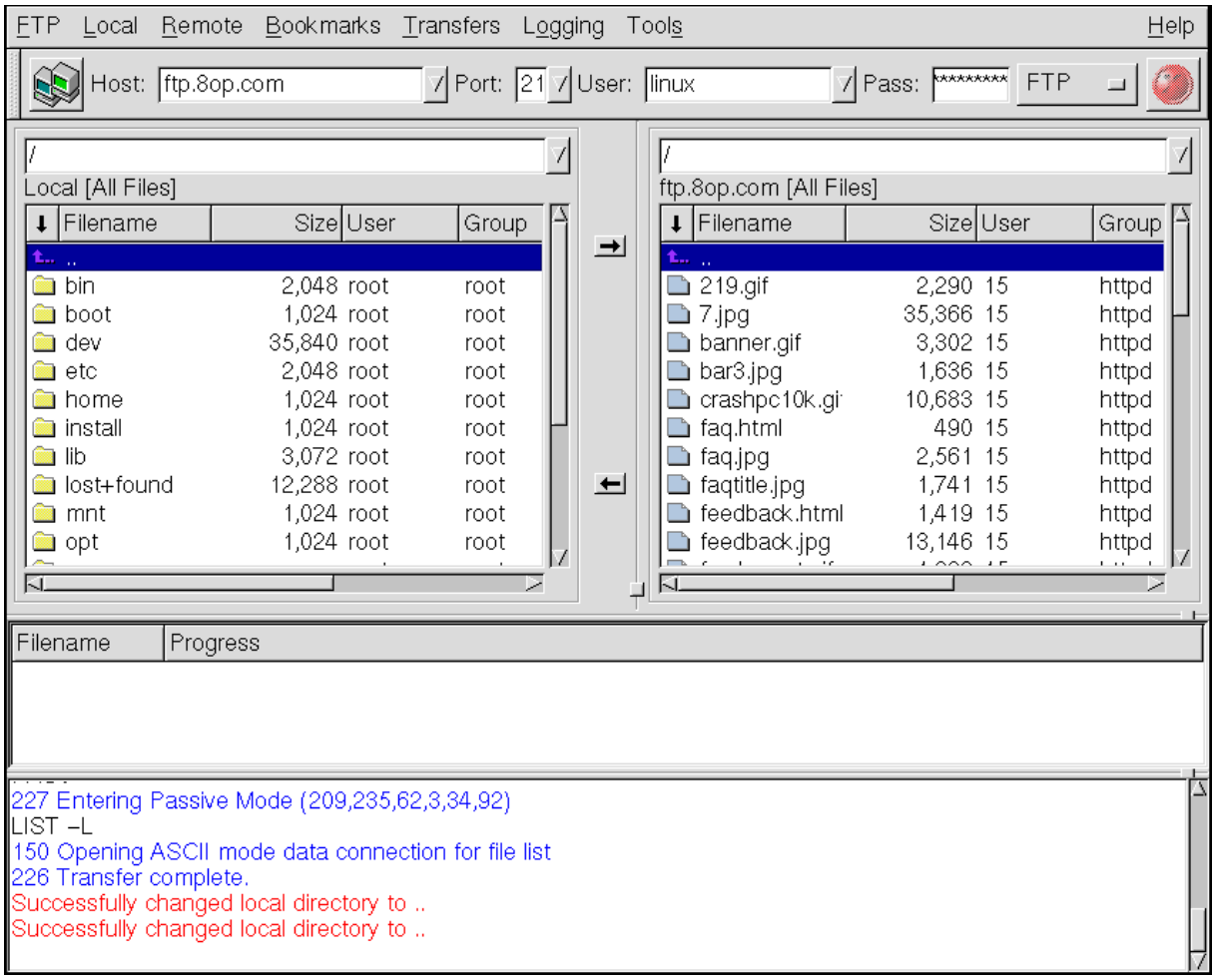
```
  exécutable ou un fichier de commandes.  <-- car client sous Windows

• ftp> !dir
•  Le volume dans le lecteur C s'appelle SYSTEM,  Le numéro de série du volume est
  94C4-CE53
•  Répertoire de C:\Documents and Settings\Administrateur
• 25/10/2006  07:26    <REP>          .
• 25/10/2006  07:26    <REP>          ..
• ……….  ……..  ……..
• 21/03/2004  00:53           176 604 ~
•               9 fichier(s)          390 771 octets
•              14 Rép(s)   3 162 386 432 octets libres
•
• ftp> quit
```

Exemple de version graphique d'un client FTP:

Sous Windows: Filezilla



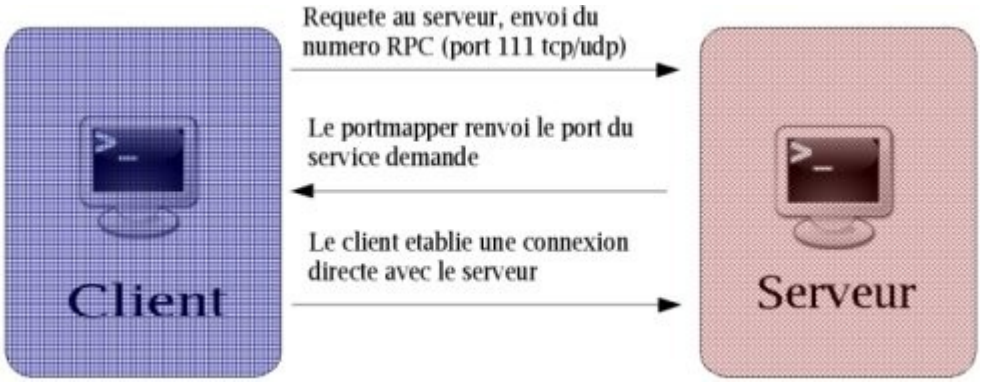
Sous Linux: GFTP



I.3.3.4 RPC

Le service RPC (*Remote Procedure Call*, RFC 1057) permet d'effectuer des appels de procédures qui s'exécutent sur des machines distantes (principe du client-serveur). Le comportement des appels distants est similaire à celui des appels locaux. Un appel RPC contient l'identificateur de la procédure distante à exécuter (adresse de la machine distante, numéro de la procédure distante et son numéro de version) et la réponse contient le résultat de l'appel. RPC peut être implémenté au dessus de TCP, mais il est implémenté le plus souvent au-dessus de UDP. Dans le modèle OSI RPC se situe au niveau de la couche 5 (session).

Lorsqu'un client veut se connecte à un serveur RPC, il lui envoie le numero de service auquel il veut accéder et le serveur lui renvoi le port associé, car le rôle du serveur est aussi de faire la correspondance entre le numero de service RPC et le port correspondant (la correspondance entre le numéro et le service est sauvegardée dans /etc/rpc). Le client n'a donc pas besoin de connaitre les details réseau. Les serveurs RPC (rpcbind et portmap) écoutent par défaut sur le port 111 tcp/udp.



Exemple de commande:

```
$ rpcinfo -p hostname
```

Cette commande (rpcinfo) permet de dialoger avec un serveur RPC (visualisation des tables...).

NFS (*Network File System*), lui aussi, utilise la méthode de connexion RPC.

I.3.3.5 NFS

NFS (Network File System) est un protocole qui permet de partager des fichiers entre les machines d'un réseau local, et de les utiliser comme s'ils se trouvaient sur le disque dur de la machine locale. Le transfert de fichier devient ainsi inutile.

Grâce à NFS, on peut accéder à certains répertoires sur le serveur tels que son répertoire d'accueil (/home) qui se trouve sur le serveur. Ce qui permet aux utilisateurs de retrouver leurs données sur toutes les machines sur lesquelles ils se loguent sans devoir se reloguer au serveur.
Un système de fichier virtuel est implémenté afin de rendre transparent l'accès à tous les fichiers. Pour pouvoir utiliser une ressource externe, on doit la monter sur un point de montage comme on monte un périphérique ou une partition locale. Le montage peut se faire à l'aide de la commande `mount`. Mais le mieux est de le faire au démarrage (pour les ressources NFS les plus utilisées). Pour ceci, il suffit d'ajouter les lignes qu'il faut aux fichiers `/etc/fstab` (pour le client) et `/etc/exports` (pour le serveur).

Exemple de fichier `/etc/fstab` (sur le client):

```
• /dev/hda1  / ext2       defaults                              1 1
• /dev/hda10       /root  ext2         defaults                 1 2
• /dev/hda9  /swap  swap         defaults                  0 0
• /dev/cdrom       /mnt/cdrom   iso9660      ro,user,noauto            0 0
• /dev/fd0   /mnt/floppy auto          rw,user,noauto            0 0
• /dev/sda1  /mnt/cleUSB auto          noauto,rw,users,dev,async 0 0
• none       /dev/pts    devpts        gid=5,mode=620            0 0
• /serverpc:/home/users/Ecole    /home/users/Ecole   nfs    defaults           0 0
• /serverpc:/var/spool/mail      /var/spool/mail           nfs    default      0 0
• ... ...
•
```

Exemple de fichier `/etc/exports` (sur le serveur):

```
• /mnt/exportedFfile    IUT-univ-1.fr
• /                     machine1(rw) machine2(rw,no_root_squash)
• /home/user1           pc001(rw,all_squash,anonuid=150,anongid=100)
```

```
• /public               (ro,insecure,all_squash)
• /home/tmp             *(rw,root_squash)
```

I.3.3.6 SMTP

SMTP (*Simple Mail Transfer Protocol*, RFC 821) est un service (protocole) de communication utilisé pour transférer le courrier électronique vers les serveurs de messagerie électronique. Il fonctionne en mode connecté (connexion point à point, encapsulée dans une trame TCP/IP). Puisque le courrier électronique est l'un des services les plus utilisés sur Internet, la suite TCP/IP offre une panoplie de protocoles permettant de gérer facilement le routage du courrier sur le réseau. Le logiciel sendmail est l'un des premiers à utiliser SMTP. Aujourd'hui la plupart des logiciels clients l'utilise (Postfix, Qmail, Exchange de Microsoft...).

Avant ce protocole ne permettait pas l'envoi de n'importe quel message (fichiers binaires), car il utilisait du texte en ASCII (7 bits). Pour pallier ce problème, des standards comme MIME ont été développés pour permettre le codage des fichiers binaires au travers de SMTP.

SMTP ne permet pas la récupération du courrier électronique. C'est les deux protocoles **POP** et **IMAP** qui se chargent de cette tâche.

Fonctionnement de SMTP:

Le fonctionnement du protocole SMTP et assez simple:

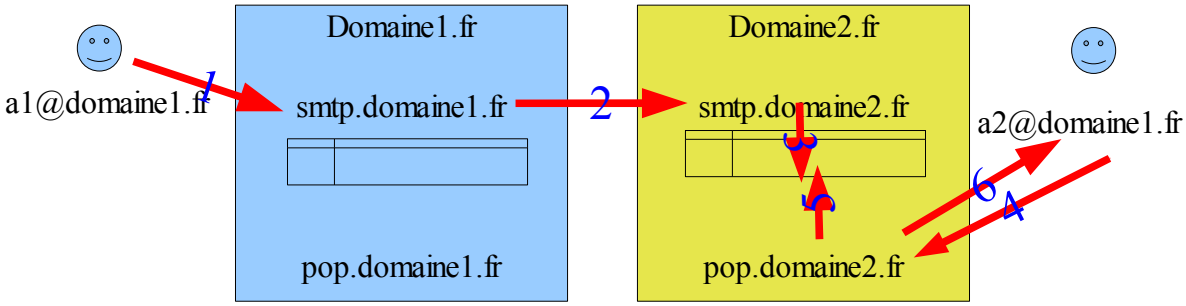
1. Quand l'utilisateur a fini d'écrire son mail et il décide de l'envoyer, le client fait une demande d'ouverture de seesion SMTP avec le serveur qui a été spécifié (serveur SMTP par défaut).
2. Lors de l'ouverture de la session SMTP, le client envoie la commande `HELLO` (ou `EHELLO`) au serveur suivie d'un espace (`SP`) et du nom de domaine de la machine cliente. Tout cela est validé par `CRLF`.
3. Le client envoie ensuite la commande `MAIL FROM`: suivie de l'adresse email de l'expéditeur.
4. Si la commande est acceptée, le serveur renvoie le message `250 OK`.
5. Le client envoie la commande `RCPT TO`: suivie de l'adresse mail du destinataire.
6. Si la commande est acceptée (le destinataire existe soit chez lui soit chez un autre serveur smtp distant), le serveur renvoie le message `250 OK`. Si le serveur SMTP constate que le destinataire n'est pas dans son domaine, il cherche alors le serveur SMTP correspondant et le contacte.
7. Le client envoie la commande `DATA`,
8. Si la commande est acceptée, le serveur renvoie le message `354` pour dire que l'envoi du corps du texte peu commencer. Le serveur accepte toutes les lignes reçues jusqu'à la ligne qui ne contient que un point qui indique la fin du corps du message. Le corps du message contient d'autres informations que le texte envoyé. Il contient par exemple les champs suivants: Date, Subject, Cc, Bcc, From.
9. Si tout est bien passé, le serveur renvoie le message `250 OK` puis la connexion sera close.

Ceci est un exemple de ce qui se passe:

```
• 220 smtp.xxxx.xxxx SMTP Ready
• HELO client
• 250 Hello client, pleased to meet you
• MAIL FROM:<user@xxxx.xxxx>
• 250 <user@xxxx.xxxx> ... Sender ok
• RCPT TO:<user2@yyyy.yyyy>
• 250 recipient ok.
• DATA
```

```
• 354 Enter mail, end with "." on a line by itself
• Cette phrase représente les données à envoyer. Les données doivent
  finir par une ligne qui ne contient qu'un seul point. Elle est juste
  ci-dessous.
• .
• 250 Ok?
• QUIT
• 221 Closing connection
• Connection closed by foreign host.
•
```

Une fois le serveur a reçu le message il le met dans sa boîte aux lettres. Ce message restera dans cette boîte aussi longtemps qu'il le faut jusqu'au moment ou le destinataire s'y connecte en envoyant une requête POP ou IMAP. Le serveur POP (ou IMAP) consulte la boîte et constate qu'il y a un nouveau message. Il le lit et lefface de la boîte (comportement par défaut. On peut lui demander de le laisser dans la boîte sans l'effacer).



Envoi d'un mail sous Unix en ligne de commande à un utilisateur:

```
• ali@localhost:~$ mail Séb
• Subject: Hello
• ici le texte que je veux t'envoyer, fini par un Ctrl D
• Cc: Ali2
• ali@localhost:~$
```

I.3.3.7 FINGER

Finger (RFC 1196) est un protocoles très simple. Il permet d'obtenir des informations sur les utilisateurs d'un système.

Syntaxe: finger machine-cible ou finger nom-utilisateur

Exemple: L'exemple suivant permet d'avoir des informations concernant tous les utilisateurs connectés à la machine machine-cible (utilisateur toto et root) et depuis quand (colonne Time: 20:43 et 16:20). Le caractère @ est équivalent au caractère * utilisé pour un listing de répertoire.

```
• [root@localhost /root]#finger @machine-cible
• Login Name Tty Idle Login Time Office toto Le toto pts/7 3d Mar 26 20
  :43
• (case)// root root pts/4 5d May 25 16 :20
```

La commande (service) finger n'est pas dangeureuse en soit même, mais il faut désactiver le démon qui lui répond si on en a pas besoin, car cette commande donne trop d'informations sur les utilisateurs systèmes. Pour désactiver ce service, il faut (sous UNIX) commenter dans le fichier `/etc/inetd.conf` la commande qui correspond à finger:

```
# finger stream tcp nowait root /usr/sbin/tcpd in.fingerd
```

I.3.3.8 PING

Ping (*Packet INternet Groper*) est un protocole simple qui emploi les messages ICMP `echo` et `echo-reply` afin de s'assurer du bon fonctionnement d'une ressource sur le réseau (bon fonctionnement des couches sur la machine locale, le bon fonctionnement d'une machine distante ou d'un routeur, l'adresse IP d'une machine distante et l'accès au réseau) et parfois d'avoir des statistiques sur des connexions.

Syntaxe: ping adresse-cible

Exemple:

```
C:\Documents and Settings\Administrateur>ping -n 2 quantum-
informatique.selfip.com

Envoi d'une requête 'ping' sur quantum-informatique.selfip.com [82.245.222.65]
avec 32 octets de données :

Réponse de 82.245.222.65 : octets=32 temps<1ms TTL=64
Réponse de 82.245.222.65 : octets=32 temps<1ms TTL=64

Statistiques Ping pour 82.245.222.65:
    Paquets : envoyés = 2, reçus = 2, perdus = 0 (perte 0%),
Durée approximative des boucles en millisecondes :
    Minimum = 0ms, Maximum = 0ms, Moyenne = 0ms

C:\Documents and Settings\Administrateur>ping 127.0.0.1

Envoi d'une requête 'ping' sur 127.0.0.1 avec 32 octets de données :

Réponse de 127.0.0.1 : octets=32 temps<1ms TTL=128
Réponse de 127.0.0.1 : octets=32 temps<1ms TTL=128
Réponse de 127.0.0.1 : octets=32 temps<1ms TTL=128
Réponse de 127.0.0.1 : octets=32 temps<1ms TTL=128

Statistiques Ping pour 127.0.0.1:
    Paquets : envoyés = 4, reçus = 4, perdus = 0 (perte 0%),
Durée approximative des boucles en millisecondes :
    Minimum = 0ms, Maximum = 0ms, Moyenne = 0ms

C:\Documents and Settings\Administrateur>
```

Remarque: La commande à utiliser pour IPv6 est: ping6.

I.3.3.9 SNMP

SNMP (*Simple Network Management Protocol*, RFC 1155, 1157, 1213) est un protocole qui permet aux administrateurs réseau de gérer leurs équipements et de diagnostiquer les problèmes de leurs réseaux (réseaux IP). Il est utilisé aussi pour la gestion à distance des applications (bases de données, serveurs...). Il est relativement simple par rapport à ce qu'il permet de faire.

Il est actuellement le protocole le plus utilisé pour la gestion des équipements de réseaux. Un autre protocole plus complexe pour la gestion des réseaux existe. Il s'agit du protocole CMOT (***C****ommon* ***M****anagement information services and protocol* ***O****ver* ***T****CP/IP*, RFC 1095).

Principe de fonctionnement:

SNMP est basé sur deux éléments principaux: la station de supervision (ou superviseur) et les agents:

1. **Le superviseur** (appelé aussi manager) est une station (poste de travail) qui permet à l'administrateur réseau d'exécuter des requêtes de management pour contrôler les éléments réseaux.
2. **Les agents** (appelés aussi éléments actifs) sont les équipements (station de travail, concentrateur, routeur, pont...) ou les logiciels que l'on cherche à gérer. Chaque équipement du réseau dispose d'une entité (module résident) appellée agent qui a pour but de récolter des informations et de répondre aux requêtes du superviseur.

Les équipements réseaux (switchs, hubs, routeurs, serveurs...) contiennent donc des objets manageables (informations matérielles, paramètres de configuration, statistiques de performance...). Ces objets sont classés dans une base de données appelée **MIB** (*Management Information Base*). Les dialogues de SNMP effectués entre le manager et les agents ont pour but de receuillir des objets sur la MIB.

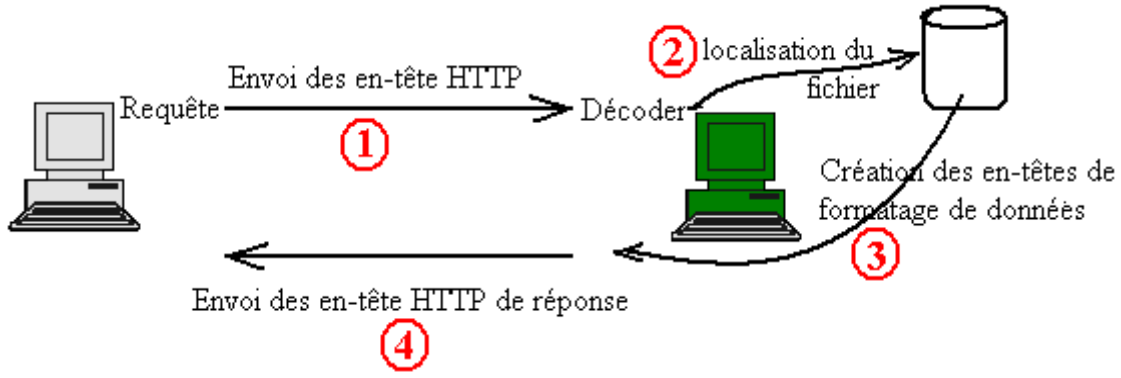
Le fonctionnement de SNMP est constitué d'un ensemble de requêtes, de réponses et d'un nombre limité d'alertes. Le manager envoie des requêtes à l'agent qui lui renvoie à son tour des réponses. Si un événement anormal s'est produit sur l'élément réseau auquel est associé cet agent, ce dernier renvoie une alerte (**trap**) au manager.

I.3.3.10 HTTP...

HTTP (*HyperText Transfer Protocol*, RFC 1945) est le protocole le plus utilisé sur Internet (à partir de 1990). Il est léger et rapide. C'est un protocole de communication client-serveur développé pour le World Wide Web (WWW, en 1990). Il sert à échanger entre un client (navigateur) et un serveur (httpd par exemple) toute sorte de données, mais principalement des pages Web écrites en langage HTML. Ces données sont localisées grâce à une chaîne de caractères appelée URL (*Uniform Ressource Locator*). La version 0.9 de HTTP était destinée uniquement à transférer des données sur Internet, alors que sa version actuelle (1.0) permet aussi de transférer des messages avec des en-têtes décrivant le contenu du message en utilisant un codage du type MIME (*Multipurpose Internet Mail Extensions*, standard initialement destiné à étendre les possibilités de SMTP en y insérant des documents).

Fonctionnement:
La communication entre le client (souvent c'est un navigateur) et le serveur se fait en deux temps: Le client effectue une requête HTTP (une ligne ASCII se terminant par "CR LF" (carriage return, line feed), puis le serveur traite cette requête et envoie une réponse HTTP.

Si le numéro de port n'est pas mentionné dans l'adresse de la requête envoyée par le client, c'est le numéro 80 qui lui sera affecté par défaut.



Cette requête, dans le cas le plus simple, est composée de la méthode GET (la méthode la plus courante), suivie de l'adresse du document et du numéro de port. Après avoir eu la réponse du serveur il faut taper la commande GET/ (voir listing ci-dessous) pour avoir le contenu de la page sollicitée.

```
telnet quantum-informatique.selfip.com 80
GET/


<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
  <title>Quantum-Informatique</title>
</head>
<body style="background-image: url(images/back.JPG);">
<div style="font-weight: bold; text-align: center;"><big><big
style="color: rgb(255, 0, 0);"><big>Q u a n t u m - I n f o r m a t i
q u e</big></big></big></div>

.......

</body>
</html>


Perte de la connexion à l'hôte.
C:\Documents and Settings\Administrateur>
```

Exemple de logiciels:

- Clients: les navigateurs Web, les robots d'indexation et les aspirateurs de site.
- Serveurs: Apache HTTP Server, Internet Information Services (IIS) et le serveur Web Zeus .

Remarques:

1. La liaison entre le client et le serveur n'est pas toujours directe. Il pourrait y avoir entre eux des machines intermediaires qui servent de relais. Ces machines s'appellent « Serveurs mandataires » (Proxy).
2. Une variante sécurisée du protocole HTTP existe. Elle s'appelle HTTPS (HTTP

Sécurisé). Elle est sécurisée avec SSL (ou TLS). Le port par défaut de HTTP est 80 et celui de HTTPS est 443.

III Projets

I « Interface de configuration graphique d'un firewall »

Réalisation d'une interface facilitant la configuration d'un firewall dédié à la sécurisation des communications d'un réseau domestique dédié à la santé.

--> Présenter aux utilisateurs une interface (une page web par exemple) qui contient un graphique représentant la structure du réseau. Il faut pouvoir introduire des informations sur ce graphique et/ou ailleurs (d'autres champs, questions/réponses). A la fin, un simple clic sur le bouton « Configurer firewall » doit créer les règles nécessaires (script) à la configuration du firewall.

II « Interface de communication en temps réel dédiée à la santé »

Réalisation d'une petite interface de communication en temps réel (tchat) dédiée à l'environnement de la santé.

Cette application doit permettre aux visiteurs de choisir un salon ou sujet de discussion.

III « Application de gestion d'accès à Internet »

Réalisation d'une application pour un fournisseur d'accès à Internet (un cyber-café). Cette application doit répondre aux besoins suivants (cahier des charges):

- Autoriser/interdire l'accès à l'extérieur (Internet) à un poste bien défini du réseau local.

- Autoriser l'accès à Internet à un poste soit pour une durée déterminée soit pour une durée indéterminée.

- Afficher au niveau du serveur et du client la durée de connexion ainsi que la somme à payer. Les tarifs diffèrent selon l'heure de connexion. Il y a un tarif de jour et un tarif de nuit.

- Dans le cas où il y a un problème de connexion, il faut pouvoir arrêter momentanement (Pause) le compteur, puis le relancer une fois le problème est résolu.

- ...

IV « Application FTP » (projet proposé par A.Aoun)

- RFC 959
- Port réservé pour le serveur = 21
- **File Transfer Protocol Overview**

  There are four essential elements to the file transfer protocol:

  - **Control connection:** An FTP client establishes a control connection to an FTP server, and uses it to send and receive all FTP commands and replies.
  - **Data connection:** Some FTP commands cause the client and server to establish a separate data connection to transfer bulk data. They close the connection when they complete the transfer.
  - **FTP commands:** An FTP client sends commands to an FTP server on the control connection. Each command begins with a three- or four-letter code. Some commands also send arguments.
  - **FTP replies:** An FTP server responds to the initial control connection from the client and to all subsequent FTP commands. Each reply begins with a three-digit code that indicates the status of the request. Commands have subsets of possible reply codes, and many of them vary.

- Trois étapes :
  * Etape 1 (9 Mai):
  Connexion/Déconnexion et Commandes USER – PASS
  * Etape 2 (30 Mai):
  Commandes PWD – CWD – LIST – DELE
  * Etape 3 (20 Juin):
  Commandes RETR – STOR

  D'autres commandes peuvent être utilisées telles que TYPE – PASV…

⟶ L'application doit être écrite avec le client sous Windows et le serveur sous Unix. Le langage de programmation est le langage C. Sous Windows, les fonctions de communication seront réalisées sous forme d'une DLL et l'interface graphique en Visual Basic.

Lien utiles:

- Guy Pujolle, Les réseaux, Edition Eyrolles, ISBN: 2-212-09119-2.
- Douglas Comer, TCP/IP: Architectures, protocoles, applications, Edition Dunod, ISBN: 2-10-008181-0.
- Olaf Kirch & Terry Dawson, Administration réseau sous Linux, Edition Oreilly, ISBN: 2-84177-125-3.
- Chauvin Hameau, Wi-fi - maîtriser le réseau sans fil, Edition:ENI, ISBN : 2746020548
- Jean-François Susbielle, Internet, multimédia et temps réel, Edition: Eyrolles, ISBN : 2212091184
- Guy Cazuguel & Bassel Solaiman & Collectif, Santé et technologies de l'information : Annales des télécommunications Tome 58 N° 5-6 Mai-juin 2003, Edition: Hermes Science Publications, ISBN : 2746207753
- Jean-François Bouchaudy, TCP/IP sous Linux : Administrer réseaux et serveurs Internet/Intranet sous Linux, Edition: Eyrolles, ISBN : 2212113692

Liens Internet:

- Le modèle TCP/IP: http://www.frameip.com/tcpip/
- C'est quoi TCP/IP (pour débutants): http://sebsauvage.net/comprendre/tcpip/
- La suite de protocoles TCP/IP: http://www.commentcamarche.net/internet/tcpip.php3
- Wikipedia: Suite des protocoles Internet: http://fr.wikipedia.org/wiki/Suite_des_protocoles_internet
- Wikipedia: Le modèle OSI: http://fr.wikipedia.org/wiki/Mod%C3%A8le_OSI
- Pour savoir où se situe géographiquement une adresse IP: http://www.ip-adress.com/
- Adressage et routage: http://www.httr.ups-tlse.fr/pedagogie/cours/internet/adresse/textes/adresse.htm
- Les réseaux (par Gonzalez, univ-lille3): http://www.grappa.univ-lille3.fr/polys/reseaux-DG/
- La page des réseaux (La théorie des réseaux locaux et étendus): http://hautrive.free.fr/reseaux/page-reseaux.html
-

La suite de ce cours (ie. partie « Systèmes de transmission et accès au réseau ») est commentée car mal rédigée (que des notes). Veuillez voir les diapositives du cours.

II.Systèmes de transmission et accès au réseau

Pour transmettre des informations sur un réseau, plusieurs supports de transmission existent. Ces supports peuvent être filaires ou sans fil.

II.1 Systèmes filaires et sans fil

----> Voir mes diapos