Chapitre 7. Le modèle OSI et la pile de protocoles TCP/IP

1 Objectifs de ce chapitre

À l'issue de ce chapitre, vous serez capable :

- De décrire le modèle conceptuel OSI, d'en donner les 7 couches dans l'ordre et de décrire brièvement leurs fonctionnalités.
- De décrire le mécanisme d'encapsulation et de donner pour les couches basses (couche 1 à 4) les principales informations qui sont encapsulées dans l'en-tête.
- De décrire la pile de protocoles TCP/IP, d'en donner les 4 couches dans l'ordre et de citer les principaux protocoles intervenants au niveau des ces différentes couches.
- De comparer le modèle OSI avec la pile de protocoles TCP/IP.
- D'expliquer le fonctionnement des principaux protocoles de la pile TCP/IP.

2 Introduction

Au cours des deux dernières décennies, le nombre et la taille des réseaux ont augmenté considérablement. Cependant, bon nombre de réseaux ont été mis sur pied à l'aide de plates-formes matérielles et logicielles différentes. Il en a résulté une **incompatibilité** entre de nombreux réseaux et il était devenu difficile d'établir des communications entre des réseaux fondés sur des spécifications différentes. Pour résoudre ce problème, **l'Organisation Internationale de Normalisation** (ISO/International Organization for Standardization) a examiné de nombreuses structures de réseau. L'ISO a reconnu l'opportunité de créer un **MODELE réseau qui aiderait les concepteurs à mettre en œuvre des réseaux capables de communiquer entre eux et de fonctionner de concert (interopérabilité)**. Elle a donc publié le **modèle de référence OSI** (Open System Interconnection) en 1984.

Dans la suite de ce cours, nous parlerons régulièrement **d'ordinateur source**, appelé aussi hôte source ou ordinateur **émetteur**, qui est l'ordinateur **d'où proviennent les messages** (données). **L'hôte de destination** ou **ordinateur récepteur** est celui auquel les **messages sont destinés.**

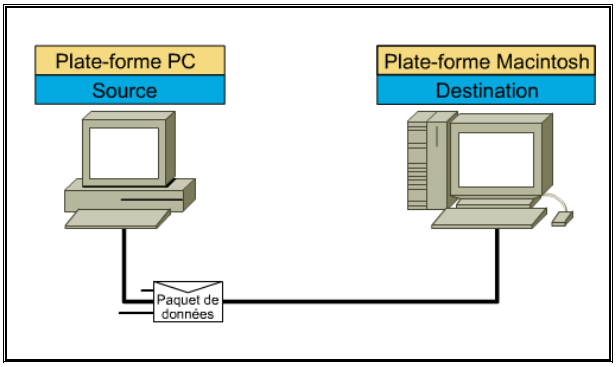


Fig. 1. Communication Réseau.

3 Les protocoles

Pour que des paquets de données puissent se rendre d'un ordinateur source à un ordinateur de destination sur un réseau, il est important que toutes les unités du réseau communiquent dans la **même langue ou protocole**. Un protocole consiste en un **ensemble de règles** qui augmentent l'efficacité des communications au sein d'un réseau.

De manière plus technique, *les protocoles sont les* ***règles qui régissent :***

- le contenu,
- le format,
- la synchronisation,
- la mise en séquence
- le contrôle d'erreurs des messages échangés entre les éléments du réseau.

Lorsqu'on parle de protocole, on se réfère généralement à un ensemble de protocoles nommé **pile de protocoles qui couvre l'ensemble du modèle OSI.** Ce concept correspond à une architecture très élaborée.

Fig. 2. Au sujet des protocoles

Système de couches

La fonction d'un réseau est de permettre l'échange d'informations entre ordinateurs. Pour ce faire, un certain nombre de **tâches** doivent être accomplies dans un **ordre chronologique** avant que les données ne puissent être envoyées sur le support de transmission.

Ces étapes constituent les **couches** de la pile de protocole utilisé. Chacune ajoute l'information nécessaire à la couche correspondante sur l'ordinateur récepteur. La **couche n** d'un ordinateur communique avec la **couche n** d'un autre ordinateur. Les règles et conventions utilisées lors de cette communication sont collectivement appelées protocole de couche n.

Chaque couche connaît des informations à son propre niveau.

- la couche Y sait comment indiquer à la carte réseau d'envoyer des données mais elle ne connaît rien du serveur ni de la machine dont les données sont redirigées.

- la couche X sait comment communiquer avec le serveur mais elle ne sait rien de la manière dont on envoie un message sur la carte réseau.

Couche X	je connais le serveur	je ne sais pas envoyer des données sur la carte
Couche Y	je sais envoyer des données sur la carte	je ne connais pas le serveur

Il faut évidemment que les 2 machines possèdent le même protocole pour que les informations ajoutées d'un côté puissent être interprétées de l'autre. Combinées les unes aux autres, ces couches permettent d'obtenir un réseau local.

Cette conception en couches donne aux concepteurs une plus grande souplesse d'adaptation. Si l'on **scinde** de cette façon les fonctionnalités, c'est parce que les différents composants matériels et logiciels sont réalisés par **différents concepteurs**.

Lorsque l'on doit s'adapter à un matériel spécifique, **seule une partie de la pile** de protocole sera modifiée. Deux groupes ont permis de définir des normes réseau dans un effort de standardisation des technologies réseau:

- ISO: Organisation Internationale de normalisation : a défini le modèle OSI
- IEEE: Institut des Ingénieurs électriciens et électroniciens : l'a défini plus en profondeur

4 Pile de protocole du modèle OSI

4.1 Historique

Au début, le développement des **LAN** (réseaux locaux), des **MAN** (réseaux métropolitains) et des **WAN** (réseaux étendus) a été plutôt chaotique à bien des égards.

Le début des **années 1980 a été marqué par une croissance exceptionnelle du nombre et de la taille des réseaux**. Les entreprises ont rapidement pris conscience des **économies** qu'elles pouvaient réaliser et des **gains de productivité** associés à la technologie des réseaux. Elles en ont donc ajouté de nouveau et développé ceux qui existaient aussi rapidement que le permettait le progrès des nouvelles technologies et des produits de gestion de réseau.

Vers le **milieu des années 1980**, les entreprises ont commencé à faire face à des problèmes consécutifs à cette extension effrénée. Il devenait en effet de plus en plus difficile pour les réseaux utilisant des implémentations et des spécifications différentes de **communiquer entre eux.** Les entreprises se sont alors rendu compte qu'elles devaient s'éloigner des systèmes de réseau *propriétaires*.

Pour résoudre le problème de l'incompatibilité des réseaux et leur incapacité à communiquer entre eux, **l'Organisation Internationale de Normalisation** (**ISO**) a examiné des structures de réseau telles que DECNET, SNA et TCP/IP afin d'en dégager un ensemble de règles. À la suite de ces recherches, l'ISO a mis au point un modèle de réseau pour aider les fournisseurs à créer des réseaux compatibles avec d'autres réseaux.

4.2 Rôle et avantages

Le modèle de référence **OSI** (Open System Interconnection - interconnexion de systèmes ouverts), publié en 1984, a ainsi été créé comme une **architecture descriptive**. Ce modèle a offert aux fournisseurs un ensemble de **normes** assurant une **compatibilité** et une **interopérabilité** accrues entre les divers types de technologies réseau produites par de nombreuses entreprises.

Le modèle OSI décrit la façon dont une **communication entre deux ordinateurs doit se décomposer**. Il prévoit **sept couches** et stipule que **chacune de ces couches doit être isolée des autres par une interface bien définie**. **Chacune** des couches correspond à une **fonctionnalité particulière d'un réseau.** Le découpage du réseau en sept couches présente les **avantages** suivants :

- il permet de diviser les communications sur le réseau en **éléments plus simples**, ce qui permet de les comprendre plus facilement;
- il **uniformise les éléments du réseau** afin de permettre le développement et le soutien **multi constructeur**;
- il permet à **différents types de matériel et de logiciel réseau de communiquer** entre eux;
- il **empêche les changements apportés à une couche d'affecter les autres couches**, ce qui assure un développement plus rapide;
- il **divise les communications sur le réseau en éléments plus petits**.

De nombreux **projets** de développement ont tenté de **créer un système réseau totalement conforme à l'architecture OSI**, mais **aucun produit n'en est sorti**.

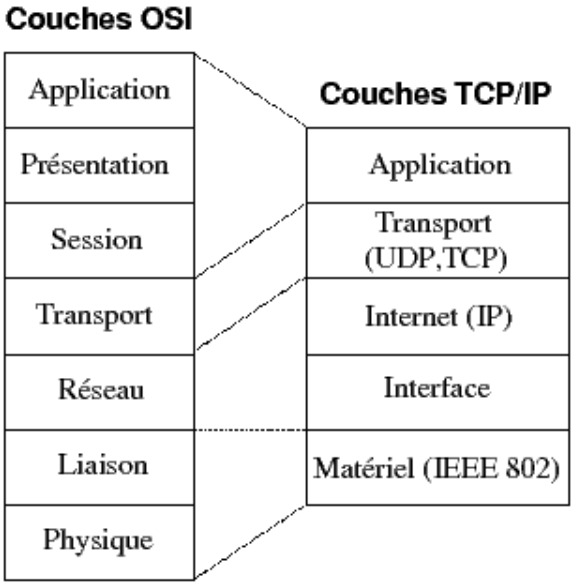


Fig. 3. Correspondance OSI et TCP/IP

Les systèmes d'exploitation réseau de la plupart des fabricants n'utilisent que **trois ou quatre protocoles**. Le modèle OSI reste un outil de référence courant pour les professionnels.

4.3 Les couches

Les couches 1, 2, 3 et 4 sont dites basses alors que les couches 5, 6 et 7 sont dites hautes.

Couches hautes	***Couche 7***	**Couche application**
	Couche 6	**Couche présentation**
	Couche 5	**Couche session**
Couches basses	***Couche 4***	**Couche transport**
	Couche 3	**Couche réseau**
	Couche 2	**Couche liaison de données**
	Couche 1	**Couche physique**
Paquet envoyé sous forme de flux de bits		

Fig. 4. Les couches du modèle OSI

Chaque couche du modèle OSI est constituée **d'éléments matériels et logiciels** et offre un **service** à la couche située immédiatement au-dessus d'elle.

v · d · m	Couches du modèle OSI [masquer]
7. Application	Gopher · SSH · NNTP · DNS · SNMP · XMPP · SMTP · POP3 · IMAP · IRC · VoIP · WebDAV · SIMPLE · HTTP · FTP · Telnet · SILC · TFTP · DHCP · H.323 · SIP · RTSP · TCAP
6. Présentation	ASCII · Vidéotex · Unicode · MIME · TDI · ASN.1 · XDR · UUCP · NCP · AFP · SSP
5. Session	AppleTalk · TLS
4. Transport	TCP · UDP · SCTP · RTP · SPX · DCCP
3. Réseau	BOOTP · RARP · NetBEUI · IPv4 · IPv6 · ARP · IPX · ICMP · OSPF · Babel · RIP · IGMP · IS-IS · CLNP · X.25 · BGP
2. Liaison	AFDX · Ethernet · LLC · Anneau à jeton · LocalTalk · FDDI · X.21 · Frame Relay · Bitnet · CAN · Wi-Fi · PPP · HDLC · STP · ATM · IEEE 802.3ad (LACP) · ARINC 429 · MIL-STD-1553 · I²C
1. Physique	Codage bipolaire · BHDn · CSMA/CD · CSMA/CA · NRZ · NRZI · NRZM · Manchester · Manchester différentiel · Miller · RS-232 · RS-449 · V.21-V.23 · V.42-V.90 · Câble coaxial · 10BASE2 · 10BASE5 · Paire torsadée · 10BASE-T · 100BASE-TX · 1000BASE-T · ISDN · PDH · SDH · T-carrier · E-carrier · EIA-422 · EIA-485 · SONET · ADSL · SDSL · VDSL · DSSS · FHSS · HomeRF · IrDA · USB · IEEE 1394 · Wireless USB · Bluetooth

Fig. 5. Quelques outils propres à chaque couche

4.4 Fonctionnement

Supposons un message devant être transmis d'une **machine émettrice A** vers une machine **réceptrice B**. Ce message est, par exemple, généré par une application tournant sur la machine A. Il va franchir les différentes couches du modèle OSI sur la machine A. Il **transite** via les différents nœuds du réseau. Chacun de ceux-ci va traiter le message au niveau des couches basses du modèle OSI. Lorsqu'il atteint la **machine B**, il remonte à travers l'ensemble des couches du modèle OSI jusqu'à l'application chargée de le traiter.

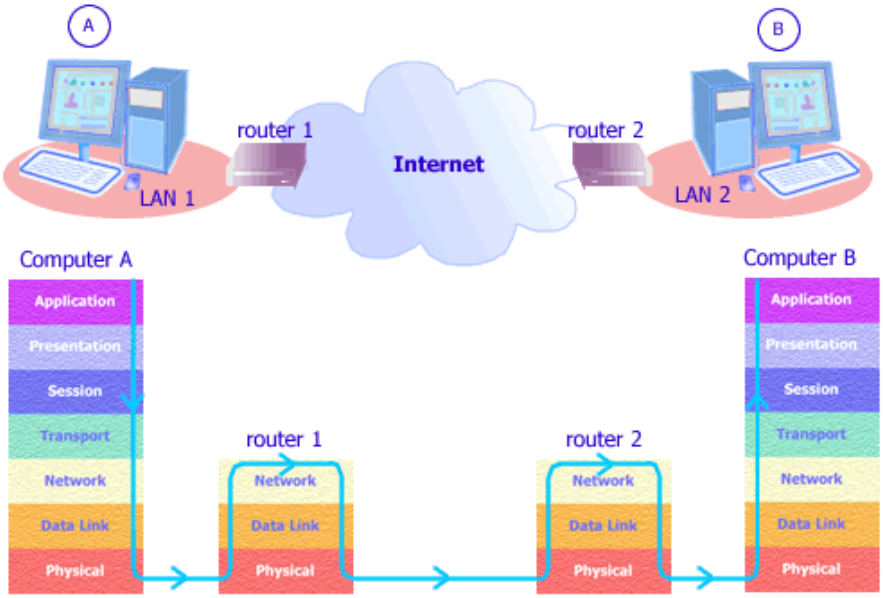


Fig. 6. Les couches du modèle OSI (anglais)

En résumé, le rôle de ces sept couches est le suivant :

7. Application Cette couche est celle que les PROGRAMMES D'APPLICATION perçoivent. Les messages qui doivent être envoyés sur le réseau entrent dans le modèle OSI par cette couche (courrier électronique, transfert de fichiers), traversent les différentes couches vers le bas jusqu'à la couche 1 (physique), arrivent à l'autre station de travail et remontent les couches jusqu'à ce qu'ils atteignent l'application de l'autre ordinateur en traversant sa propre couche Application. Cette couche se distingue des autres couches du modèle OSI en ce sens qu'elle ne fournit aucun service aux autres couches du modèle OSI, mais seulement aux applications à l'extérieur de celui-ci.

6. Présentation Lorsque des ordinateurs IBM, Apple, DEC, NeXT et Burroughs veulent communiquer entre eux, il faut bien entendu effectuer un certain nombre de CONVERSIONS et de réorganisations des octets. La couche Présentation s'assure que les informations envoyées par la couche application d'un système sont lisibles par la couche application d'un autre système. Elle se charge également de la COMPRESSION et du CRYPTAGE des données à transmettre puis du décryptage et de la décompression de celles-ci sur l'ordinateur destinataire.

5. Session Cette couche fournit des services à la couche Présentation. Les fonctions de cette couche permettent aux applications fonctionnant sur deux stations de travail de COORDONNER LEURS COMMUNICATIONS en une seule session (qui peut être considérée comme un dialogue très structuré). La couche Session est responsable de la création de la session, de la gestion des paquets envoyés dans un sens et dans l'autre durant celle-ci ainsi que de sa clôture.

Alors que les **couches Application, Présentation et Session se rapportent aux applications, les quatre couches dites inférieures se rapportent au transport des données.**

4. Transport La couche Transport SEGMENTE **les données** envoyées par le système de l'hôte émetteur et les **rassemble** en flux de données sur le système de l'hôte récepteur. La frontière entre la couche Transport et la couche Session peut être vue comme la frontière entre les protocoles d'application et les protocoles de flux de données.

Les questions comme la façon d'assurer la fiabilité du transport entre deux systèmes hôtes relèvent de la couche Transport. La fourniture d'un service fiable lui permet **d'assurer la détection et la correction des erreurs, ainsi que le contrôle du flux d'informations**. Lorsque plusieurs paquets sont traités simultanément, comme lorsqu'un fichier est fragmenté en de multiples paquets à transmettre, la couche Transport contrôle la **SEQUENCE** des composants du message et régule le flux du trafic pour qu'il ne franchisse pas certaines limites. Lorsqu'un **DOUBLON** de paquet arrive, cette couche le DETECTE ET L'ELIMINE. **SPX** et **TCP** sont **des protocoles de la couche Transport.**

3. Réseau Cette couche AIGUILLE **les paquets** en fonction des besoins pour les faire parvenir à leur destination. Elle est responsable de l'adressage et de la livraison des paquets de messages. Alors que, la couche Liaison de données ne connaît que l'ordinateur immédiatement adjacent, la **couche Réseau est responsable de la** ROUTE COMPLETE **d'un paquet, de la source à la destination. IPX** et **IP** sont des exemples de **protocoles de la couche réseau**. Cette couche **traduit** les **adresses logiques** de réseau et les noms en leur contrepartie **physique**.

Liaison de données La couche Liaison de données assure un TRANSIT FIABLE **des données sur une LIAISON PHYSIQUE**. Ainsi, la couche Liaison de données s'occupe de **l'adressage physique** (plutôt que logique), de la **topologie du réseau**, de **l'accès au réseau**. Cette couche connaît la **représentation des données définie par le réseau (structure des bits, modes de codages et jetons)**. C'est à ce niveau, que les **erreurs de transmission** sont détectées. Du fait de sa complexité, la couche de liaison de données est souvent subdivisée en une couche de ***contrôle d'accès au support*** (MAC) et une couche de ***contrôle de liaison logique*** (LLC). La couche de contrôle d'accès au support régit l'accès au réseau (passage de jetons et détection de collisions) et contrôle ce réseau. La couche de contrôle de liaison logique, a pour rôle d'envoyer et de recevoir les messages de données de l'utilisateur. **Ethernet** et **Token Ring** sont des protocoles de la couche de liaison de données.

Couche physique Cette partie du modèle OSI définit les CARACTERISTIQUES PHYSIQUES ET ELECTRIQUES DES CONNEXIONS qui constituent le réseau (**paires torsadées, câbles en fibre optique, câbles coaxiaux, connecteurs, répéteurs, etc.**). Cette couche est assimilable à une couche de niveau **matériel**. Elle convertit les bits en signaux ou l'inverse.

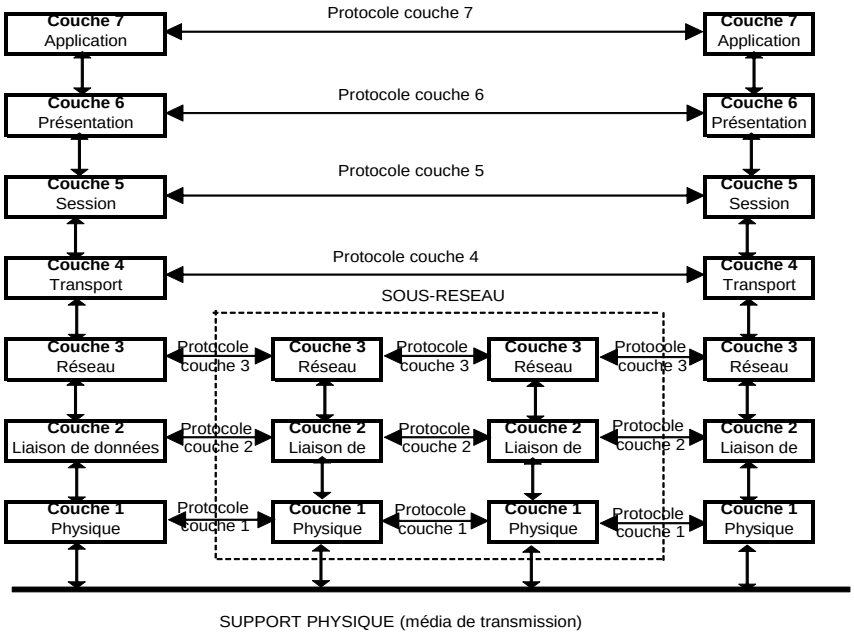


Fig. 7. Schéma de principe de l'échange entre deux ordinateurs

5 L'encapsulation

Au sein d'un réseau, toutes les communications partent d'une source et sont acheminées vers une destination. Les **informations envoyées sur le réseau sont appelées données ou paquets de données.** Si un ordinateur (hôte A) veut envoyer des données à un autre ordinateur (hôte B), les données doivent être **préparées** grâce à un processus appelé ***encapsulation***.

Ce processus conditionne les données **en leur AJOUTANT des informations relatives au protocole avant de les transmettre sur le réseau.** Ainsi, en **descendant** dans les couches du modèle OSI, les données reçoivent des **EN-TETES, DES EN-QUEUES et d'autres informations**.

Pour comprendre comment se produit l'encapsulation, examinons la manière dont les données traversent les couches. Les **données** qui sont envoyées par la source **traversent** les couches. Comme vous pouvez le constater, la présentation et le flux des données échangées subissent des changements au fur et à mesure que les réseaux fournissent leurs services aux utilisateurs. Les réseaux doivent effectuer **cinq étapes de conversion** afin d'encapsuler les données.

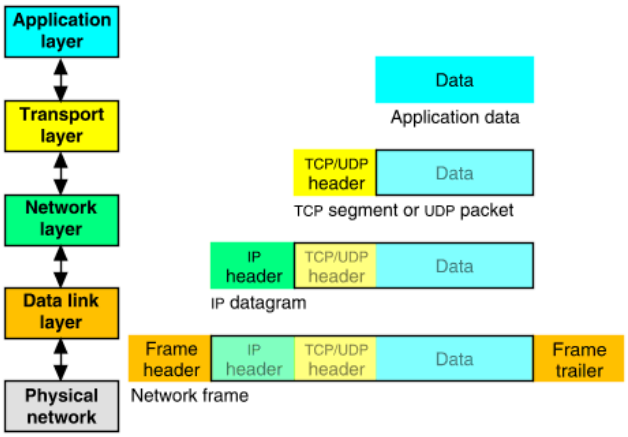


Fig. 8. Illustration du processus d'encapsulation des données.

5.1 Etapes

Couches hautes : Lorsqu'un utilisateur envoie des données sur le réseau, celles-ci vont subir une série de conversions en traversant les couches hautes du modèle OSI afin de les rendre **aptes à circuler dans l'interréseau**.

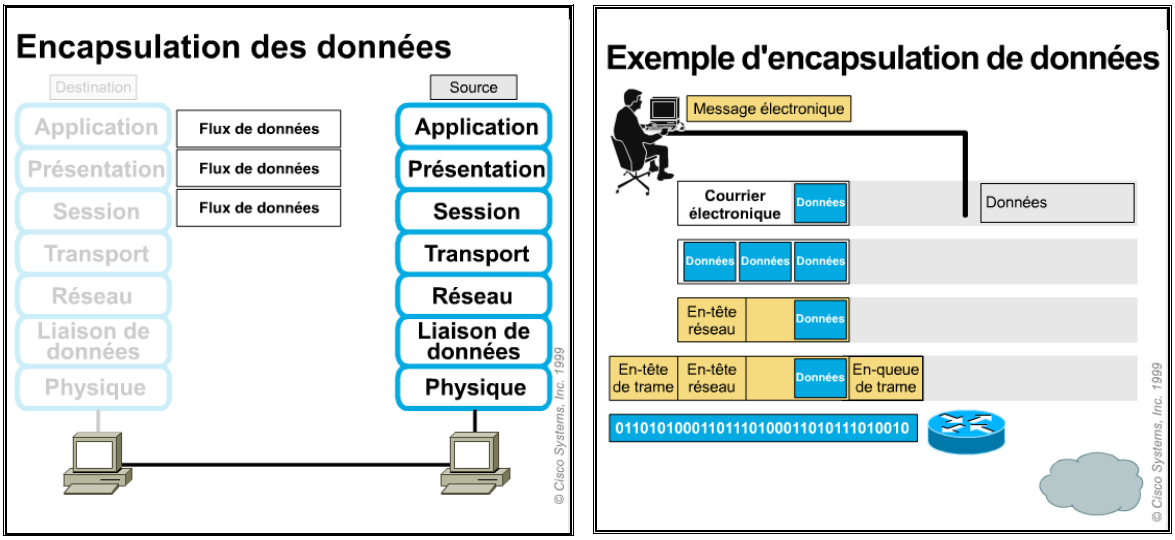


Fig. 9. Encapsulation des données dans les couches hautes du modèle OSI

Couche transport : Cette phase consiste à préparer les données pour le transport sur l'interréseau. En utilisant des **segments**, la fonction de transport s'assure que les systèmes hôtes situés à chaque extrémité peuvent communiquer de façon fiable.

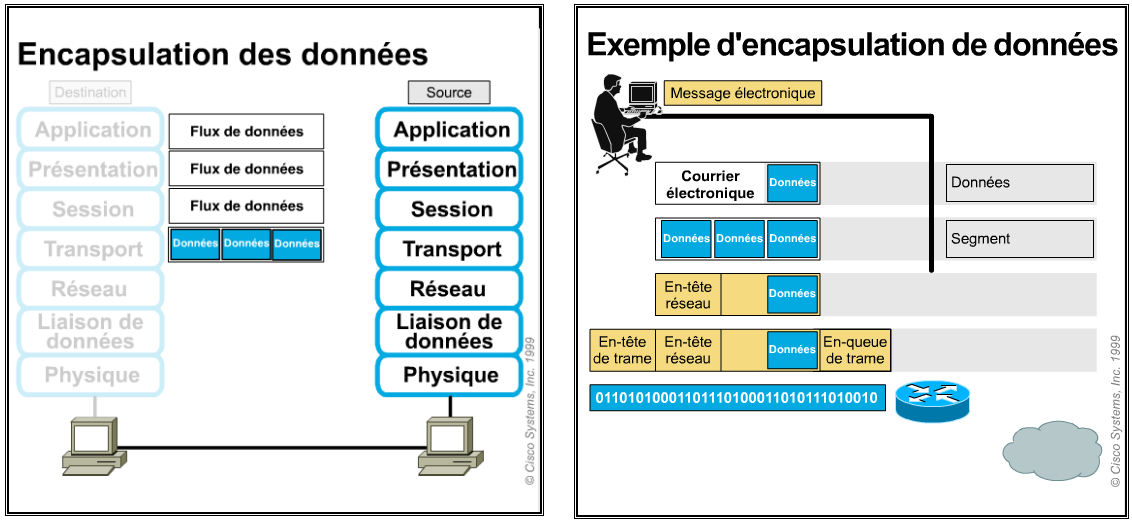


Fig. 10. Encapsulation des données par la couche Transport du modèle OSI

Couche réseau : Ensuite, les données sont organisées en paquets contenant un **en-tête réseau constitué des adresses logiques d'origine et de destination**. Ces adresses aident les unités réseau à acheminer les paquets dans le réseau suivant un chemin déterminé.

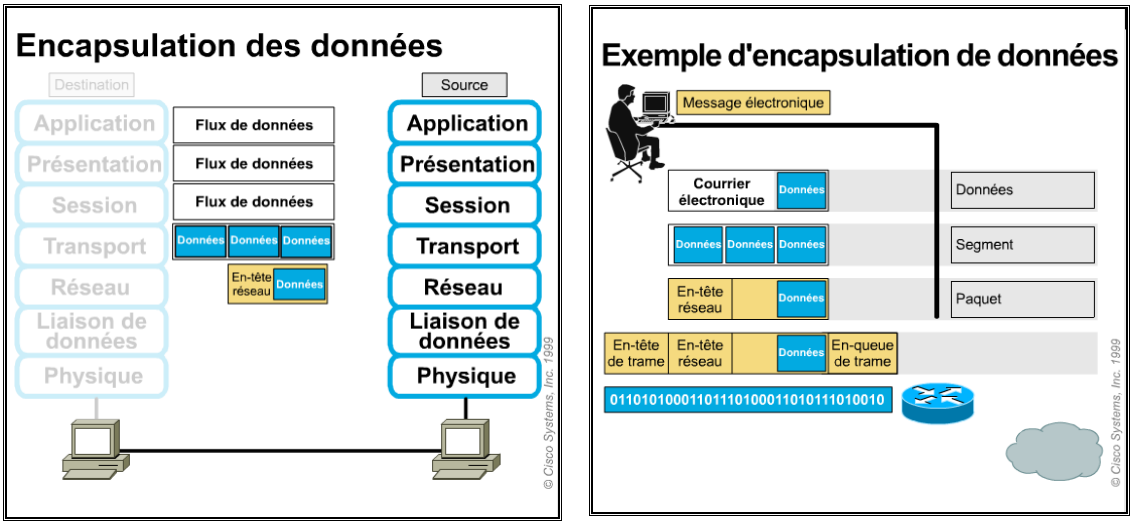


Fig. 11. Encapsulation des données par la couche Réseau du modèle OSI

Couche liaison de données : Lors de la phase suivante, **l'adresse locale** est ajoutée à **l'en-tête de liaison**. Chaque unité réseau doit placer le paquet dans une **trame**. La trame permet d'établir la connexion avec la *prochaine unité réseau directement connectée* dans la liaison. Chaque unité se trouvant sur le chemin réseau choisi doit effectuer un verrouillage de trame pour pouvoir se connecter à la prochaine unité.

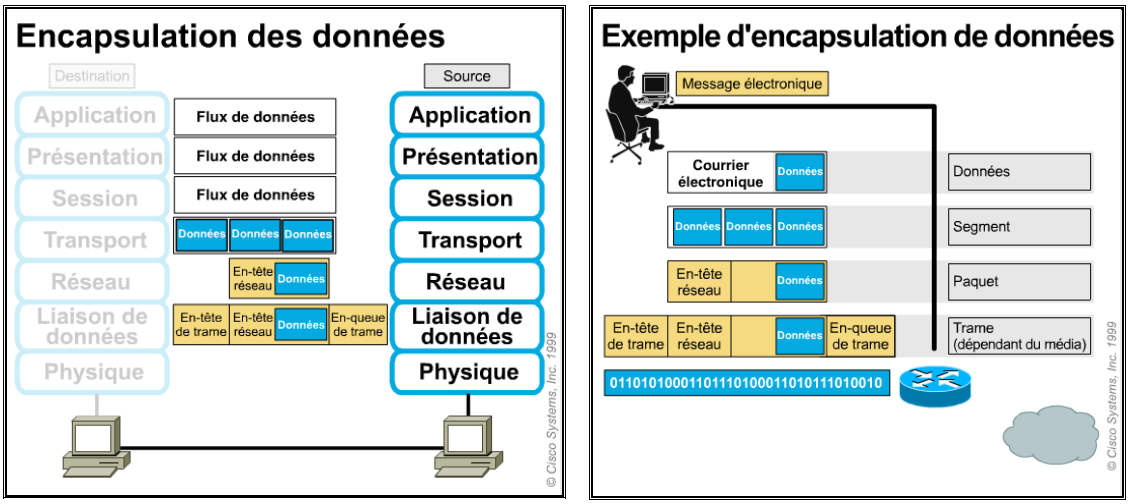


Fig. 12. Encapsulation des données par la couche Liaison de données

Couche physique : Enfin, la **trame** doit **être convertie en une série de bits pour la transmission sur le média** (habituellement un fil). Une fonction de synchronisation permet aux unités de distinguer ces bits lorsqu'ils circulent sur le média. Tout au long du trajet suivi dans l'interréseau physique, le média peut varier. Ainsi, le message électronique peut provenir d'un réseau local, traverser le backbone d'un campus, sortir par une liaison WAN pour atteindre sa destination sur un autre LAN éloigné.

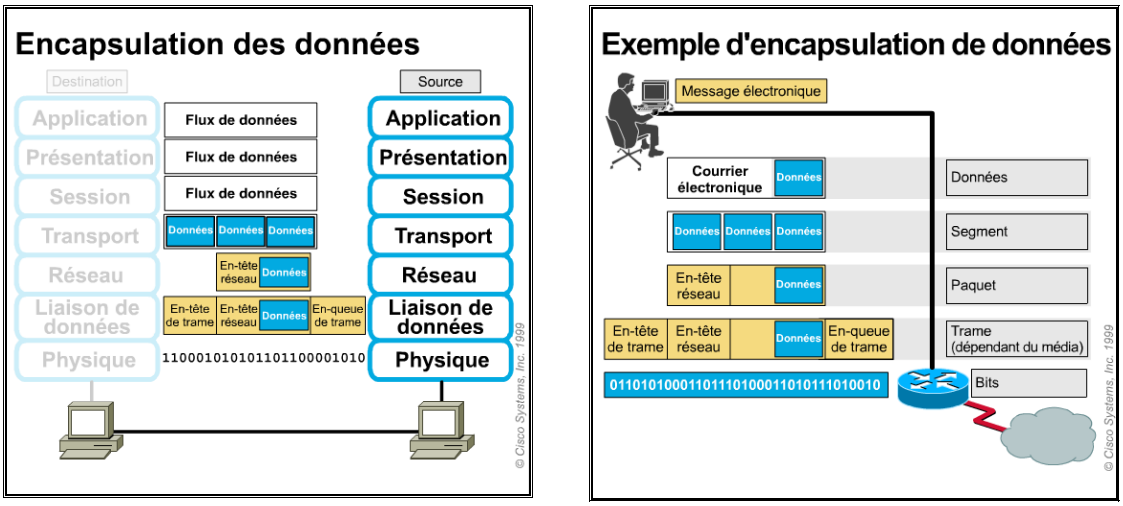


Fig. 13. Encapsulation des données par la couche Physique du modèle OSI

Nous pouvons donc constater que des **en-têtes et en-queues sont ajoutés au fur et à mesure que les données descendent dans les couches du modèle OSI.** Elles seront décodées par les couches correspondantes de la machine destination.

5.2 Désignation des données au niveau de chaque couche

Afin de permettre l'acheminement des paquets de données entre l'ordinateur source et celui de destination, **chaque couche du modèle OSI au niveau de l'ordinateur source doit communiquer avec sa couche homologue sur l'ordinateur de destination**. Cette forme de communication est appelée ***communication d'égal à égal*.**

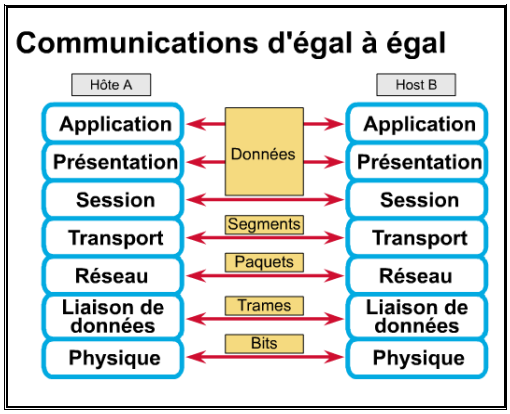


Fig. 14. Communication d'égal à égal entre couches correspondantes du modèle OSI

Au cours de ce processus, le **protocole de chaque couche sur la machine source** assure **l'échange d'informations avec la couche homologue sur la machine de destination** comme l'illustre la figure. Les structures de données manipulées par chaque couche s'appellent ***unités de données de protocole*** (*ou PDU).*

Chaque couche s'appuie sur les services de la **couche OSI sous-jacente**. Pour fournir ce service, la couche inférieure a recours à l'encapsulation pour placer l'unité de données de protocole de la couche supérieure dans son champ de données. Elle ajoute ensuite les en-têtes et les en-queues dont elle a besoin pour remplir ses fonctions.

Dès que les couches 7, 6 et 5 ont ajouté leurs informations, la couche 4 en ajoute d'autres. L'unité de données de protocole de couche 4, est appelée ***segment***.

Ensuite, la couche **réseau** fournit un service à la couche transport, qui présente les données au sous-système de l'interréseau. La couche réseau est chargée de déplacer les données à travers l'interréseau. Pour ce faire, elle encapsule les données et leur annexe un en-tête de manière à créer un ***paquet*** (soit la PDU de couche 3). L'en-tête contient les informations requises pour effectuer le transfert, notamment les **adresses logiques de la source et de la destination**.

La couche **liaison de données** fournit un service à la couche réseau. Elle encapsule les informations de couche réseau dans une ***trame*** (PDU de couche 2). L'en-tête de trame contient les informations (des **adresses physiques**, notamment) nécessaires à l'exécution des fonctions de liaison.

La couche **physique** fournit ensuite un service à la couche liaison de données. Elle **code** la trame de liaison en une série de bits en vue de la transmettre sur un média (habituellement un **fil**) au niveau de la couche 1.

6 La couche physique

La couche physique ne s'occupe que de la **transmission des bits de façon brute**. Il faut veiller à ce **qu'un bit à 1 envoyé par l'émetteur soit reçu comme un bit à 1.**

Les normes et standards de la couche physique définissent **le type de signaux émis** (modulation, puissance, portée), la **nature et les caractéristiques des supports** (câble, fibre optique), les **sens de transmission**, le **codage et la synchronisation** de bits.

La couche physique a également comme objectif de **fixer les caractéristiques des matériels utilisés** pour **relier physiquement** les équipements d'un réseau.

6.1 Types de transmission

Dans une transmission en signal de base, il est nécessaire de définir des niveaux de tension correspondant à un bit 0 ou 1. On peut considérer que le 0 correspond à une tension nulle alors que la valeur 1 est représentée par une tension +V.

Supposons que l'on transmette l'octet suivant : **01100100**. On a donc le signal de la figure ci-dessous.

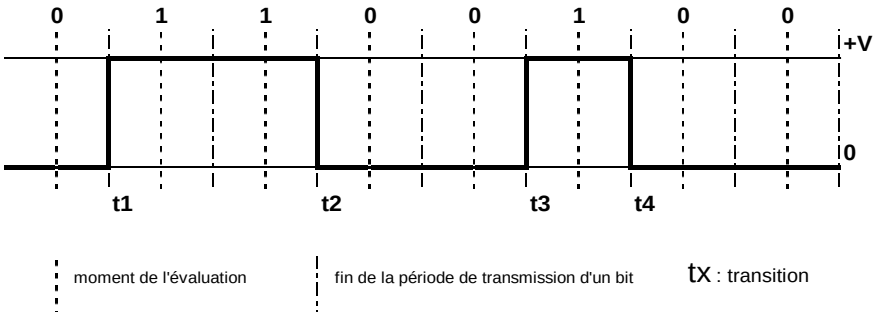


Fig. 15. Transmission en signal de base

Pour que le récepteur puisse interpréter correctement le signal, il faut que l'émetteur et lui définissent **l'instant où le signal doit être évalué**. Supposons que cette évaluation a lieu au **milieu de la durée de transmission d'un bit**. Chaque **transition** du signal de 0 à +V ou inversement correspond au début de la transmission d'un nouveau bit. Le récepteur peut ainsi **se recaler** avec l'émetteur puisqu'il sait qu'une demi durée de transmission plus tard il doit évaluer le signal pour l'interpréter.

Lors de la transmission d'une séquence de plusieurs bits identiques, il peut y avoir un **décalage** entre l'émetteur et le récepteur et par conséquent des erreurs d'interprétation. On a donc intérêt à multiplier le nombre de transition afin d'aider la synchronisation entre émetteur et récepteur.

Le signal tout ou rien, qui vient d'être envisagé, présente un autre inconvénient, au cours de la transmission le signal se superpose avec des signaux parasites (bruit électronique) véhiculé par la ligne. Il peut être difficile de distinguer la tension correspondant à un bit à 1 de celle d'un bit à 0 (il n'y a jamais absence de signal sur la ligne).

Pour éviter ce problème, on peut adopter une codification qui consiste à représenter le 1 par +V et le 0 par –V. On obtient alors un signal **NRZ** (non-retour à zéro). La même séquence donne :

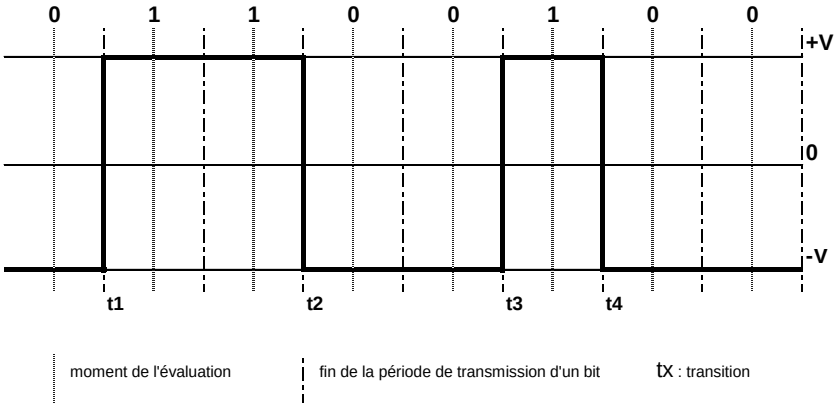


Fig. 16. Transmission en signal de base avec codage NRZ

Avec ce codage, la machine réceptrice ne doit plus détecter l'absence de signal sur la ligne mais le **changement de sens de la tension**. Mais, cette codification ne résout pas le problème du décalage entre l'émetteur et le récepteur lors de la transmission de longue séquence de bits identiques.

Différentes codifications ont été imaginées afin d'augmenter le nombre de transitions.

Le codage Manchester est un codage biphasé, c'est-à-dire sans retour à zéro. Ici, le signal change au milieu du temps de transmission de chaque bit. Pour coder un 0, le courant est négatif sur la première moitié de l'intervalle et positif sur l'autre moitié. Pour coder un 1, on utilise le codage inverse.

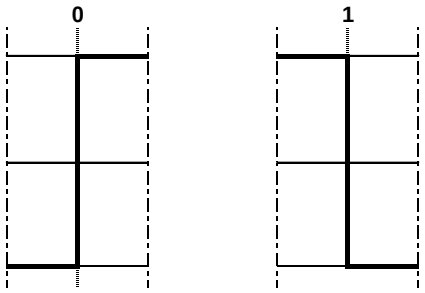


Fig. 17. Codage Manchester

Notre séquence devient :

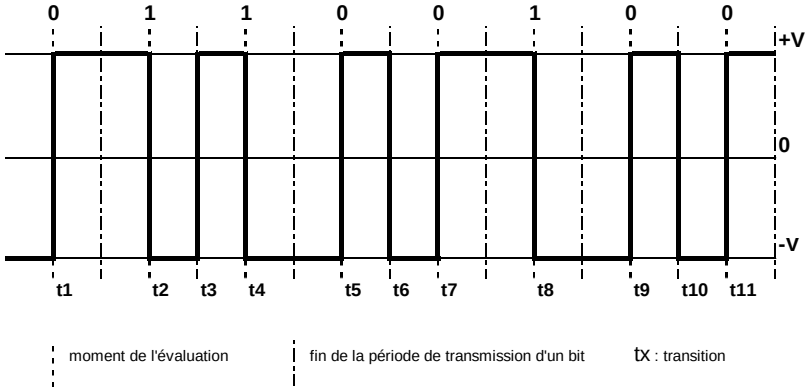


Fig. 18. Transmission en signal de base avec codage Manchester

7 La couche liaison de données

Les véritables messages sont composés non pas de simples bits mais de **groupes** de bits ayant une **signification**. La couche liaison de données reçoit les messages, appelés **trames**[1], provenant des couches supérieures. Une des fonctions principales de cette couche consiste à désassembler ces trames en bits de transmission ou de les reconstituer à partir des bits reçus.

La couche liaison de données est responsable du **transfert sans erreur** de ces trames d'un ordinateur à l'autre par l'intermédiaire de la couche physique. Cela permet à la couche réseau d'assurer en principe un transfert sans erreur sur le réseau.

[1] Trame de données : structure logique et organisée (niveau liaison de données) dans laquelle les données sont placées.

En principe, lorsque la couche liaison de données envoie une trame, elle attend un **accusé de réception** de la part du récepteur. La couche liaison de données du récepteur détecte les problèmes qui se sont éventuellement produits avec la trame lors de la transmission. **Les trames qui n'ont pas fait l'objet d'un accusé de réception** ou les trames qui ont été endommagées lors de la transmission, sont **de nouveau envoyées**.

Son rôle est de **détecter**, si possible, et de corriger **les erreurs dues au support physique** et de signaler à la couche réseau les erreurs irrécupérables.

Nous commencerons par aborder le problème des différentes techniques de détection et de correction d'erreur.

7.1 La parité verticale

Les systèmes les plus simples font appel à la parité qui ajoute, à chaque bloc de i bits (i= 7 ou 8) émis, un bit de parité de telle sorte que parmi les i + 1 bits transmis, le nombre de bits à 1 soit toujours pair (ou impair).

Par exemple, dans le cas d'une **PARITE PAIRE**, si le bloc initial de 7 bits est égal à **1001001**, le bloc de 8 bits émis sera **10010011**. A la **réception**, le décodeur calcule le nombre de bits à 1 et dans le cas de notre parité paire, la transmission sera considérée **correcte** si le nombre est pair. Par contre, si le nombre de bits à 1 est **impair**, on peut conclure qu'il y a eu **erreur** de transmission. On ne peut pas détecter le ou les bits erronés. On peut juste déduire qu'il y a eu erreur et demander la réémission du bloc en erreur.

Source (PARITE PAIRE) :	**Récepteur:**
1 0 0 1 0 0 1 1	1 0 0 **0** 0 0 1 1
données parité	faux → erreur détectée
	1 0 0 **0** 0 0 **0** 1
	correct → erreurs non détectées.

Comme on peut le constater dans le deuxième exemple ci-dessus, cette technique ne permet malheureusement **pas de détecter un nombre pair d'erreurs**. Deux erreurs consécutives dans le même caractère se masquent mutuellement, la parité du bloc reste en effet inchangée dans ce cas. Le bit de parité caractère permet de détecter un nombre impair d'erreurs dans la transmission du caractère mais sans pouvoir les localiser donc les corriger.

7.2 La parité longitudinale

Il existe un autre type de parité, dite **parité longitudinale**. Elle consiste, à ajouter à chaque bloc de caractères, un champ de contrôle supplémentaire composé de la façon suivante : on ajoute à chaque colonne (représentant une suite de bits de même rang) un bit de parité calculé comme la parité verticale. On effectue la même opération sur le rang des bits de parité verticale. On obtient ainsi un caractère supplémentaire appelé caractère de **Redondance Longitudinale** (Longitudinal Redundancy Check ou LRC).

Considérons l'exemple suivant et essayons d'en déterminer les limites. Le système travaille en parité paire.

									Parité verticale
	0	1	0	1	0	0	1	0	**1**
	1	0	0	1	0	1	1	0	**0**
	0	1	1	0	0	1	0	1	**0**
	1	1	1	0	0	1	0	1	**1**
	0	1	0	0	0	1	0	0	**0**
	1	1	0	0	1	1	0	0	**0**
	1	0	0	0	1	0	0	0	**0**
	1	1	0	0	1	1	1	0	**1**
Parité horizontale	**1**	**0**	**0**	**0**	**1**	**0**	**1**	**0**	**1**

S'il se produit une erreur de transmission, non seulement elle est détectée mais peut également être corrigée comme le montre l'exemple ci-dessous.

									Parité verticale
	0	0	0	1	0	0	1	0	1 **Erreur**
	1	0	0	1	0	1	1	0	**0**
	0	1	1	0	0	1	0	1	**0**
	1	1	1	0	0	1	0	1	**1**
	0	1	0	0	0	1	0	0	**0**
	1	1	0	0	1	1	0	0	**0**
	1	0	0	0	1	0	0	0	**0**
	1	1	0	0	1	1	1	0	**1**
Parité longitudinale	**1**	**0**	**0**	**0**	**1**	**0**	**1**	**0**	**1**
		Erreur							

Même s'il se produit plusieurs erreurs, elles peuvent, généralement, être détectées mais plus forcément corrigées. C'est le cas lorsque deux erreurs se produisent dans le même caractère ou le même rang

									Parité verticale
	0	0	0	1	0	0	0	0	1 **Erreur non détectée**
	1	0	0	1	0	1	1	0	**0**
	0	1	1	0	0	1	0	1	**0**
	1	1	1	0	0	1	0	1	**1**
	0	1	0	0	0	1	0	0	**0**
	1	1	0	0	1	1	0	0	**0**
	1	0	0	0	1	0	0	0	**0**
	1	1	0	0	1	1	1	0	**1**
Parité longitudinale	**1**	**0**	**0**	**0**	**1**	**0**	**1**	**0**	**1**
		Erreur					**Erreur**		

Même si les erreurs de transmission augmentent, il reste possible soit de les détecter et de les corriger soit de les détecter sans pouvoir remédier au problème. Dès que deux erreurs se produisent dans la même ligne ou la même colonne, elles restent détectables mais non corrigeables. Cette situation est représentée à la figure suivante.

									Parité verticale
0	0	0	1	0	0	0	0	**1**	**Erreur non détectée**
1	0	0	1	0	1	1	0	**0**	
0	1	1	0	0	1	0	1	**0**	
1	1	1	0	0	0	0	1	**1**	**Erreur**
0	1	0	0	0	1	0	0	**0**	
1	1	0	0	1	1	0	0	**0**	
1	0	0	0	1	0	0	0	**0**	
1	1	0	0	1	1	1	0	**1**	
Parité longitudinale	**1**	**0**	**0**	**0**	**1**	**0**	**1**	**0**	**1**
		Erreur				**Erreur**	**Erreur**		

Cette détection des erreurs n'est pas fiable, il suffit que les erreurs se produisent dans les deux mêmes rangs de deux caractères pour tromper le contrôle à l'aide des parités verticale et longitudinale comme le montre la figure suivante.

									Parité verticale
0	0	0	1	0	0	0	0	**1**	**Erreur non détectée**
1	0	0	1	0	1	1	0	**0**	
0	1	1	0	0	1	0	1	**0**	
1	1	1	0	0	1	0	1	**1**	
0	1	0	0	0	1	0	0	**0**	
1	1	0	0	1	1	0	0	**0**	
1	0	0	0	1	0	0	0	**0**	
1	0	0	0	1	1	0	0	**1**	**Erreur non détectée**
Parité longitudinale	**1**	**0**	**0**	**0**	**1**	**0**	**1**	**0**	**1**
	Erreur non détectée					**Erreur non détectée**			

Lorsque les erreurs sont disposées en rectangle, plus aucune détection n'est possible. De plus, ces techniques, étudiées pour les systèmes de sauvegarde sur bande magnétique, ne sont pas vraiment applicables pour la transmission. En effet dans ce cas, la communication est du type sériel et les bits, représentant un caractère, sont expédiés les uns à la suite des autres.

Le système de détection d'erreurs couramment utilisé en transmission de données est celui faisant appel aux codes cycliques (Cyclic Redundant Check : CRC). Les bits transmis, appelés mots de code, sont représentés sous forme polynomiale dans laquelle la suite de bits à transmettre $M = m_1\ m_2\ m_3\ \ldots m_n$ est représentée par le polynôme m(x) tel que :

$$M(x) = m_1\ x^{n-1} + m_2\ x^{n-2} + m_3\ x^{n-3} + \ldots. + m_{n-1}\ x\ + m_n$$

Par exemple, la suite de bits 11001001 est représentée par le polynôme suivant :

$$x^7 + x^6 + 0\ x^5 + 0\ x^4 + x^3 + 0\ x^2 + 0\ x + 1 \text{ ou } x^7 + x^6 + x^3 + 1$$

Le principe de détection des erreurs est fondé sur l'utilisation du reste de la division polynomiale M(x) par un polynôme diviseur G(x), donnant un quotient Q(x) et un reste R(x). Le reste R(x) est transmis au récepteur en tant que bits de contrôle derrière la suite de bits de M.

...

Outre son rôle de détection de transmission, la couche liaison de données est chargée d'exécuter les fonctions suivantes :

- **Etablissement et libération de la liaison de données** sur des connexions physiques préalablement activées.
- **Supervision du fonctionnement** de la liaison de données, selon le mode de transmission (synchrone ou asynchrone), la nature de l'échange (unidirectionnel, bidirectionnel à l'alternat ou simultané), le type de liaison (point à point, multipoint, boucle).
- **Définition de la structure syntaxique des messages** valides, de la manière d'enchaîner émission et réception selon un protocole spécifique normalisé ou non.

Le protocole BSC (Binary Synchronus Communication), procédure orientée caractères, est basé sur la transmission de blocs de caractères représentés principalement en ASCII (7 bits) ou en EBCDIC (8 bits) avec acquittement à l'alternat.

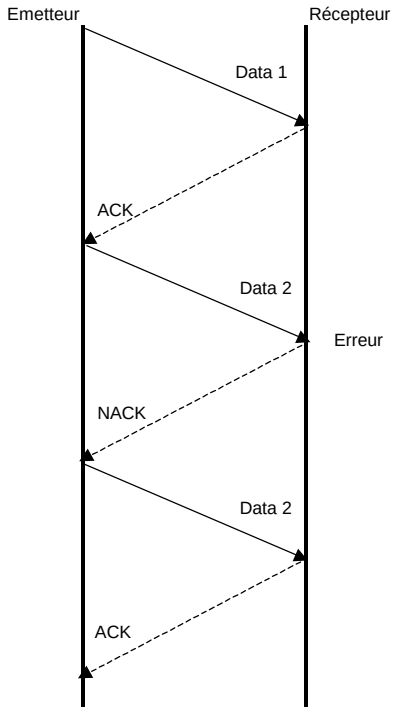


Fig. 19. Protocole BSC

Les erreurs détectées sont corrigées par demande de retransmission. Les messages émis selon le protocole BSC le sont sous forme de blocs de taille appropriée aux possibilités de la ligne.

7.3 Sous-niveaux

La couche liaison de données est découpée en deux sous niveaux pour les réseaux locaux :

- Les techniques d'accès au support (**Medium Access Control**) – niveau MAC. Ce niveau définit le moyen d'accéder au support de transmission commun à toutes les stations connectées sur le réseau local
- La procédure de communication proprement dite, désignée sous le terme de niveau de contrôle du lien logique (**Logical Link Control**) – niveau LLC. Ce niveau comporte une grande partie des fonctionnalités de la couche liaison de données.

La couche liaison de données est couverte par plusieurs normes de l'IEEE (802.1, 802.2, 802.3, 802.4, 802.5).

- La norme 802.1 concerne les fonctions de gestion de couches définies dans le cadre de l'administration de réseau. Elle concerne tous les types de réseaux locaux.
- La norme 802.2 définit le niveau LLC.
- Les normes 802.3 et suivantes concernent les différents niveaux MAC.
  - 802.3 : CSMA/CD pour le Bus
  - 802.4 : le Bus à jetons (Token Bus)
  - 802.5 : anneau à jetons (Token Ring)
  - 802.6 : définition des réseaux métropolitains

7.3.1 Contrôle d'accès au matériel

La sous-couche MAC de la couche liaison de données assure l'**interface** avec la carte réseau. Ces fonctionnalités permettent de faciliter l'accès au réseau par l'intermédiaire de la carte réseau. Celles-ci comprennent les méthodes de **contrôle d'accès** et la **topologie du réseau.**

La couche liaison de données contrôle aussi la méthode de transmission (synchrone ou asynchrone) utilisée pour accéder au média de transmission.

7.3.2 Gestion de l'adressage

La couche liaison de données gère les **adresses des différents dispositifs du réseau** ce qui permet d'envoyer un message à une machine particulière.

Ces adresses sont appelées **adresses physiques**. Il s'agit d'adresses uniques associées aux composants matériels relatifs au réseau de l'ordinateur. Dans le cas de réseau Ethernet ou Token Ring, cette adresse physique est **enregistrée dans la MEMOIRE ROM DE LA CARTE RESEAU.**

Le format de l'adresse est associé à la méthode de contrôle d'accès au média de transmission en **relation directe avec la sous-couche MAC** de la couche liaison de données. Cela explique pourquoi l'adresse physique est souvent désignée par **ADRESSE MAC**[2].

[2] Sur un réseau informatique, le filtrage par adresse MAC permet de n'**autoriser que certaines machines à se connecter** : il faut au préalable renseigner dans le panneau d'administration de la box l'identifiant unique (adresse MAC) de chaque machine officiellement acceptée (vos ordinateurs, smartphones, consoles, etc.). Le

L'adresse MAC est un numéro unique définit par les constructeurs et composé de douze caractères hexadécimaux. Exemple : 00:07:CB:C4:48:C9. Deux ordinateurs équipés de carte réseau ne peuvent donc pas avoir la même adresse MAC, ce qui permet d'organiser un filtrage, machine par machine.

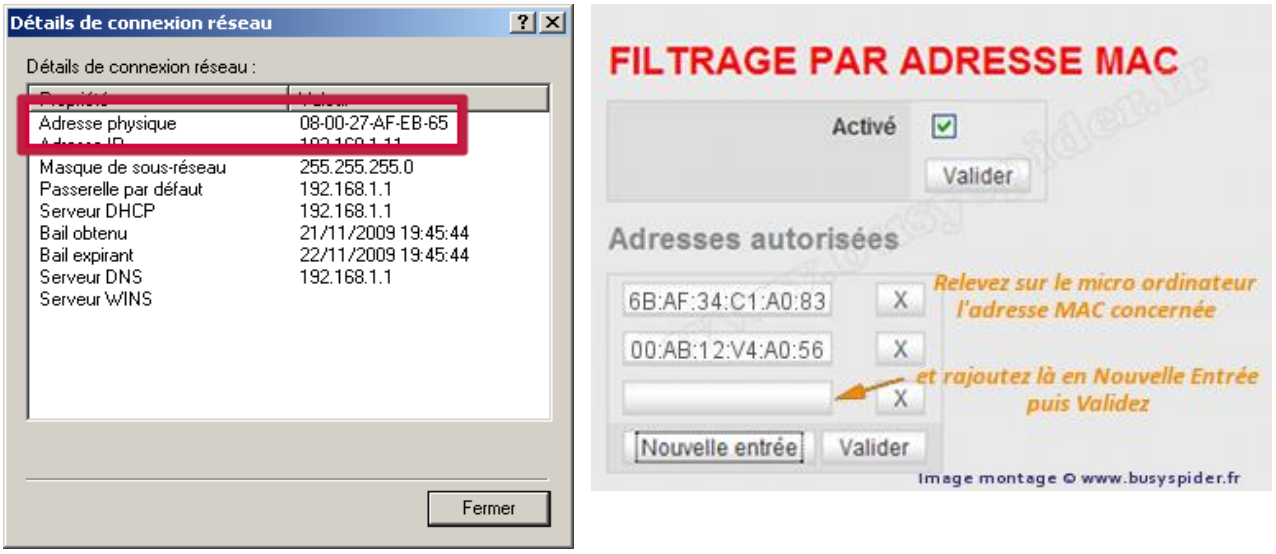


Fig. 20. Adresse Mac - Filtrage par adresse Mac

Dans la plupart des réseaux LAN, les **paquets envoyés vers une station particulière** sont en fait **expédiés à l'ensemble des stations du réseau**. Chacune d'elles est donc obligée d'interpréter la trame suffisamment loin pour déterminer l'adresse du destinataire. **Seule la machine à qui la trame est destinée va la répercuter vers les couches supérieures du modèle OSI**. Les autres machines vont ignorer le reste de la trame.

Dans certaines circonstances, des trames expédiées sur le média de transmission sont destinées à **L'ENSEMBLE DES MACHINES DU RESEAU**. Dans ce cas, l'adresse de destination est une adresse particulière dite adresse de diffusion ou de **BROADCAST** qui va être acceptée par toutes les machines.

La sous-couche LLC de la couche liaison de données assure le **contrôle d'erreurs** et le contrôle de flux pour les liaisons uniques entre des dispositifs communicants. Le contrôle d'erreur, à ce niveau, se contente de **détecter les erreurs de transmission dans les trames reçues et à en demander la retransmission.**

---

filtrage MAC ne concerne ici que les connexions par le **Wi-Fi**. Si vous devez prêtez temporairement votre connexion à un ami de passage, vous pourrez lui proposer de se connecter en Ethernet (câble RJ 45 reliant la box à un ordinateur ou autre matériel réseau). S'il insiste pour se connecter en Wi-Fi, vous devrez alors ajouter son adresse MAC dans la liste des adresses autorisées par le filtrage. Logique !
Le filtrage MAC n'est pas une solution de sécurité aussi performante qu'auparavant, certains pirates parviennent à la contourner facilement : grâce à des logiciels spécialisés (appelés *sniffers*), ils peuvent "écouter" à distance les informations qui circulent sur un réseau Wi-Fi, jusqu'à repérer une ou plusieurs adresses MAC acceptées sur le réseau. Bien que cela soit illégal, les pirates peuvent ensuite modifier l'adresse MAC de leur ordinateur portable, et la faire correspondre avec une adresse acceptée sur le réseau visé. Cette opération peut être assez rapide. Le filtrage MAC reste cependant une sécurité complémentaire qui peut repousser certaines intrusions. Il serait par contre risqué de miser toute la sécurité de son réseau sur ce seul dispositif.

8 La couche réseau

8.1 Rôle

Si la couche **liaison de données gère la communication entre les entités d'un MEME RESEAU**, la couche **réseau** effectue cette opération entre **des entités se trouvant sur des RESEAUX** logiquement **SEPARES.**

Au niveau de la couche liaison de données, les adresses permettent d'identifier les trames de données, et chaque entité est chargée de surveiller le réseau et de réceptionner les trames qui lui sont adressées. Il est clair que sur les **interréseaux**, qui peuvent être de très grande taille et construits à partir de réseaux de types différents, il est **impossible d'acheminer les données uniquement à l'aide de l'adresse physique**. Imaginez que votre carte réseau soit obligée de contrôler chaque paquet provenant de n'importe quel point du réseau Internet afin de rechercher la correspondance avec votre adresse physique, c'est tout à fait impossible.

Les réseaux de plus grande taille nécessitent donc la mise **en œuvre de moyens de ROUTAGE et de FILTRAGE** des paquets de manière à réduire le trafic sur le réseau et les temps de transmission. La couche réseau utilise des algorithmes de routage pour **guider les paquets depuis leur réseau source vers le réseau de destination.** Elle dirige le processus d'adressage et d'acheminement des paquets et **supervise le processus de définition des chemins de livraison des paquets** à travers l'interréseau.

Au niveau de la couche réseau, chaque réseau faisant partie d'un interréseau se voit attribuer une **adresse de réseau servant au routage des paquets**. Ces adresses logiques sont attribuées lors de la configuration des réseaux et doivent être uniques dans l'interréseau.

La plupart des interréseaux sont des réseaux maillés **incluant des voies de données redondantes** pouvant être utilisées pour acheminer les messages. Généralement, un paquet part de l'ordinateur source situé sur un segment de LAN et traverse d'autres segments de LAN jusqu'à atteindre le segment de l'ordinateur de destination.

8.2 Techniques de commutation

Plusieurs techniques de commutation peuvent être utilisées pour acheminer les données d'un segment de réseau vers un autre segment. On peut citer :

- la commutation de circuits
- la commutation de messages
- la commutation de paquets

8.2.1 La commutation de circuits

La commutation de circuits établit un chemin qui demeure **fixe** pour toute la durée de la connexion

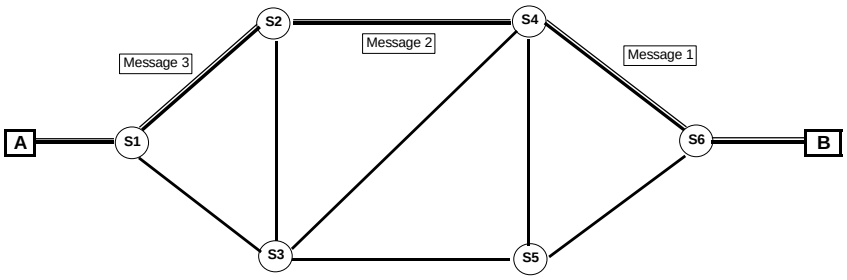


Fig. 21. Acheminement des paquets par commutation de circuits

Mais, établir une connexion entre deux entités peut prendre beaucoup de temps. De plus, comme le chemin dédié ne peut être partagé, il peut résulter un gaspillage de la bande passante disponible. Ce type de réseaux doit disposer d'un excédent de bande passante, leur mise en œuvre a donc tendance à être onéreuse.

8.2.2 La commutation de messages

La commutation de message traite chaque message comme une **entité indépendante**. Chaque message contient des informations d'adressage qui décrivent la destination du message et qui sont utilisées au niveau de chaque commutateur pour transférer le message au commutateur suivant sur le chemin.

Les commutateurs de messages sont programmés de façon à détenir des renseignements sur les autres commutateurs du réseau pouvant être sollicités pour expédier les messages. Ils peuvent aussi posséder des informations sur les itinéraires les plus efficaces à emprunter.

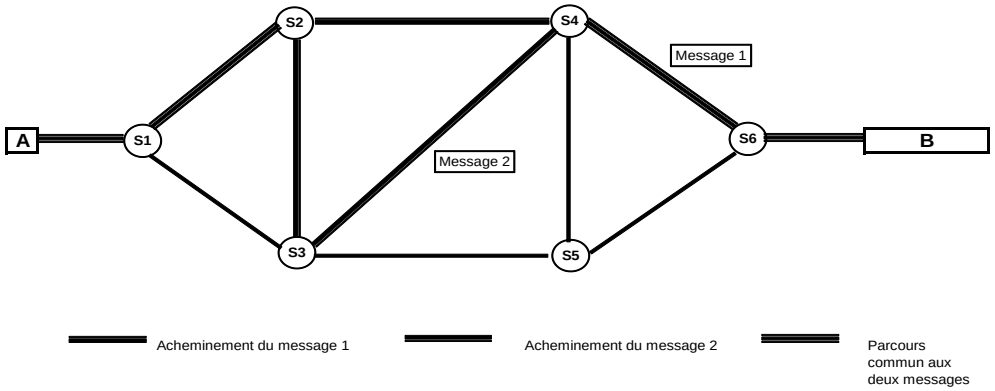


Fig. 22. Acheminement des paquets par commutation de message

La commutation de message assure le transfert complet du message d'un commutateur vers le suivant où il est stocké avant d'être réexpédié vers le commutateur suivant. Les commutateurs de messages sont généralement des ordinateurs à usage général qui doivent disposer d'une capacité de stockage suffisante pour stocker les messages jusqu'à ce que la transmission soit possible.

La commutation de messages est couramment employée pour l'envoi de courrier électronique. Elle se contente de dispositifs relativement peu coûteux pour transmettre les messages et peut fonctionner correctement sur des voies de communication lentes.

La commutation de messages offre les avantages suivants :

- les canaux de données sont partagés par les entités communicantes ce qui améliore le niveau d'exploitation de la bande passante disponible
- les commutateurs de messages peuvent stocker les messages jusqu'à ce qu'un canal se libère réduisant ainsi les probabilités de congestion du réseau
- des priorités de message peuvent être utilisées pour gérer le trafic sur le réseau

Par contre, elle ne convient pas pour les applications exigeant des transmissions en temps réel.

8.2.3 La commutation de paquets

Dans la commutation de paquets, les messages sont divisés en morceaux de plus petite taille appelés **paquets**. Chacun de ceux-ci contient des renseignements sur l'adresse source et l'adresse de destination afin de pouvoir être acheminé indépendamment sur l'inter réseau. Les paquets qui composent le message peuvent emprunter des itinéraires tout à fait différents à travers l'inter réseau et par conséquent connaître des temps d'acheminement différents.

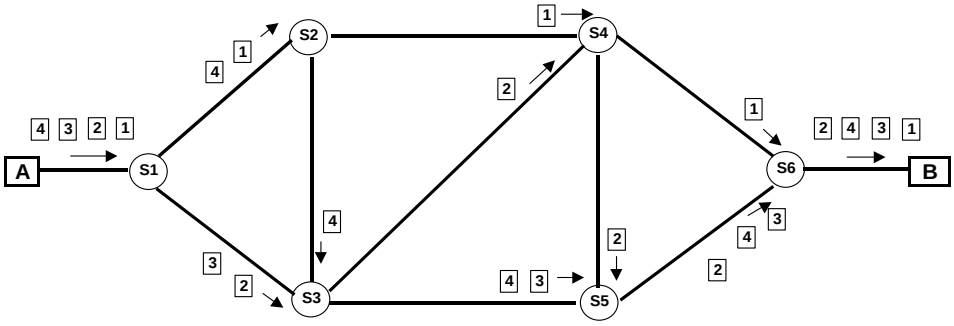


Fig. 23. Acheminement des paquets par commutation de paquets

L'un des avantages de la commutation de paquets sur la commutation de messages réside dans le fait que, contrairement aux messages, les paquets sont limités à une certaine taille. Cela permet au commutateur de gérer les paquets entièrement en mémoire, ce qui élimine le besoin de recourir au stockage temporaire des données sur disque. Les paquets sont ainsi acheminés plus rapidement et plus efficacement à travers le réseau.

La commutation de paquets présente les **avantages suivants** :

- elle permet d'optimiser l'utilisation de la bande passante en autorisant les nombreux dispositifs de routage à transmettre des paquets à travers les mêmes voies de com-

munication du réseau. A tout moment, un commutateur peut acheminer des paquets de plusieurs entités réceptrices différentes, adaptant les itinéraires selon les besoins afin d'obtenir les meilleurs résultats.

- comme les messages complets ne sont pas stockés au niveau des commutateurs avant d'être envoyés, les délais de transmission sont bien plus courts que ceux requis par la commutation de messages.

Les unités de commutation doivent être équipées de grandes quantités de mémoire et de puissance de traitement suffisante pour pouvoir exécuter les protocoles de routage plus complexes que requiert la commutation de paquets. Un système doit être mis en place pour permettre aux unités de commutation de détecter la perte d'un paquet afin d'en demander la retransmission.

8.3 Le contrôle de flux

Le contrôle de flux consiste à gérer les paquets pour qu'ils transitent le plus rapidement possible entre l'émetteur et le récepteur. Il vise à éviter les problèmes de congestion du réseau qui surviennent lorsque trop de messages y circulent.

Parmi les méthodes de contrôle de flux, on peut citer :

- **le contrôle par crédits** : seuls N paquets sont autorisés à circuler simultanément sur le réseau. Un paquet ne peut donc entrer dans le réseau qu'après avoir reçu un jeton qu'il libère dès qu'il est arrivé à destination. Dans cette méthode, les jetons sont banalisés et la principale difficulté réside dans une distribution correcte des jetons entre les différents nœuds afin d'assurer un fonctionnement optimal.
- **le contrôle par jeton dédié** : ici les jetons sont dédiés aux nœuds d'entrée dans le réseau. Chaque nœud gère avec ses jetons une file d'attente des paquets qu'il émet. Lorsqu'un paquet arrive à destination, le récepteur réexpédie le jeton vers le nœud émetteur.

8.4 Le problème de la congestion

Les problèmes de congestion se produisent lorsque les **nœuds d'un réseau saturent leur file d'attente et par conséquent perdent des paquets**. Ce problème peut se produire quels que soient les efforts pour contrôler le flux.

Il faut alors arriver à résoudre le problème sans l'aggraver. Si les **paquets perdus sont réexpédiés** ou si des **messages de gestion de réseau se mettent à circuler en grand nombre** les performances du réseau vont s'écrouler très vite.

On essaie d'éviter le problème de la congestion en autorisant un paquet à ne rester dans le réseau qu'un temps limité fixé par le gestionnaire du réseau. Tout paquet est donc **émis avec une date fixée par une horloge commune au réseau**. Si un nœud s'aperçoit que le temps de présence dans le réseau d'un paquet est dépassé, il le **détruit**. Cela permet de détruire les paquets perdus par erreur d'adressage ou de routage ainsi que ceux bloqués dans un nœud.

Fig. 24. Le problème de la congestion

8.5 Le routage

Le routage des paquets dans un réseau maillé consiste à fixer par quelle ligne de sortie chaque commutateur réexpédie les paquets qu'il reçoit.

Ceci se fait en fonction de la destination finale du paquet et selon une table de routage qui indique pour chaque destination finale quelles sont les voies de sortie possibles.

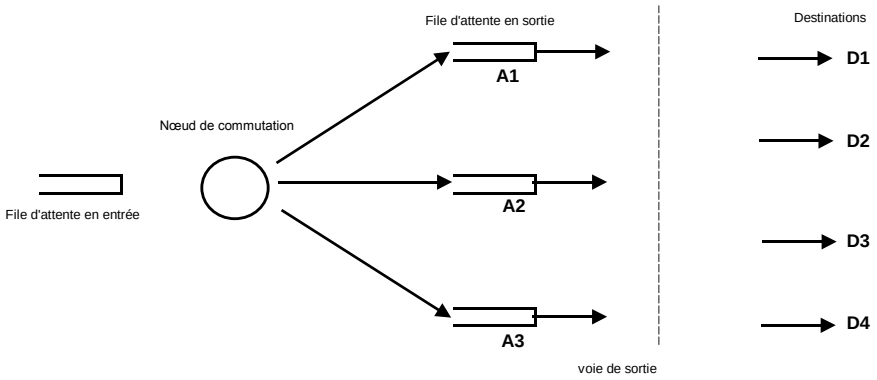


Fig. 25. Schéma de routage

Pour le fonctionnement du routage, voir les chapitres sur l'adressage IP et le protocole RIP.

9 La couche transport

La couche transport comporte les procédures garantissant la **délivrance fiable des messages à leur destinataire**. Le terme "fiable" ne signifie nullement sans erreur mais indique que celles-ci **doivent être détectées**. Si des erreurs, telles que la **perte de données** sont **identifiées**, la couche transport **demande la retransmission** des messages ou bien en **informe les protocoles des couches supérieures pour qu'ils en assurent la correction**.

La couche transport assure **l'interfaçage entre les couches supérieures avec le réseau leur dissimulant la complexité de fonctionnement du réseau.**

Une des fonctions importantes de cette couche est de **DIVISER LES MESSAGES DE GRANDE TAILLE EN SEGMENTS** appropriés à leur transmission sur le réseau.

Parmi les activités de cette couche, on peut citer :

- **La réorganisation des segments** : lorsque de grands messages sont découpés en segments avant d'être envoyés sur le réseau, la couche transport doit **réordonner les segments à leur arrivée** du côté récepteur avant de pouvoir recomposer le message d'origine.
- **Le contrôle d'erreurs** : si des segments sont **perdus** lors de leur transmission ou s'il y a **duplication** de certains d'entre eux, la couche transport doit lancer le processus de correction d'erreur. Cette couche détecte aussi les segments altérés en assurant un contrôle de bout en bout à l'aide de techniques telles que les sommes de contrôle (checksum).
- **Le contrôle de flux de bout en bout** : la couche transport utilise des **acquittements** pour gérer le contrôle de flux de bout en bout entre deux entités connectées. En plus de l'acquittement négatif, certains protocoles de ce niveau peuvent demander la re-transmission des segments les plus récents.

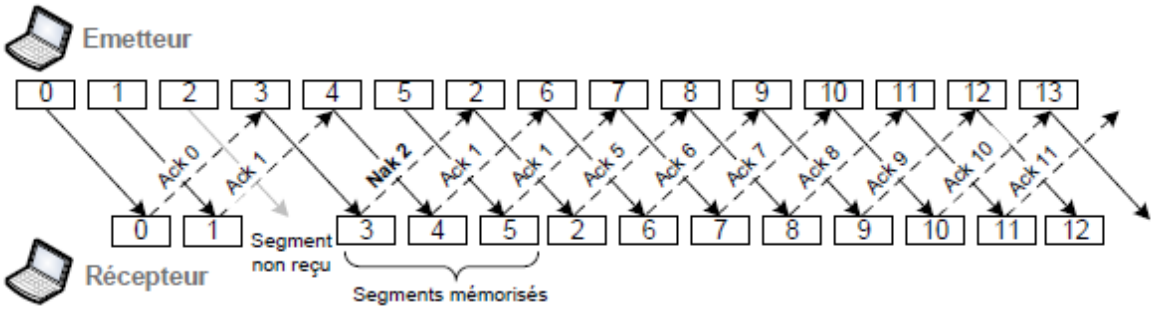
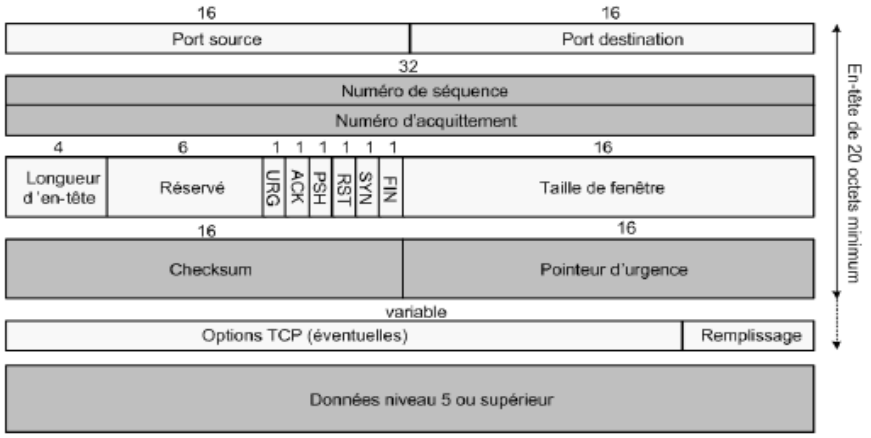


Fig. 26. Envoi des segments



Flags : Urgent, Ack, Psh, Rst, Syn, Fin

Fig. 27. Format du paquet de la couche Transport Control Protocol

10 La couche session

La couche session s'occupe du **dialogue entre deux ordinateurs**, en établissant, en supervisant et en terminant les communications.

Une session est une **connexion formelle entre un demandeur et un fournisseur de services**.
Une session se compose au minimum de 4 phases.

- **Etablissement de la connexion** : durant cette phase, le demandeur de service sollicite l'initiation d'un service. Lors du processus de mise en place, la communication est établie et les règles sont négociées.
- **Transfert des données** : grâce aux règles négociées, chacune des parties est informée de ce qu'elle doit recevoir. La communication est par conséquent efficace et les erreurs sont faciles à détecter.
- **Libération de la connexion** : lorsqu'une session s'achève, la communication est terminée de façon méthodique.
- **Correction d'erreur** : la couche session dispose de mécanismes permettant de rechercher les erreurs dans les paquets réassemblés qui ont été reçus de la couche transport.

La phase d'établissement de la connexion définit les paramètres de la session de communication. Cette phase comprend plusieurs tâches parmi lesquelles nous pouvons citer :

- spécification des services requis.
- **authentification** de la demande de connexion utilisateur et exécution d'autres procédures de sécurité.
- **négociation des protocoles** et paramètres de connexion.
- instauration d'un **contrôle de communication** et reconnaissance des procédures de numérotation et de retransmission.

Une fois la connexion établie, les dispositifs impliqués peuvent initier le dialogue (phase de transfert des données). En plus des données, les entités échangent des acquittements et d'autres données de contrôle qui gèrent le dialogue. La couche session insère des points de synchronisation dans le flot de données et supervise leur réception. En cas d'erreur de transmission, l'ordinateur émetteur peut retransmettre les données en commençant par celles envoyées après le dernier point de synchronisation au lieu de renvoyer la totalité du message.

La couche session peut aussi intégrer des protocoles permettant de **reprendre les connexions ayant été interrompues**. Après la mise en place d'une communication formelle, les entités reconnaissent une **perte de connexion lorsque celle-ci n'a pas été libérée selon les règles**. Un dispositif détecte donc une telle situation lorsque la transmission ou acquittement attendu n'a pas été reçu.

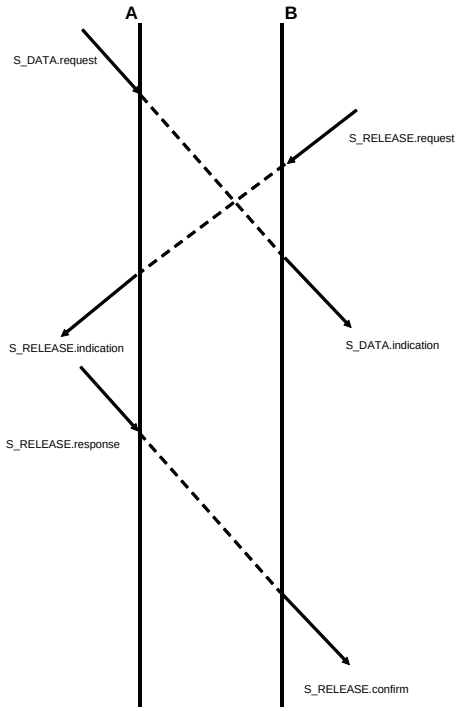


Fig. 28. Libération d'une connexion

L'ouverture d'une session nécessite la négociation de plusieurs paramètres entre les utilisateurs des extrémités, certains sont identiques à ceux de la couche transport à laquelle ils sont passés tels quels. Un paramètre permet, notamment, de décider quelle extrémité aura l'initiative du dialogue dans le cas d'une session bidirectionnelle.

Une session sera terminée par la primitive **S_RELEASE.request** qui réalise une libération ordonnée de la connexion sans perte de données appelée **Terminaison négociée**.

11 La couche présentation

La couche présentation est responsable de la **syntaxe[3] et de la sémantique[4] des informations transportées** en se chargeant notamment de la **représentation des données.**

A ce niveau, les données provenant de la couche application **spécifique à un système** (ordinateur compatible **IBM** ou **Macintosh** par exemple), sont converties dans un **format commun** indépendant du type de machine et mieux adaptés aux protocoles des couches inférieures.

Par exemple, sur les ordinateurs à base de **processeur Intel**, les *nombres entiers sont représentés avec les bits de poids forts à droite et ceux de poids faible à gauche*. Alors que sur les ordinateurs à base de processeur **Motorola** de la famille 68000 (Apple), c'est exactement *l'inverse*[5]. C'est la couche présentation qui va se charger des opérations nécessaires à la bonne interprétation des données transmises.

[3] Syntaxe : branche de la linguistique qui étudie la façon dont les mots se combinent pour former des phrases ou des énoncés dans une langue. La syntaxe s'intéresse à l'ordre des mots,...
[4] Sémantique : branche de la linguistique qui étudie les signifiés (le sens), soit, ce dont parle un énoncé.
[5] Pour plus de précision, nous vous renvoyons au cours de technologie des ordinateurs.

En fait, le modèle OSI définit une syntaxe de transfert commune (syntaxe abstraite numéro 1 – Abstract Syntax Notation 1) utilisée par toutes les applications transmettant des données sur un réseau.

C'est également au niveau de cette couche que sont implantées les techniques de **compression** et de **cryptage** des données.

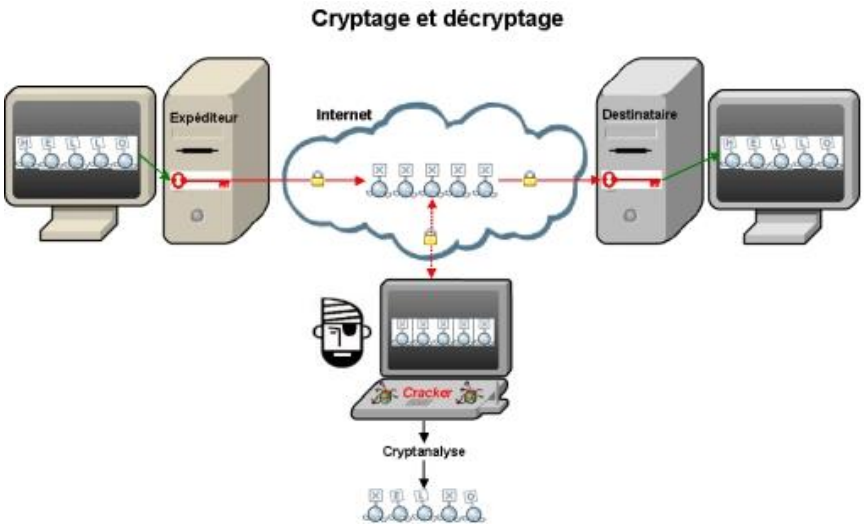


Fig. 29. Cryptage des données

Au niveau de la machine réceptrice, la couche présentation convertit les données de format indépendant reçues du réseau dans le format requis par le système local. Cette conversion peut comprendre les étapes suivantes :

le formatage des données : organisation des données, comportant 4 phases dont

- *Conversion de séquencement de bits*
- *Conversion de séquencement d'octets*
- *Conversion de codage de caractères*
  Les tables de codage utilisées pour représenter les caractères sous forme binaire peuvent varier d'un ordinateur à l'autre. Un codage ASCII sur 7 bits a existé et permettait de représenter 128 caractères ce qui était suffisant pour l'anglais.
  Par contre, l'ASCII étendu utilise 8 bits ce qui permet d'étendre à 256 le nombre de caractères. Même lorsque le nombre de bits utilisés est le même, la codification peut présenter des discordances. Le codage UNICODE sur 16 bits peut également être utilisé.
- *Conversion de syntaxe de fichier*

le cryptage : il permet de rendre illisibles les données pour les utilisateurs non autorisés. On distingue les systèmes de cryptage suivants :

- *à clé publique : cette technique utilise une règle de cryptage, appelée clé publique, et une valeur connue. La manipulation de ces éléments produit un mécanisme qui permet de décrypter les données.*
- *à clé privée : cette variante utilise seulement une clé. Seuls les composants qui détiennent la clé peuvent décrypter les données.*

12 La couche application

La couche application fournit aux applications le moyen d'accéder aux ressources du réseau, à savoir :

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

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

- notion de fichier virtuel représenté sous forme d'arbre pour les applications de transfert de fichiers, opérations permises sur un fichier, accès concurrentiels.
- Découpage des fonctions d'une application de courrier électronique qui se compose d'un contenu (en-tête et corps) et d'une enveloppe. Une fonctionnalité de l'application gère le contenu et une autre le transfert en interprétant l'enveloppe.

C'est la couche application qui vous renseigne les services disponibles pour un ordinateur sur le réseau. Par exemple, lorsque vous **double-cliquer sur l'icône Réseau** en Windows, c'est cette couche qui vous fait apparaître la liste des ordinateurs dont les services sont disponibles pour les utilisateurs du réseau.

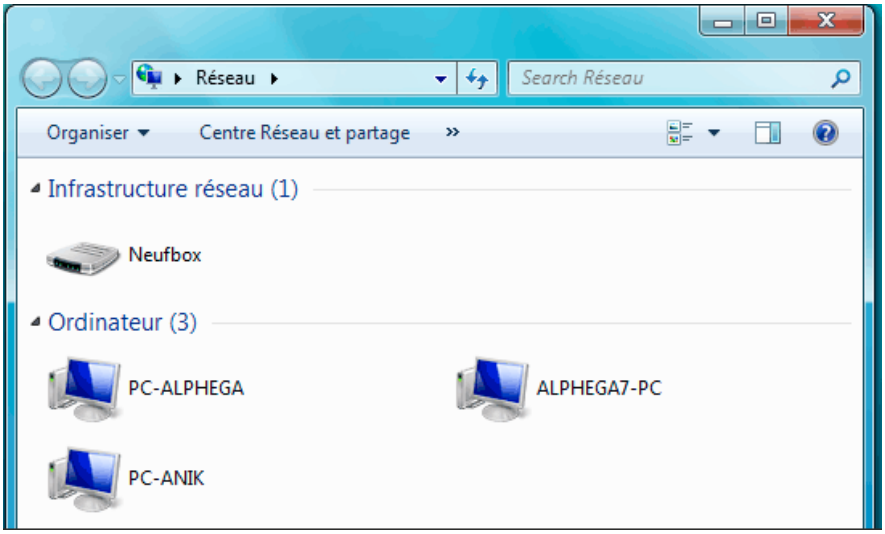


Fig. 30. Service de la couche application

Parmi les services disponibles, nous pouvons citer :

- la sécurité de Windows NT
- la gestion des fichiers et des impressions
- la messagerie SMTP (Simple Mail Transfer Protocol)

13 La pile de protocoles TCP/IP

Même si le **modèle de référence OSI est universellement reconnu**, **historiquement** et **techniquement**, la **norme OUVERTE D'INTERNET EST LA PILE DE *PROTOCOLES TCP/IP*** (*Transmission Control Protocol/Internet Protocol*). Le modèle de référence TCP/IP et la pile de protocoles TCP/IP rendent possible l'échange de données entre deux ordinateurs, **partout dans le monde**, à une vitesse quasi équivalente à celle de la lumière. Le modèle TCP/IP présente une importance historique semblable aux normes qui ont permis l'essor des industries du téléphone, de l'électricité, du chemin de fer, de la télévision et de la bande vidéo.

Le **ministère américain de la Défense a créé le modèle de référence TCP/IP** parce qu'il avait besoin d'un **réseau pouvant résister à toutes les conditions**, même à une guerre nucléaire. Imaginez en effet un monde en guerre, quadrillé de connexions de toutes sortes : fils, microondes, fibres optiques et liaisons satellites. Imaginez ensuite que vous ayez besoin de faire circuler les informations/les données (sous forme de paquets), peu importe la situation d'un nœud ou d'un réseau particulier de l'interréseau (qui pourrait avoir été détruit par la guerre). Le ministère de la Défense voulait que ses paquets se rendent chaque fois d'un point quelconque à tout autre point, peu importe les conditions. C'est ce problème de conception très épineux qui a mené à la création du modèle TCP/IP qui, depuis lors, est devenu la norme sur laquelle repose Internet.

Lors de vos lectures sur les couches du modèle TCP/IP, gardez à l'esprit le but initial d'Internet, cela vous aidera à comprendre pourquoi certaines choses sont ainsi. Le modèle TCP/IP comporte quatre couches :

- la couche application.
- la couche transport.
- la couche *Internet.*
- la couche d'accès au réseau.

Couche Application	Gère tous les aspects liés aux applications
Couche Transport	Gère les questions de qualité de service touchant la fiabilité, le contrôle de flux et la correction des erreurs.
Couche Internet	Le rôle de la couche Internet consiste à envoyer des paquets sources à partir d'un réseau quelconque de l'interréseau et à les faire parvenir à destination, indépendamment du trajet et des réseaux traversés pour y arriver.
Couche Accès au réseau	Cette couche se charge de tout ce dont un paquet IP a besoin pour établir une liaison physique, puis une autre liaison physique.

Fig. 31. Le modèle TCP/IP

Comme vous pouvez le constater, certaines couches du modèle TCP/IP portent le même nom que des couches du modèle OSI. **Il ne faut pas confondre les couches des deux modèles**, car la couche application comporte des fonctions différentes dans chaque modèle.

14 La couche Application

Les concepteurs du modèle TCP/IP estimaient que les protocoles de niveau supérieur devaient inclure les détails des couches Session et Présentation. Ils ont donc simplement créé une couche Application qui gère les protocoles de haut niveau, les questions de représentation, le code et le contrôle du dialogue. Le modèle TCP/IP regroupe en une seule couche tous les aspects liés aux applications et suppose que les données sont préparées de manière adéquate pour la couche suivante.

15 La couche Transport

La couche Transport est chargée des questions de qualité de service touchant la fiabilité, le contrôle de flux et la correction des erreurs. L'un de ses protocoles, TCP (Transmission Control Protocol - protocole de contrôle de transmission), fournit d'excellents moyens de créer, en souplesse, des communications réseau fiables, circulant bien et présentant un taux d'erreurs peu élevé. Le protocole TCP est orienté connexion. Il établit un dialogue entre l'ordinateur source et l'ordinateur de destination pendant qu'il prépare les informations de couche application en unités appelées segments.

Un protocole orienté connexion ne signifie pas qu'il existe un circuit entre les ordinateurs en communication (ce qui correspondrait à une commutation de circuits). Ce type de fonctionnement indique qu'il y a un échange de segments de couche 4 entre les deux ordinateurs hôtes afin de confirmer l'existence logique de la connexion pendant un certain temps. C'est ce que l'on appelle la commutation de paquets.

16 La couche Internet

Le rôle de la couche Internet consiste à envoyer des paquets sources à partir d'un réseau quelconque de l'interréseau et à les faire parvenir à destination, indépendamment du trajet et des réseaux traversés pour y arriver. Le protocole qui régit cette couche est appelé protocole IP (Internet Protocol). L'identification du meilleur chemin et la commutation de paquets ont lieu au niveau de cette couche. Pensez au système postal. Lorsque vous postez une lettre, vous ne savez pas comment elle arrive à destination (il existe plusieurs routes possibles), tout ce qui vous importe c'est qu'elle arrive à bon port.

17 La couche d'accès au réseau

Le nom de cette couche a un sens très large et peut parfois prêter à confusion. On lui donne également le nom de couche hôte réseau. Cette couche se charge de tout ce dont un paquet IP a besoin pour établir une liaison physique, puis une autre liaison physique. Cela comprend les détails sur les technologies LAN et WAN, ainsi que tous les détails des couches Physique et Liaison de données du modèle OSI.

Le diagramme illustré dans la figure 3.10 est appelé schéma de protocoles.

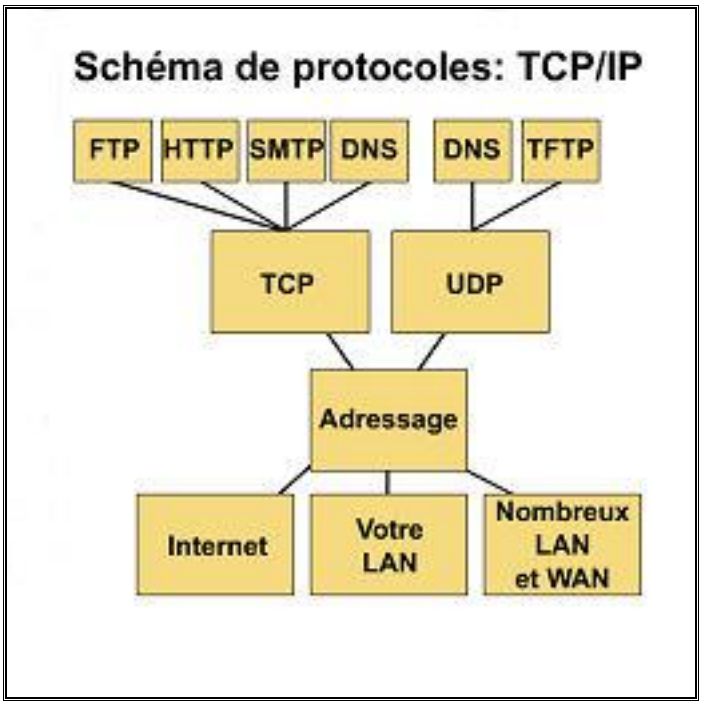


Fig. 32. Schéma de protocoles du modèle TCP/IP

Il présente certains protocoles communs spécifiés par le modèle de référence TCP/IP. Au niveau de la couche application, vous ne reconnaîtrez peut-être pas certaines tâches réseau, mais vous les utilisez probablement tous les jours en tant qu'internaute. Ces applications sont les suivantes :

- FTP - Protocole de transfert de fichiers (File Transfer Protocol).
- HTTP - Protocole HTTP (Hypertext Transfer Protocol).
- SMTP - Protocole SMTP (Simple Mail Transfer Protocol).
- DNS - Système DNS (Domain Name System).
- TFTP - Protocole TFTP (Trivial File Transfer Protocol).

Le modèle TCP/IP met l'accent sur une souplesse maximale, au niveau de la couche application, à l'intention des développeurs de logiciels. La couche transport fait appel à deux protocoles : le protocole TCP (Transmission Control Protocol) et le protocole UDP (User Datagram Protocol). La couche inférieure, soit la couche d'accès au réseau, concerne la technologie LAN ou WAN utilisée.

Dans le modèle TCP/IP, IP (Internet Protocol) est le seul et unique protocole utilisé, et ce, quels que soient le protocole de transport utilisé et l'application qui demande des services réseau. Il s'agit là d'un choix de conception délibéré. IP est un protocole universel qui permet à tout ordinateur de communiquer en tout temps et en tout lieu.

18 Comparaison des modèles OSI et TCP/IP

Similitudes	Différences
• Tous deux comportent des couches. • Tous deux comportent une couche Application, bien que chacune fournisse des services très différents. • Tous deux comportent des couches Réseau et Transport comparables. • Tous deux supposent l'utilisation de la technologie de commutation de paquets (et non de commutation de circuits).	• TCP/IP intègre la couche Présentation et la couche Session dans sa couche Application. • TCP/IP regroupe les couches Physique et Liaison de données OSI au sein d'une seule couche. • TCP/IP semble plus simple, car il comporte moins de couches.

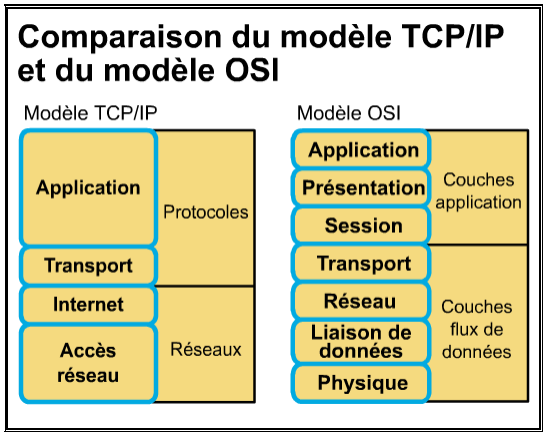


Fig. 33. Comparaison du modèle TCP/IP et du modèle OSI

Les protocoles TCP/IP constituent la norme sur laquelle s'est développé Internet. Aussi, le modèle TCP/IP a-t-il bâti sa réputation sur ses protocoles. En revanche, les réseaux ne sont généralement pas architecturés autour du protocole OSI, bien que le modèle OSI puisse être utilisé comme guide.

19 Les protocoles de la couche Application

19.1 Le DNS – Domain Name System

Pourquoi un DNS

L'adressage sur Internet est basé sur un système hiérarchique (adressage IP) qui permet de trouver un chemin entre une source et une destination donnée. Il est néanmoins très difficile de **mémoriser** des serveurs distants à partir de leur adresse IP (ex : 194.153.205.26). Celle-ci n'introduit **aucune relation entre l'adresse d'un serveur et son contenu.**

Le système de noms de domaine a été mis en place afin de pouvoir **associer le contenu d'un site avec son adresse.** On peut définir un **domaine** comme un **ensemble d'ordinateurs associés sur base de leur localisation géographique ou de leur secteur d'activités.**

Un nom de domaine est une suite de lettres ou de chiffres utilisée à la place de l'adresse numé-

rique d'un site sur Internet. Ces adresses sont appelées adresses FQDN (Fully Qualified Domain Name, "**Nom de Domaine Totalement Qualifié**") ex : [www.commentcamarche.net].

À l'heure actuelle, il existe plus de 200 domaines de haut niveau. Certains correspondent à des zones géographiques : .uk pour l'Angleterre, .fr pour la France ou .be pour la Belgique. D'autres correspondent à des secteurs d'activités : .edu pour les institutions académiques aux États-Unis, com pour les sites commerciaux, .gov pour les sites gouvernementaux américains.

Les serveurs de noms

Si les noms de domaine sont plus facilement mémorisables, ce sont toujours les adresses IP qui sont utilisées pour accéder à un serveur sur Internet. Il est donc nécessaire d'effectuer une **translation entre les noms de domaine et leur adresse IP**. C'est le serveur DNS (domain name server) qui est chargé de répondre à ces demandes de **translations**.

Aux origines de TCP/IP, étant donné que les réseaux étaient très peu étendus ou autrement dit que le nombre d'ordinateurs connectés à un même réseau était faible, les administrateurs réseau créaient des **fichiers** appelés **tables de conversion manuelle**. Ces tables étaient des fichiers séquentiels, généralement nommés hosts ou hosts.txt, associant sur chaque ligne l'adresse IP de la machine et le nom littéral associé, appelé nom d'hôte.

192.168.10.3	www.bilib.be
192.118.15.10	www.solil.fr

Ce système de tables de conversion nécessitait néanmoins **la mise à jour manuelle** des tables de tous les ordinateurs en cas d'ajout ou de modification d'un nom de machine. Ainsi, avec l'explosion de la taille des réseaux, et de leur interconnexion, il a fallu mettre en place un système de gestion des noms hiérarchisé et **plus facilement administrable**. Le système nommé **Domain Name System** (DNS), traduisez Système de nom de domaine, a été mis au point en **novembre 1983** par Paul Mockapetris (RFC 882 et RFC 883), puis révisé en 1987 dans les RFCs 1034 et 1035. Le DNS a fait l'objet depuis de nombreuses RFCs[6].

Un espace de noms hiérarchiques permettant de garantir **l'unicité d'un nom** dans une structure arborescente, à la manière des systèmes de fichiers d'Unix.

Ce système dispose aussi de clients permettant de « **résoudre** » les noms de domaines, c'est-à-dire **interroger** les serveurs afin de connaître l'adresse IP correspondant à un nom.

Système hiérarchique

Le système DNS est basé sur une hiérarchie qui crée différents niveaux de serveurs DNS.

Si un serveur DNS local appelé **serveur primaire** (primary domain name server) est capable d'effectuer la translation d'un nom de domaine vers l'adresse IP correspondante, il effectue le travail et renvoie le résultat au client. S'il ne peut traduire le nom de domaine, il **transmet** la demande à un serveur DNS situé plus haut dans la hiérarchie, un **serveur de nom secondaire** (secondary domaine name server).

Chaque serveur de nom **est déclaré dans un serveur de nom de domaine de niveau immédiatement supérieur**, ce qui permet implicitement une délégation d'autorité sur les domaines. Le système de nom est une **architecture distribuée**, où chaque entité est responsable de la gestion de son nom de domaine. Il n'existe donc pas d'organisme ayant à charge la gestion de l'ensemble des noms de domaines.

[6] Les requests for comments (RFC), littéralement « demande de commentaires », sont une série numérotée de documents officiels décrivant les aspects techniques d'Internet, ou de différent matériel informatique (routeurs, serveur DHCP). Peu de RFC sont des standards, mais tous les standards Internet publiés par l'IETF sont des RFC.

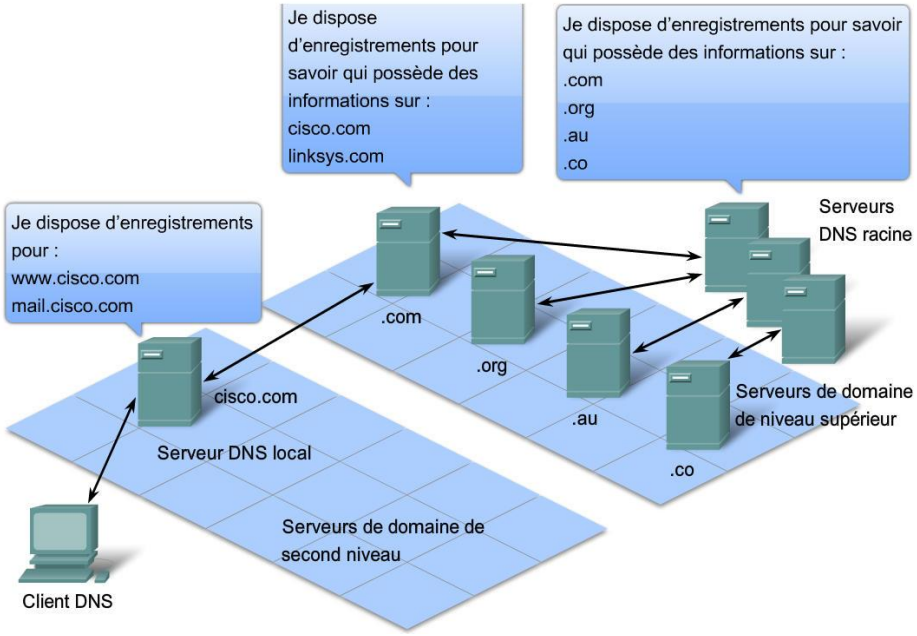


Fig. 34. Structure hiérarchique des serveurs DNS

La séquence de résolution DNS est illustrée aux figures suivantes. Dans un premier temps, le client d'adresse IP 198.150.11.40 effectue une requête dans le navigateur vers le site www.cisco.com.

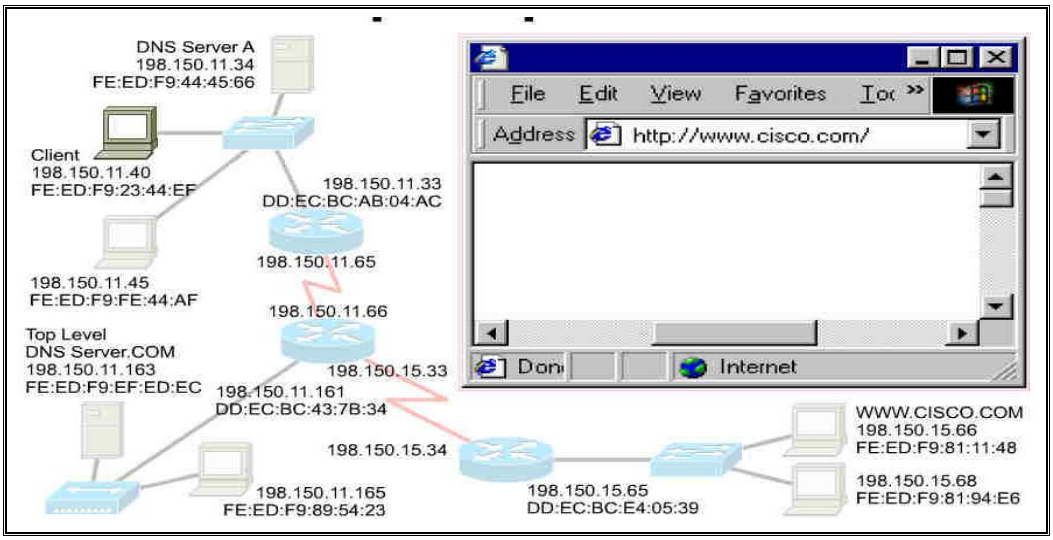


Fig. 35. Le client introduit l'URL du site souhaité

Le **navigateur** va construire un paquet à envoyer vers le serveur DNS dont l'adresse IP figure dans la configuration TCP/IP du client. La MAC adresse de ce serveur figure déjà probablement dans le cache ARP de la machine sinon le client envoie une requête ARP pour obtenir l'adresse MAC du serveur.

Les serveurs correspondant aux domaines de plus haut niveau (TLD) sont appelés « serveurs de noms racine ». Il en existe **treize**, répartis sur la planète, possédant les noms « a.root-servers.net » à « m.root-servers.net ».

Un serveur de noms définit une zone, c'est-à-dire un ensemble de domaines sur lequel le serveur a autorité. Le système de noms de domaine est transparent pour l'utilisateur, néanmoins il ne faut pas oublier les points suivants :

- Chaque ordinateur doit être configuré avec l'adresse d'une machine capable de transformer n'importe quel nom en une adresse IP. Cette machine est appelée Domain Name Server. Pas de panique: lorsque vous vous connectez à internet, le fournisseur d'accès va automatiquement modifier vos paramètres réseau pour vous mettre à disposition ces serveurs de noms.
- L'adresse IP d'un second Domain Name Server (secondary Domain Name Server) doit également être définie : le serveur de noms secondaire peut relayer le serveur de noms primaire en cas de dysfonctionnement.

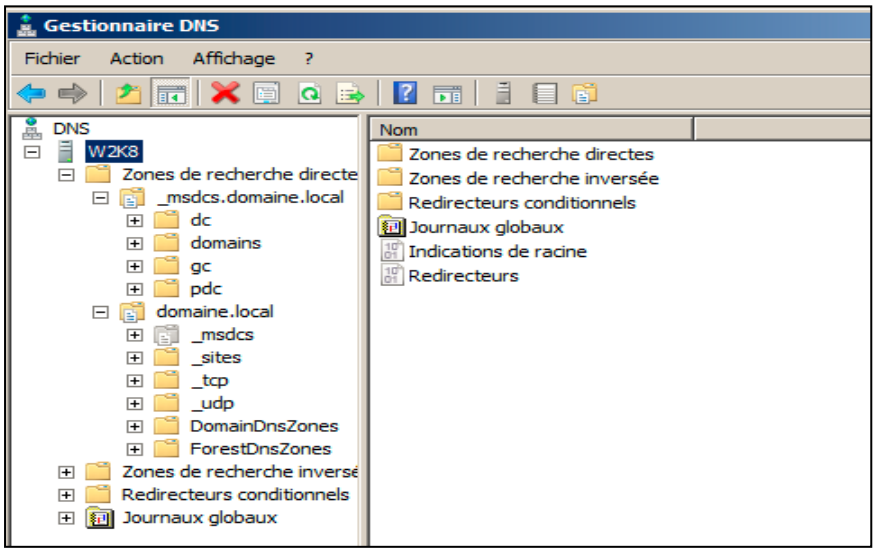


Fig. 36. Gestionnaire de DNS sur un serveur 2008

Types d'enregistrements

Un DNS est une base de données répartie contenant des enregistrements, appelés **RR** (Resource Records), concernant les noms de domaines. Seules sont concernées par la lecture des informations ci-dessous les personnes responsables de l'administration d'un domaine, le fonctionnement des serveurs de noms étant totalement transparent pour les utilisateurs.

En raison du système de cache permettant au système DNS d'être réparti, les enregistrements de chaque domaine possèdent une durée de vie, appelée TTL (Time To Live, traduisez espérance de vie), permettant aux serveurs intermédiaires de connaître la date de péremption des informations et ainsi savoir s'il est nécessaire ou non de la revérifier.

D'une manière générale, un enregistrement DNS comporte les informations suivantes :

Nom de domaine (FQDN)	TTL	Type	Classe	RData
www.commentcamarche.net.	3600	A	IN	163.5.255.85

- **Nom de domaine** : le nom de domaine doit être un nom FQDN, c'est-à-dire être terminé par un point. Si le point est omis, le nom de domaine est relatif, c'est-à-dire que le nom de domaine principal suffixera le domaine saisi ;
- **Type** : une valeur sur 16 bits spécifiant le type de ressource décrit par l'enregistrement. Le type de ressource peut être un des suivants :
  - A : il s'agit du type de base établissant la correspondance entre un nom canonique et une adresse IP. Par ailleurs il peut exister plusieurs enregistrements A, correspondant aux différentes machines du réseau (serveurs).

- CNAME (Canonical Name) : il permet de faire correspondre un alias au nom canonique. Il est particulièrement utile pour fournir des noms alternatifs correspondant aux différents services d'une même machine.
- HINFO : il s'agit d'un champ uniquement descriptif permettant de décrire notamment le matériel (CPU) et le système d'exploitation (OS) d'un hôte. Il est généralement conseillé de ne pas le renseigner afin de ne pas fournir d'éléments d'informations pouvant se révéler utiles pour des pirates informatiques.
- MX (Mail eXchange) : correspond au serveur de gestion du courrier. Lorsqu'un utilisateur envoie un courrier électronique à une adresse (utilisateur@domaine), le serveur de courrier sortant interroge le serveur de nom ayant autorité sur le domaine afin d'obtenir l'enregistrement MX. Il peut exister plusieurs MX par domaine, afin de fournir une redondance en cas de panne du serveur de messagerie principal. Ainsi l'enregistrement MX permet de définir une priorité avec une valeur pouvant aller de 0 à 65 535 :
- NS : correspond au serveur de noms ayant autorité sur le domaine.
- PTR : un pointeur vers une autre partie de l'espace de noms de domaines.
- SOA (Start Of Authority) : le champ SOA permet de décrire le serveur de nom ayant autorité sur la zone, ainsi que l'adresse électronique du contact technique (dont le caractère « @ » est remplacé par un point).

- **Classe** : la classe peut être soit IN (correspondant aux protocoles d'internet, il s'agit donc du système utilisé dans notre cas), soit CH (pour le système chaotique) ;
- **RDATA** : il s'agit des données correspondant à l'enregistrement. Voici les informations attendues selon le type d'enregistrement :
  - A : une adresse IP sur 32 bits ;
  - CNAME : un nom de domaine ;
  - MX : une valeur de priorité sur 16 bits, suivi d'un nom d'hôte ;
  - NS : un nom d'hôte ;
  - PTR : un nom de domaine ;
  - SOA : plusieurs champs.

Domaines de haut niveau

Il existe deux catégories de TLD (Top Level Domain, soit domaines de plus haut niveau) :
Les domaines dits « **génériques** », appelés gTLD (generic TLD). Les gTLD sont des noms de domaines génériques de niveau supérieur proposant une classification selon le secteur d'activité. Ainsi chaque gTLD possède ses propres règles d'accès :

gTLD historiques :

.arpa correspond aux machines issues du réseau originel ;

.com correspondait initialement aux entreprises à vocation commerciale. Désormais ce TLD est devenu le « TLD par défaut » et l'acquisition de domaines possédant cette extension est possible, y compris par des particuliers.

.edu correspond aux organismes éducatifs ;

.gov correspond aux organismes gouvernementaux ;

.int correspond aux organisations internationales ;

.mil correspond aux organismes militaires ;

.net correspondait initialement aux organismes ayant trait aux réseaux. Ce TLD est devenu depuis quelques années un TLD courant. L'acquisition de domaines possédant cette extension est possible, y compris par des particuliers.

.org correspond habituellement aux entreprises à but non lucratif.

nouveaux gTLD introduits en novembre 2000 par l'ICANN :

.aero correspond à l'industrie aéronautique ;

.biz (business) correspondant aux entreprises commerciales ;

.museum correspond aux musées ;
.name correspond aux noms de personnes ou aux noms de personnages imaginaires ;
.info correspond aux organisations ayant trait à l'information ;
.coop correspondant aux coopératives ;
.pro correspondant aux professions libérales.

Il est plus simple de naviguer sur Internet en utilisant des noms de domaine plutôt qu'avec des suites de chiffres. A chaque fois que vous appelez une page, que des images sont chargées ou que vous téléchargez des fichiers, un serveur DNS est utilisé.

Votre fournisseur d'accès à Internet dispose de ses propres serveurs DNS qui communiquent avec tous les autres serveurs DNS du monde afin de tenir une liste des correspondances noms de domaine / adresses IP.

Les serveurs DNS utilisés par les fournisseurs d'accès à Internet ne sont pas les plus rapides. Certains services mettent gratuitement à votre disposition des serveurs DNS très véloces. Vous pouvez donc accélérer votre navigation sur Internet en utilisant ces serveurs DNS plutôt que ceux de votre fournisseur d'accès à Internet.

Avec le logiciel **gratuit** Namebench, **vous allez pouvoir tester la vitesse des serveurs DNS de votre fournisseur d'accès à Internet ainsi que celle de nombreux serveurs DNS publics et gratuits depuis chez vous**, même s'ils appartiennent à un autre fournisseur d'accès à Internet. Vous pourrez alors **choisir et utiliser les serveurs DNS les plus performants** en modifiant simplement les paramètres de votre connexion réseau sur votre ordinateur.

Description de l'attaque d'Anonymous.
http://securityaffairs.co/wordpress/3184/cyber-crime/anonymous-dns-amplification-attacks-for-operation-global-blackout.html
http://www.isotf.org/news/DNS-Amplification-Attacks.pdf

19.2 La messagerie électronique

Un client de messagerie permet l'envoi des messages (via le protocole SMTP - Simple Mail Transfer Protocol sur le port 25) et récupère les messages reçus dans la boîte aux lettres du client (via le protocole POP - Post Office Protocol sur le port 110).

Le format de message du protocole SMTP utilise un ensemble rigide de commandes et de réponses. Ces commandes prennent en charge les procédures du protocole SMTP, telles que l'ouverture de session, le transfert du courriel, la vérification des noms de boîte aux lettres, le développement des listes de diffusion et les échanges de début et de fin.

Le serveur de messagerie utilise deux processus distincts :

- Agent de transfert des messages (MTA - Mail Transfer Agent).
- Agent de remise des messages (MDA - Mail Delivery Agent).

L'application de messagerie (MUA - Mail User Agent) transfère le message vers le serveur de messagerie configuré. L'agent de transfert de message (MTA) reçoit le message et vérifie si le destinataire est situé sur ce serveur. Si c'est le cas, le mail est transmis à l'agent de remise des messages (MDA), qui le place dans le mailbox du destinataire. L'agent de remise procède à une écoute pour détecter le moment où le client se connecte au serveur. Une fois la connexion établie, le serveur peut remettre le courriel au client.

Si ce n'est pas le cas, le MTA transfère le mail vers le MTA du Serveur de Mail de destination.

Le MDA peut prendre en charge : une vérification de virus, le filtrage des spams et la gestion des accusés de réception.

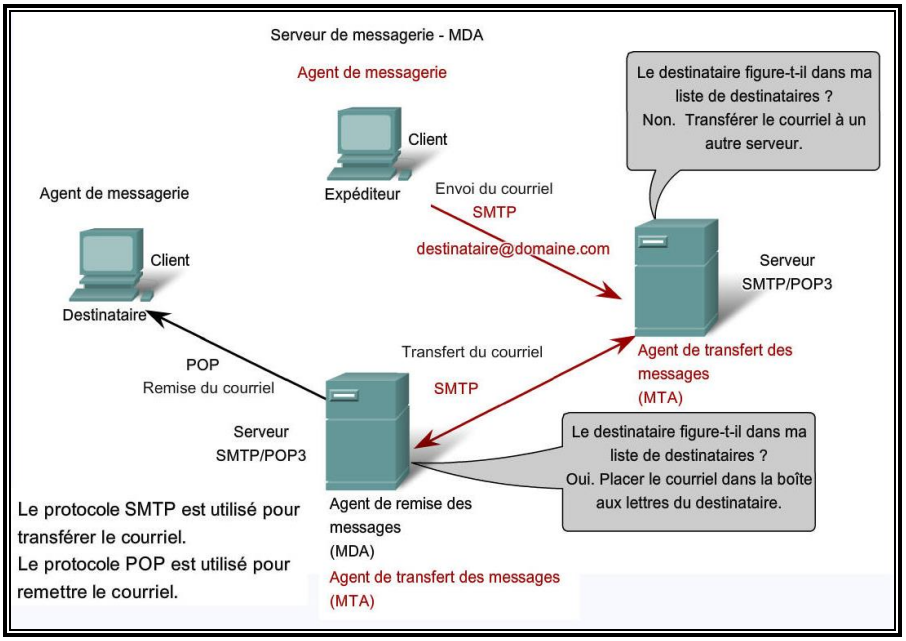


Fig. 37. Mécanisme de fonctionnement d'un système de messagerie électronique

Le client de messagerie, interne à une entreprise, peut être connecté à des applications disposant de leur propre format de messagerie interne et utilise des protocoles propriétaires pour communiquer avec le serveur (Lotus Note, Groupwise de Novell ou Exchange). Ces produits disposent d'une passerelle assurant le reformatage des messages permettant de communiquer en dehors de l'entreprise.

Il est également possible de se connecter à un service de messagerie sur un navigateur sans avoir besoin d'un agent de messagerie.

Lorsqu'on regarde la structure de l'adresse e-mail utilisée dans un client de messagerie, on constate que celle-ci est constituée de deux parties :

- La première partie correspond à l'identifiant du destinataire
- La deuxième spécifie l'adresse du "bureau de poste" de destination

Le client e-mail va commencer par faire appel à un serveur DNS afin d'obtenir l'adresse IP correspondant au nom de domaine du serveur de destination. Le client va pouvoir encapsuler les données et les transmettre au serveur de destination.

Lorsque le serveur de destination reçoit le mail, il va extraire le nom du destinataire et vérifier qu'il dispose bien d'un compte sur le serveur Post Office. Si c'est le cas, le mail sera stocké dans la boîte de réception du destinataire jusqu'à ce que celui-ci vienne le relever. S'il n'y a pas de compte correspondant au destinataire, le serveur va générer un message d'erreur à destination du serveur Post Office de l'expéditeur.

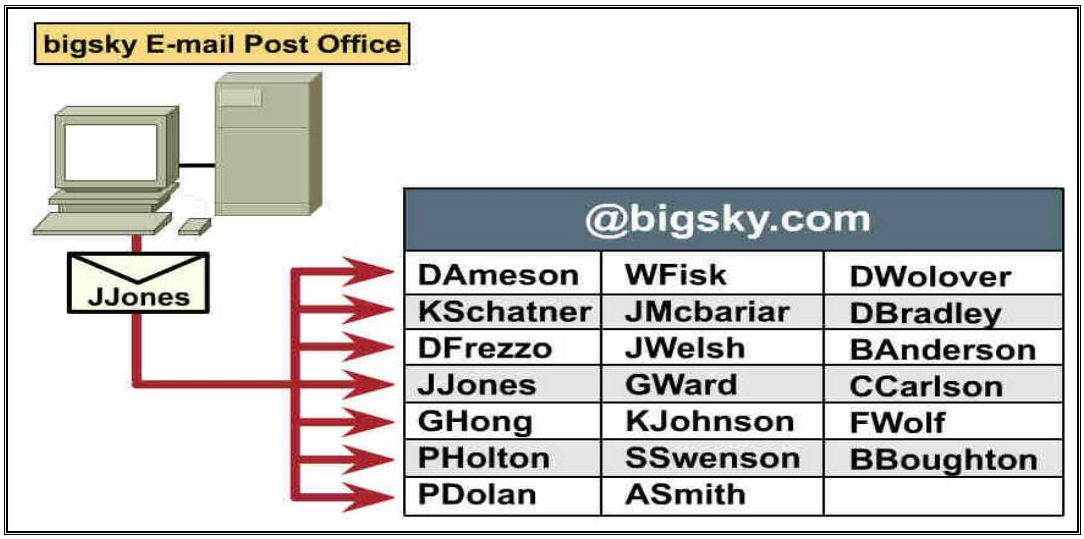


Fig. 38. L'existence du compte destinataire est vérifiée sur le serveur Post Office

La phase suivante sera effectuée par le destinataire qui se connecte sur le serveur Post Office. Lors de cette connexion, le destinataire sera la plupart du temps invité à introduire un identifiant et un mot de passe. Lorsque ceux-ci ont été introduits ou récupérés dans les paramètres de configuration du client de messagerie, celui-ci va générer une requête à destination du serveur Post Office (qui est également configuré dans les paramètres du client de messagerie).

Lorsque le serveur Post Office reçoit la requête, il vérifie l'exactitude des paramètres d'identification du destinataire. Si ceux-ci sont corrects, le serveur transmet les fichiers en attente vers la machine du destinataire.

19.3 L'émulation de terminal (Telnet)

Un programme d'émulation de terminal (Telnet) permet de se connecter à distance sur une autre machine. Il est même possible de se connecter sur une machine à travers l'Internet. Lorsqu'on a effectué la procédure de connexion (login) sur la machine distante, il devient possible d'exécuter des commandes sur celle-ci. Le client Telnet est la machine locale alors que serveur Telnet est une machine distante sur laquelle est exécuté un service particulier (parfois appelé Daemon).

Pour établir la connexion depuis le client, il est nécessaire de spécifier un certain nombre de paramètres : l'adresse IP de la machine distante ou son nom DNS ainsi que le type d'émulation que l'on souhaite (vt100).

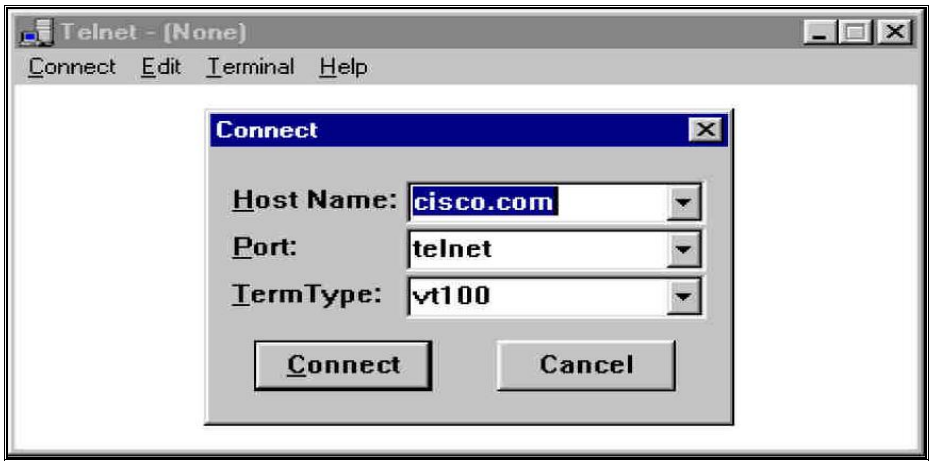


Fig. 39. Ouverture d'une session Telnet

Il faut noter que les opérations effectuées sous Telnet ne consomment aucune ressource sur la machine locale. Celle-ci se contente de transmettre les frappes clavier à la machine distante. Les instructions sont exécutées sur celle-ci en utilisant ses ressources processeur, mémoire et disque. La machine locale récupère en retour les écrans d'affichage de l'application.

Une session Telnet est établie sur base des protocoles TCP/IP et d'une série de facilités connues sous le nom de "*Network Virtual Terminal* (NVT). L'application Telnet travaille essentiellement au niveau des trois couches hautes du modèle OSI : la couche application (pour les commandes), la couche présentation (pour le format, généralement de l'ASCII) et la couche session (pour la transmission).

Les données vont continuer à traverser les couches suivantes du modèle OSI afin d'être transmises sur le média. Lorsqu'elles sont reçues par la machine distante, elles vont remonter à travers le modèle OSI jusqu'au service Telnet exécuté sur celle-ci.

Les commandes sont exécutées sur la machine distante et les résultats renvoyés au client en suivant le même processus.

Ce processus "envoi commandes – réception des résultats" sera répété jusqu'à ce que le client local ait terminé son travail. Lorsque celui-ci est fini, c'est le client qui met fin à la session.

19.4 Le transfert de fichiers (FTP)

Le protocole FTP (File Transfer Protocol) est indiqué pour downloader ou uploader des fichiers. FTP est une application de type client/serveur. Il faut donc qu'une application serveur soit exécutée sur la machine à laquelle on souhaite accéder.

Il utilise le port TCP 21 pour les commandes et le port TCP 20 pour le transfert de fichier.

Une session FTP est établie comme une session Telnet, elle est maintenue jusqu'à ce que le client y mette fin ou qu'une erreur de transmission se produise. Lorsqu'on ouvre une session FTP, on peut être invité à introduire un login et un mot de passe. Il est néanmoins possible de se connecter comme "anonymous".

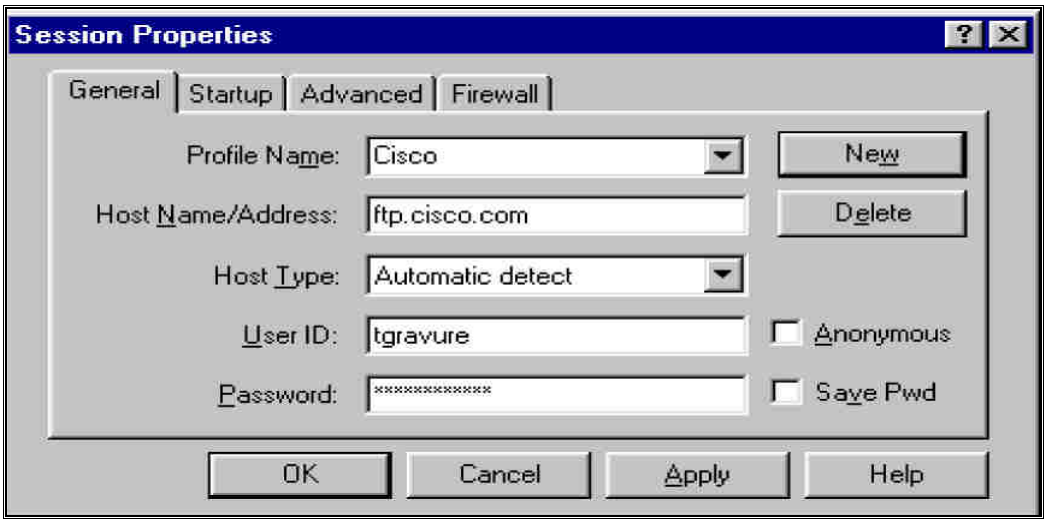


Fig. 40. Paramétrage du client FTP

Lorsqu'on effectue son login, une liaison de commande s'ouvre entre la machine cliente et le serveur FTP. Cette liaison permet d'échanger des instructions entre les deux machines. Cela permet au client de créer des dossiers, de changer de répertoire, de supprimer ou de renommer des fichiers. Il est possible d'effectuer sur la machine distante les mêmes commandes de gestion de fichiers que sur la machine locale.

Lorsqu'on souhaite échanger des fichiers entre la machine cliente et le serveur FTP, une deuxième connexion est établie. Il s'agit de la liaison de données qui va être utilisée pour le transfert. Celui-ci peut avoir lieu en mode ASCII ou en mode binaire. Ces deux modes déterminent comment le fichier de données va être transféré entre les deux machines.

Lorsque le transfert est terminé, la connexion de données est automatiquement fermée. Si le travail est terminé, vous pouvez vous déconnecter. C'est cette déconnexion qui va mettre fin à la liaison de commande et à la session.

19.5 HTTP (Hypertext Transfer Protocol)

Le protocole HTTP travaille avec le World Wide Web qui constitue la partie d'Internet qui connaît le plus haut taux de croissance. Cette croissance s'explique par la grande facilité d'accès à l'information que le Web permet.

Le navigateur Web est une application de type client/serveur. Cela signifie qu'il nécessite une composante client et serveur pour être fonctionnel.

Un navigateur Web présente les données sous un format multimédia combinant du texte, des graphiques, du son et de la vidéo. Les pages Web sont créées à partir d'un langage de description appelé HTML (*Hypertext Markup Language*). Le langage HTML permet au navigateur de présenter l'apparence de la page d'une certaine manière. HTML précise également l'emplacement du texte, des fichiers et des objets qui vont être transférés du serveur Web vers le navigateur Web.

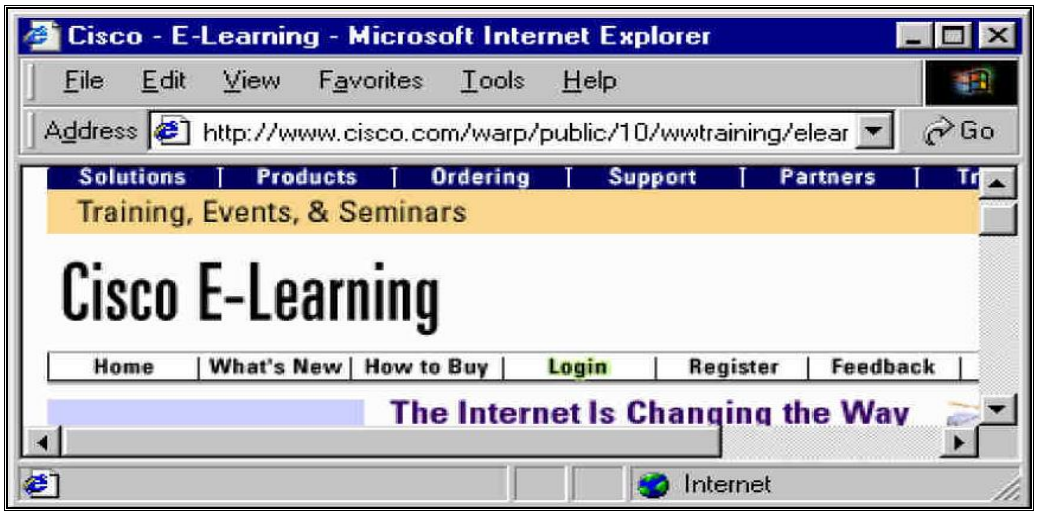


Fig. 41. Apparence d'une page Web classique

L'utilisation de liens hypertexte dans la page Web rend la navigation sur le World Wide Web très facile. Un lien hypertexte est un objet (mot, phrase ou icône) inséré sur une page web et qui assure le transfert vers une autre page Web.

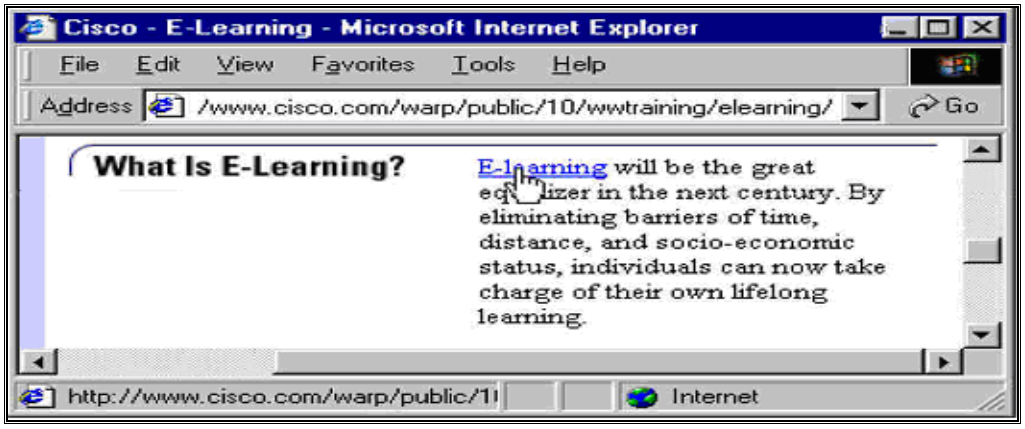


Fig. 42. Lien hypertexte inséré dans une page Web

Si l'on examine la structure d'une l'URL, on constate que celle-ci est constituée de plusieurs entités qui chacune a une signification particulière.

http://	**www.**	**Cisco.com**	**/edu/**
Permet d'identifier le protocole qui doit être utilisé par le navigateur.	Identifie le type de site qui va être contacté par le navigateur. Il s'agit ici d'un serveur fournissant des ressources Web.	Représente le nom de domaine du site. C'est à partir de cette information qu'on obtiendra son adresse IP.	Identifie le dossier où est située la page Web sur le serveur. En cas d'absence d'information à ce niveau, le navigateur chargera la page par défaut identifiée par le serveur.

Lorsqu'on lance le navigateur Web, celui-ci s'ouvre sur la page d'accueil spécifiée au niveau de ses paramètres.

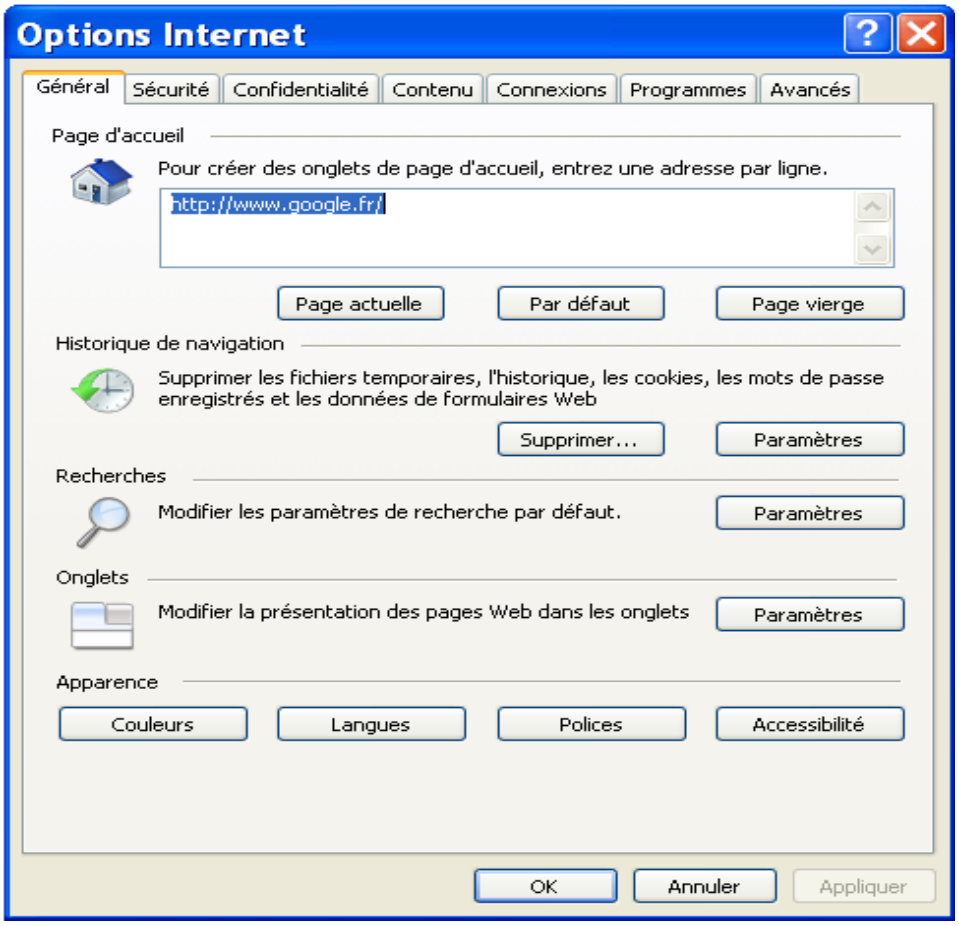


Fig. 43. Paramétrage de la page d'accueil dans le navigateur

À partir de cette page, nous pouvons cliquer sur un des liens hypertextes ou introduire une autre URL dans la barre d'adresse du navigateur. Le navigateur Web examine le protocole auquel on fait appel pour savoir si un autre programme doit être ouvert. Il détermine également l'adresse IP du serveur Web de destination.

Les couches transport, réseau, liaison de données et physique initialisent une session avec le serveur Web. Les informations transmises au serveur HTTP fournissent le nom du dossier où est située la page Web. En l'absence d'information plus spécifique, c'est la page par défaut du serveur qui sera utilisée (celle-ci est spécifiée dans les paramètres de configuration du serveur).

Le serveur va répondre à la demande en envoyant le texte, les données audio et vidéo ainsi que les éléments graphiques spécifiés dans le code HTML au client. Le navigateur sur le poste client va réassembler tous les éléments afin de fournir une visualisation de la page Web. Il met ensuite fin à la session.

Si l'utilisateur clique sur une autre page située sur le même serveur ou sur un serveur différent, l'ensemble du processus est réexécuté.

20 Les protocoles de couche transport du modèle TCP/IP

Les deux principaux rôles de la couche transport sont le contrôle de flux et la fiabilité. Elle définit une connectivité de bout en bout entre les applications hôtes.

Voici quelques services de transport de base :

- Segmentation des données d'application de couche supérieure.
- Établissement d'une connexion de bout en bout.
- Transport des segments d'un hôte d'extrémité à un autre.
- Contrôle du flux assuré par les fenêtres glissantes.
- Fiabilité assurée par les numéros de séquence et les accusés de réception.

La couche Transport fournit une connexion logique entre les hôtes source et de destination. Les protocoles de transport segmentent et rassemblent les données envoyées par des applications de couche supérieure en un même flux de données, ou connexion logique, entre les deux points d'extrémité.

20.1 Établissement, maintenance et fermeture de la session

Les applications envoient des segments de données suivant la méthode du premier arrivé, premier servi. Ils peuvent être acheminés vers une même destination ou vers des destinations différentes. Dans le modèle de référence OSI, plusieurs applications peuvent partager la même connexion de transport. On parle alors de multiplexage des conversations de couche supérieure. Plusieurs conversations simultanées de couche supérieure peuvent être multiplexées sur une seule connexion.

Un des rôles de la couche Transport est d'établir une session orientée connexion entre des unités identiques de la couche application. Pour que le transfert de données puisse débuter, les applications source et de destination doivent informer leur système d'exploitation qu'une connexion va être initiée. Un des nœuds initie la connexion qui doit être acceptée par l'autre. Les modules logiciels de protocole des deux systèmes d'exploitation communiquent en envoyant des messages sur le réseau pour vérifier si le transfert est autorisé et si les deux ordinateurs sont prêts.

La connexion est établie et le transfert des données peut commencer après synchronisation. Pendant le transfert, les deux ordinateurs continuent de communiquer au moyen de leur logiciel de protocole pour vérifier si les données sont bien reçues.

La figure suivante montre une connexion type établie entre deux systèmes. La première étape du protocole d'échange bidirectionnel demande la synchronisation. Le deuxième échange accuse réception de la demande de synchronisation initiale et synchronise les paramètres de connexion dans la direction opposée. Le troisième segment d'échange est un accusé de réception indiquant à la destination que la connexion peut être établie des deux côtés. Une fois la connexion établie, le transfert des données commence.

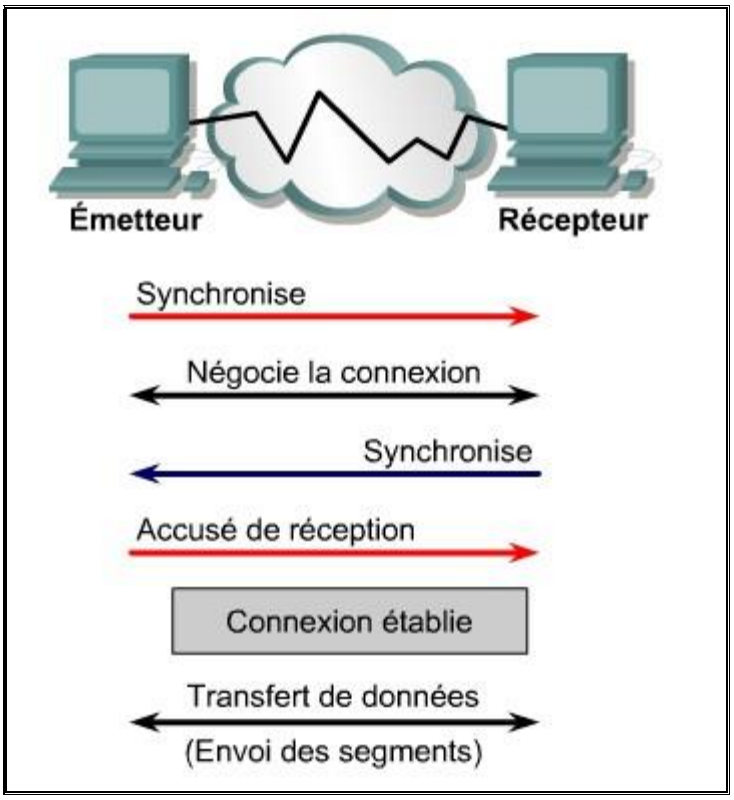


Fig. 44. Ouverture d'une session entre deux machines

Internet est souvent représenté par un nuage. La couche transport envoie des paquets de données d'une source à une destination à travers ce nuage. Le rôle principal de la couche transport est d'assurer une fiabilité et un contrôle de bout en bout lors du transfert des données à travers ce nuage. Les fenêtres glissantes, les numéros de séquençage et les accusés de réception permettent d'obtenir ce résultat. La couche Transport définit également une connectivité de bout en bout entre les applications hôtes.

TABLE DES MATIERES

TABLE DES ILLUSTRATIONS