I.U.T. BORDEAUX 1
DEPARTEMENT INFORMATIQUE

# TELEINFORMATIQUE

## *LIVRET III*

**- Septembre 2005 -**

**SOMMAIRE**

# 6. Le modèle OSI

## 6.1 Objectif de la normalisation OSI

Les premières architectures de réseaux proposées par les constructeurs (SNA System Network Architecture chez IBM, DSA Distributed System Architecture chez Bull...) dépendaient du matériel (topologie du réseau, type de connexion) et du logiciel (système d'exploitation, application) utilisés et n'étaient naturellement pas compatibles entre elles. Afin de permettre la constitution de réseaux téléinformatiques à partir d'interconnexion de systèmes hétérogènes, l'organisme de normalisation ISO (International Standard Organization) propose une définition de norme pour les architectures de réseaux. Cette norme, appelée OSI (Open System Interconnection) permet d'interconnecter les matériels des constructeurs qui l'adoptent et de réaliser des applications sur des matériels hétérogènes. La norme OSI définit des protocoles normalisés pour l'échange d'informations entre systèmes (ordinateurs, terminaux, nœuds du réseau...).

## 6.2 Concepts de base du modèle OSI

### 6.2.1 Découpage en couches

Pour réaliser un système complexe, on le décompose en éléments plus simples et donc plus facilement réalisables. Pour s'en convaincre, il n'est qu'à imaginer la complexité d'un programme qui devrait gérer :

- le dialogue entre processus,
- le choix d'un chemin pour l'acheminement de l'information,
- l'utilisation d'un réseau,
- les procédures,
- les coupleurs,
- les reprises en cas d'erreur,
- etc.

L'architecture peut être fonctionnellement découpée en plusieurs couches, chaque couche réalisant un niveau d'abstraction (par exemple, on a vu comment transmettre physiquement des données binaires sur une voie : ceci peut constituer une couche dite physique d'une architecture). Cette norme n'implique aucune implémentation particulière, mais spécifie seulement un découpage en couches fonctionnelles de telle sorte que chaque couche ne fasse pas l'objet d'un degré de complexité élevé. Pour effectuer ce découpage, l'organisme ISO s'est appuyé sur les principes suivants :

- Créer des couches distinctes pour traiter les fonctions qui sont manifestement différentes sur le plan du traitement exécuté ou de la technologie mise en œuvre,
- Créer une couche lorsque le traitement se fait à un niveau d'abstraction différent (structure, syntaxe, sémantique),
- Permettre des changements dans une couche sans affecter les autres couches,
- Organiser et regrouper les fonctions à l'intérieur d'une couche en sous-couche.

Résultat : un système à 7 couches.

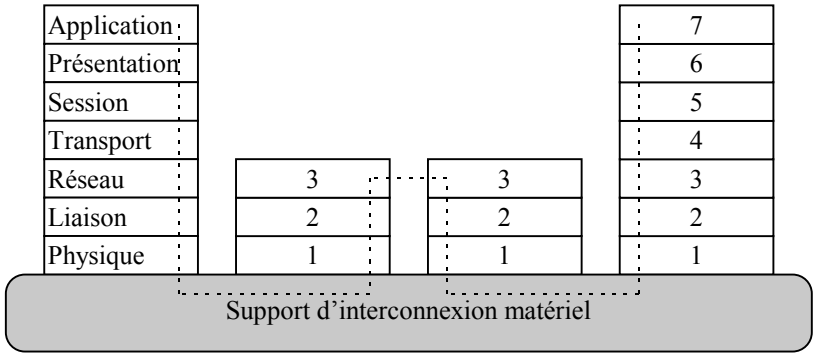



*Figure 6-1 Les 7 couches du modèle OSI*

### 6.2.2 Les 7 couches du modèle OSI et leurs fonctions

Les couches basses 1, 2 et 3 (appelées aussi couches de communication) permettent le transfert à travers le réseau des informations provenant des couches hautes 4, 5, 6 et 7 (appelées aussi couches de traitement).
**Couche physique** : elle décrit les moyens physiques, interfaces mécaniques et électriques permettant de transmettre les données brutes (séquence de bits).
**Couche liaison de données** : elle permet le transfert sans erreur d'informations entre systèmes adjacents, c'est à dire directement connectés.
**Couche réseau :** elle permet le routage et l'acheminement des informations au travers de réseaux pouvant comprendre 1 ou plusieurs systèmes intermédiaires.
**Couche transport** : elle permet le contrôle de bout en bout du transport de l'information entre 2 systèmes terminaux.
**Couche session** : elle définit l'organisation des échanges et la structuration du dialogue entre applications.
**Couche présentation** : elle définit la représentation sur le plan de la syntaxe des informations échangées.
**Couche application** : elle définit des mécanismes communs aux processus des utilisateurs et contient la "signification" (sémantique) des informations échangées.

### 6.2.3 Terminologie

Le système est découpé en couches superposées, chacune d'elles étant chargée d'offrir un service à la couche supérieure. On appelle *interface* l'ensemble des règles qui définissent le dialogue entre 2 couches adjacentes. On appelle *protocole* l'ensemble des règles qui définissent le dialogue entre 2 couches de même niveau.
Une *entité* est un élément actif d'une couche : une entité d'un site (site 1) de la couche N demande des services à la couche immédiatement inférieure appelée couche N-1, dialogue avec son entité homologue d'un autre site (site 2) pour fournir des services à la couche immédiatement supérieure, appelée couche N+1.

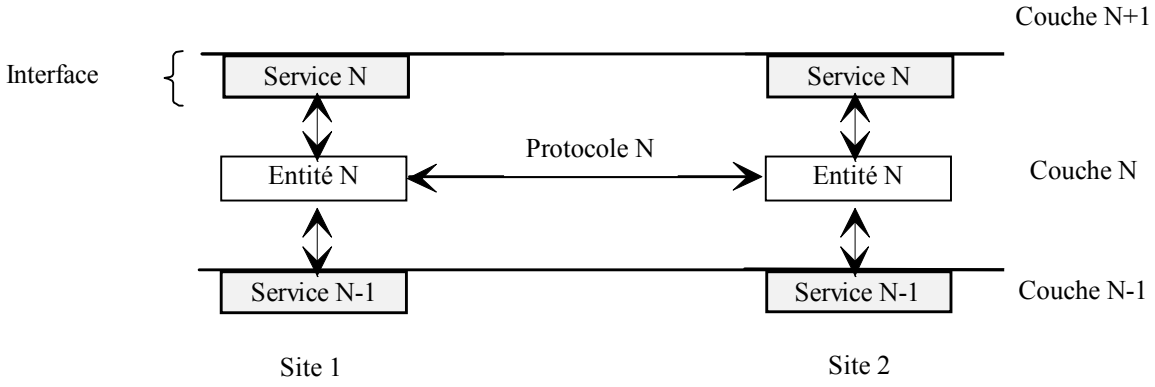



*Figure 6-2 Couches, services et protocoles.*

### 6.2.4 Modes connecté et non connecté

En mode connecté, avant tout dialogue, on établit une connexion entre 2 couches de même niveau. Une dialogue se déroule suivant le schéma :

- Phase de connexion
- Echange
- Libération de la connexion

En mode non connecté, on émet des messages sans s'assurer au préalable que le correspondant est à l'écoute. Les protocoles OSI utilisent le mode connecté.

### 6.2.5 Services et primitives

Une couche peut fournir, par exemple, des services de connexion (CONNECT), d'échange de données (DATA) ou de déconnexion (DISCONNECT).
Pour utiliser un service de la couche N-1, une entité de la couche N dispose de primitives pour formuler ses demandes de services, pour être informée d'un événement ou pour connaître la suite donnée à une demande de service :

- Requête (REQUEST) : une entité sollicite un service pour exécuter une activité,
- Confirmation (CONFIRM) : une entité est informée de sa demande de service,
- Indication (INDICATION) : une entité est informée d'un événement,

- Réponse (RESPONSE) : Une entité répond à un événement.

On peut imaginer un service à connexion simple avec les 8 primitives suivantes :

- CONNECT REQUEST : Demande d'établissement d'une connexion,
- CONNECT INDICATION : signalisation à la partie appelée,
- CONNECT RESPONSE : utilisée par la partie appelée pour accepter ou rejeter l'appel,
- CONNECT CONFIRM : Indique à l'appelant si l'appel est accepté,
- DATA REQUEST : demande d'envoi de données,
- DATA INDICATION : signale l'arrivée de données,
- DISCONNECT REQUEST : demande de relâchement de la communication,
- DISCONNECT INDICATION : signale la demande de fin de communication.

Prenons l'exemple classique du téléphone qui fournit à l'usager un service de dialogue en mode connecté. Les primitives ci-dessus peuvent être illustrées par le scénario suivant.
Vous invitez votre Mamie Nova pour le goûter :

1 CONNECT REQUEST : faire le numéro de Mamie Nova,
2 CONNECT INDICATION : ça sonne chez elle,
3 CONNECT RESPONSE : elle décroche son téléphone,
4 CONNECT CONFIRM : Vous entendez l'arrêt de la sonnerie,
5 DATA REQUEST : Vous l'invitez pour le goûter,
6 DATA INDICATION : Elle entend votre invitation,
7 DATA REQUEST : Elle dit qu'elle serait ravie de venir,
8 DATA INDICATION : Vous entendez qu'elle accepte,
9 DISCONNECT REQUEST : Vous raccrochez,
10 DISCONNECT INDICATION : Elle l'entend et raccroche également.

### 6.2.6 Echange d'informations entre entité

L'échange d'informations entre couches adjacentes se fait par l'intermédiaire des primitives de services en utilisant des unités de données de service ou SDU (Service Data Unit).
Une entité de la couche N dialogue avec son homologue sur un site adjacent en s'échangeant des unités de données de protocole ou PDU (Protocol Data Unit).
Pour gérer le protocole au niveau N, des informations de gestion du protocole ou PCI (Protocol Control Information) sont rajoutées au SDU pour constituer des PDU.
Pour s'y retrouver, on préfixe PDU, SDU et PCI par l'identification de la couche concernée. Ces principes sont illustrés dans la figure suivante.

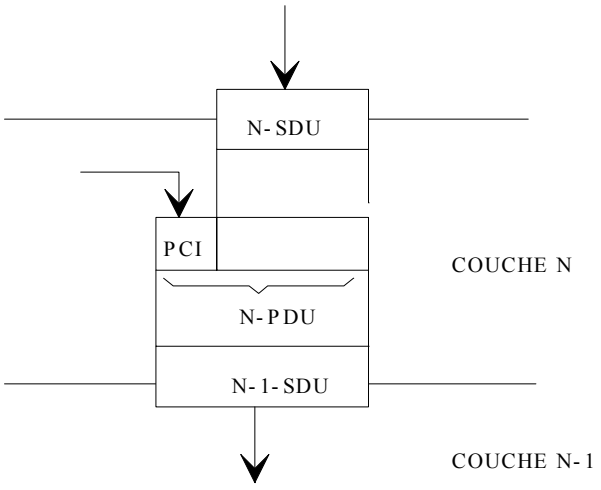



*Figure 6-3 Dialogue entre couches adjacentes.*

Pour une couche particulière du modèle OSI, on utilise les préfixes suivants :

- T = Transport (couche transport),
- N = Network (couche réseau),
- L = Link (couche liaison),

Par exemple, un paquet (X25.3) peut être appelé un NPDU.

## 6.3 Le transport ISO

Sa fonction essentielle est le contrôle de bout en bout, nécessaire en particulier quand on traverse un certain nombre de nœuds intermédiaires.

### 6.3.1 Fonctions de la couche transport ISO

- Détection d'erreurs : perte, déséquencement, duplication, altération de T-PDU.
- Reprise sur erreur : erreur signalée par le réseau, l'utilisateur du transport n'est pas averti.
- Contrôle de flux : permet de réguler l'échange de T-PDU de données entre 2 entités transport.
- Multiplexage/Démultiplexage : partage d'une connexion de réseau par plusieurs connexions de transport.
- Eclatement/recombinaison : C'est le mécanisme inverse, utilisation de plusieurs connexions de réseau pour une connexion de transport
- Segmentation/réassemblage : permet d'éclater une T-SDU dans plusieurs T-PDU, qui seront réassembler pour retrouver la T-SDU.
- Concaténation/Séparation.
- Données 'express'.

### 6.3.2 Les classes de transport

Toutes ces fonctionnalités ne sont pas forcément nécessaires : le niveau transport doit satisfaire un ensemble de critères qui dépend de la nature des besoins de l'utilisateur et parmi ces critères, certains sont déjà satisfaits par les services rendus par la couche réseau. On détermine donc une typologie des réseaux en fonction de leurs performances. On distingue deux catégories d'erreur : l'erreur signalée et l'erreur (résiduelle) non signalée. La première correspond à l'erreur survenue au niveau transport et la deuxième à l'erreur survenue au niveau des couches 1,2 et 3. Si la qualité exigée par l'utilisateur est équivalente à la qualité fournie par le réseau, la couche transport apporte peu de mécanismes complémentaires. Dans les autres cas, la couche transport devra cacher les imperfections de la couche réseau.

- Réseaux de type A : taux faible d'erreurs résiduelles (non signalées), taux faible d'erreurs signalées.
- Réseaux de type B : taux faible d'erreurs résiduelles (non signalées), taux élevé d'erreurs signalées.
- Réseaux de type C : taux élevé des deux catégories d'erreurs.

L'ISO définit 5 classes de service transport en fonction du degré de fiabilité du service rendu. Le choix d'une classe de service de la couche transport dépend de la qualité des services fournis par les couches inférieures et de la qualité exigée par la couche supérieure.

- Classe 0 : pas de multiplexage, pas de reprise sur erreur ; convient aux réseaux de type A.
- Classe 1 : pas de multiplexage, reprise sur erreur signalée ; convient aux réseaux de type B.
- Classe 2 : multiplexage avec ou sans contrôle de flux, pas de reprise sur erreur ; convient aux réseaux de type A.
- Classe 3 : multiplexage avec ou sans contrôle de flux, reprise sur erreur signalée ; convient aux réseaux de type B.
- Classe 4 : multiplexage avec ou sans contrôle de flux, reprise sur erreur signalée et non signalée ; convient aux réseaux de type C.

| | **Classe 0** | **Classe 1** | **Classe 2** | **Classe 3** | **Classe4** |
|---|---|---|---|---|---|
| Reprise sur erreur signalée | | oui | | oui | oui |
| Multiplexage | | | oui | oui | oui |
| Contrôle de flux | | | oui/non | oui/non | oui/non |
| Reprise sur erreur non signalée | | | | | oui |

*Figure 6-4 Les classes de la couche transport.*

### 6.3.3 Unités de données et primitives

L'unité manipulée par le transport ISO s'appelle un TPDU. Les primitives vont permettre l'établissement d'une connexion (T-CONNECT REQUEST, T-CONNECT RESPONSE, T-CONNECT INDICATION et T-CONNECT CONFIRM), le transfert de données normales (T-DATA REQUEST et T-DATA INDICATION), ), le transfert de données express (T-EXPEDITED-DATA REQUEST et T-EXPEDITED-DATA INDICATION),

et la libération de connexion (T-DISCONNECT REQUEST et T-DISCONNECT INDICATION). Une connexion de transport comprend les 3 phases initialisation, transfert et libération.

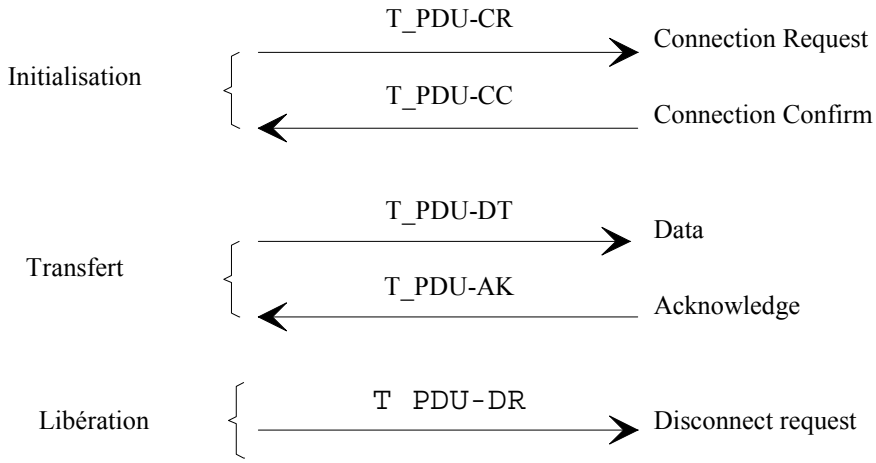



*Figure 6-5 Les 3 phases de la couche transport.*

## 6.4 La couche session

Elle établit et maintient des connexions entre processus. Ceci comprend la gestion de :

- la terminaison négociée (une entité ayant formulé une demande de terminaison négociée est encore capable de traiter les messages qu'elle reçoit jusqu'à la réception de la confirmation de terminaison),
- la synchronisation de processus (permet de poser des points de synchronisation à la demande de l'une ou l'autre application, et assure la resynchronisation dans les mêmes conditions).

Elle gère les droits de parole au moyen de jetons. Ceci permet d'établir un dialogue à l'alternat ou bien en full duplex, mais cela est indépendant de la transmission des données proprement dite.

## 6.5 La couche présentation

A l'origine, son utilité était justifiée par la grande variété de standards existant en matière de terminaux. Ce problème est maintenant traité par la couche application, la couche présentation s'occupant principalement de la représentation interne des données. Cela concerne :

- Le code utilisé : les grands systèmes IBM utilisent le code EBCDIC, alors que la plupart des autres ordinateurs utilisent l'ASCII.
- La taille des mots : 16 ou 32 bits, ainsi que la représentation des valeurs négatives par complément à 1 ou complément à 2.
- La numérotation des bits : les circuits 80286 et 80386 numérotent les bits de droite à gauche, les 68020 et 68030 le font de gauche à droite.

Le résultat d'un transfert de données entre machine hétérogènes peut donc être totalement imprévisible.
La norme ISO 8825 définit la Notation de Syntaxe Abstraite dite ASN.1 permettant de s'affranchir du problème de la représentation interne des données, qui définit, entre autre, les grands types de données (entier, booléen, suite de bits, suite d'octets…).
La couche présentation peut également s'occuper :

- du cryptage des données
- de la compression des données

## 6.6 La couche application

Parmi les applications normalisées, citons :

- les messageries (X400),
- les transferts de fichiers (FTAM),
- le terminal virtuel (VTS : Virtual Terminal Service),
- le dialogue d'application à application, (correspondant à la LU6.2 d'IBM), sous le nom de ISO-TP.

Ces applications sont désignées sous le nom de ASE (Application Service Elements). Comme il est apparu de nombreuses similitudes entre applications, on a introduit le concept de ACSE (Association Control Service Elements) permettant de gérer les associations d'application (messagerie et annuaire par exemple).

### 6.6.1 Les messageries X400

Une messagerie évoluée doit pouvoir résoudre les problèmes suivants :

- envoyer un message à un groupe de personnes,
- définir une structure de messages,
- savoir si un message envoyé est arrivé ou non,
- gérer le renvoi des messages ou le stockage en cas d'absence du destinataire,
- disposer d'un éditeur sans quitter la messagerie,
- créer et envoyer des documents multimédias (Texte, Image, Son).

La norme X400 discerne le message de son enveloppe, celle-ci portant des informations telle que l'adresse, la priorité, le niveau de sécurité, le type de terminal nécessaire à sa visualisation.
Les services offerts par une messagerie de type X400 sont principalement :

- la composition qui permet de construire un message,
- le transfert qui est la phase d'acheminement du message,
- l'information qui permet de dire à l'émetteur ce qui est advenu de son message,
- la conversion qui permet de s'adapter si cela est possible au terminal du destinataire (les types possibles pouvant être du texte ASCII, de la télécopie analogique ou numérique, du vidéotex, de la voix numérisée, du Télex, du Télétex).
- la mise en page pour visualiser un document,
- la remise permet au destinataire de traiter le message reçu.

Une messagerie X400 possède la structure indiquée sur la figure suivante.

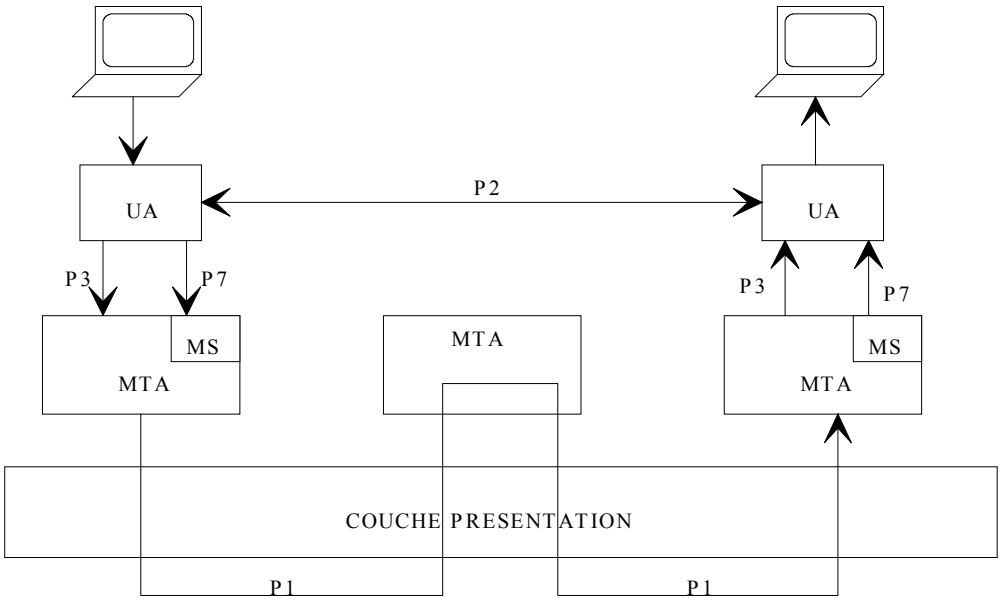



*Figure 6-6 Structure d'une messagerie X400.*

L'Agent Utilisateur (UA) gère l'utilisateur devant son terminal, dialogue avec l'agent de transfert de message, et manipule la mémoire de message.
L'Agent de Transfert de Message gère l'acheminement du message de proche en proche.
Le protocole P1 définit les règles de dialogue entre MTAs.
Le protocole P2 définit les règles de dialogue entre UAs.
Le protocole P3 définit les règles de dialogue entre UA et MTA.
Le protocole P7 définit l'interface entre UA et MS (boîte à lettre).
A noter que la norme X400 ne gère pas le contenu du message ; en particulier si deux correspondants échangent un fichier issu d'un traitement de texte, rien ne leur permet de s'indiquer quel est le logiciel utilisé, encore moins d'assurer la conversion entre deux formats.

### 6.6.2 Transfert de fichiers FTAM

### 6.6.3 Terminal virtuel VTS

## 6.7 Déroulement du dialogue entre 2 applications

Une application A1 du site 1 souhaite communiquer avec une application A2 du site 2. Elle demande à la couche session de la mettre en relation avec l'application A2 sur le site 2. La couche session demande à la couche transport de la mettre en relation avec son homologue. La couche transport détermine un chemin et choisit par exemple un réseau X.25. La couche réseau demande à la couche liaison d'établir une connexion. La couche liaison établit donc une connexion avec un commutateur X25 (Figure 6-7)

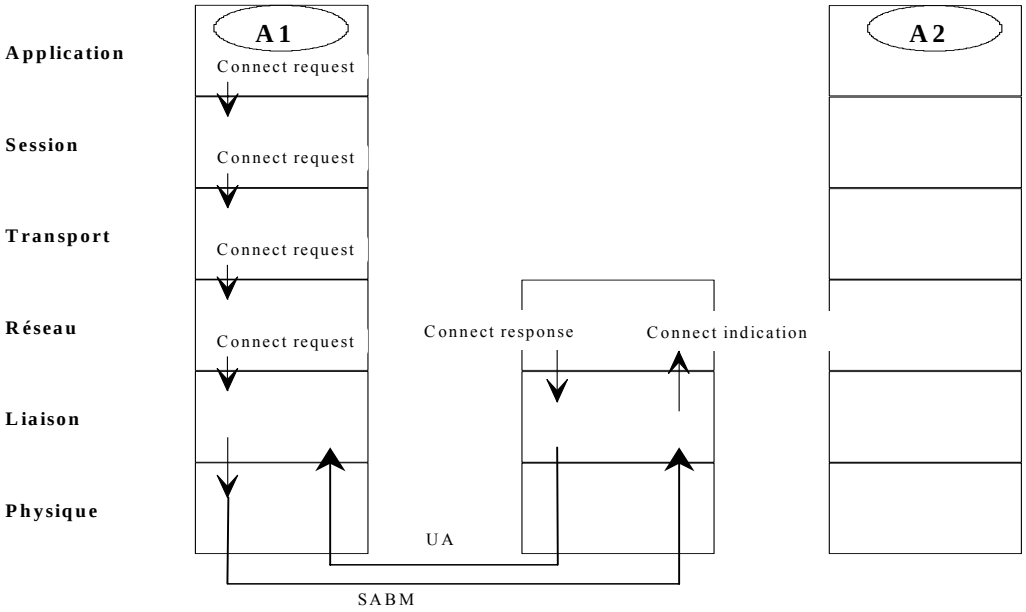



*Figure 6-7 Connexion liaison établie*

La couche réseau constitue un paquet d'appel qui est transmis au destinataire à travers le réseau. Le commutateur a au préalable établi une connexion HDLC avec le destinataire (Figure 6-8).

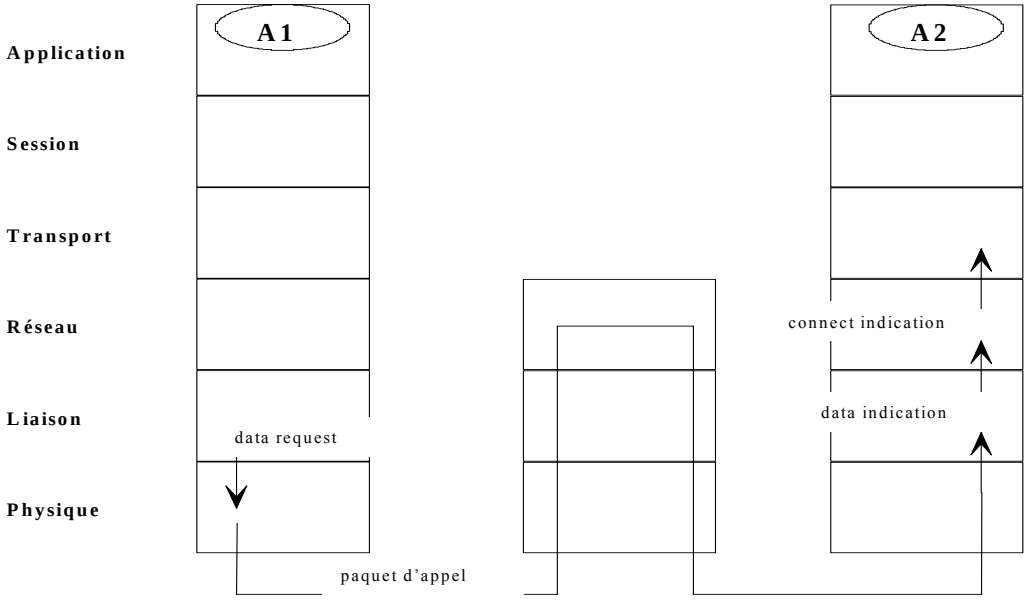



*Figure 6-8 Réception du paquet d'appel sur le site 2*

Le paquet d'appel est accepté et la connexion au niveau 3 est établie (Figure 6-9).

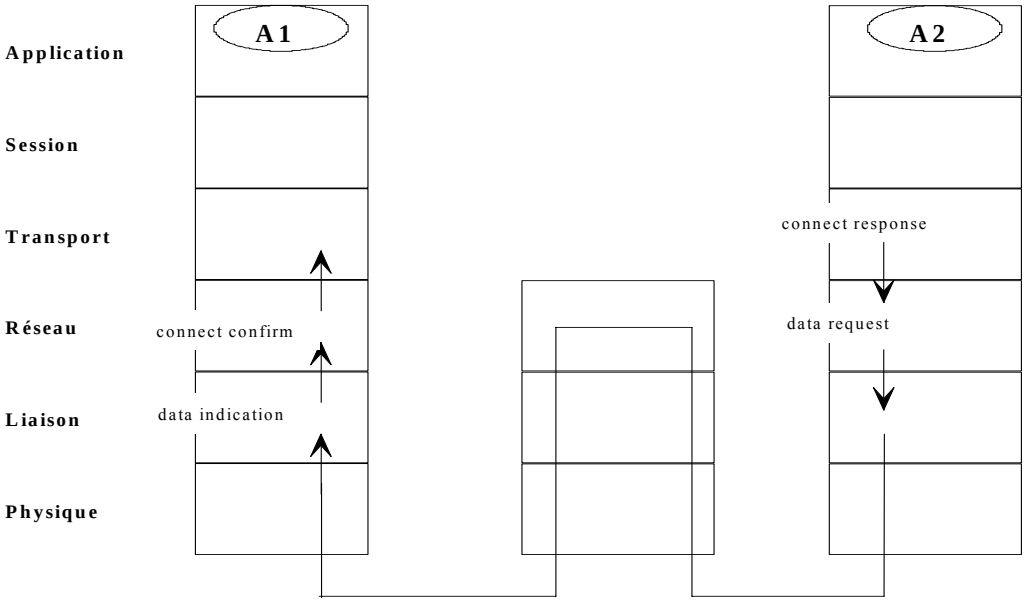



*Figure 6-9 Connexion réseau établie*

On établit la connexion au niveau transport (Figure 6-10).

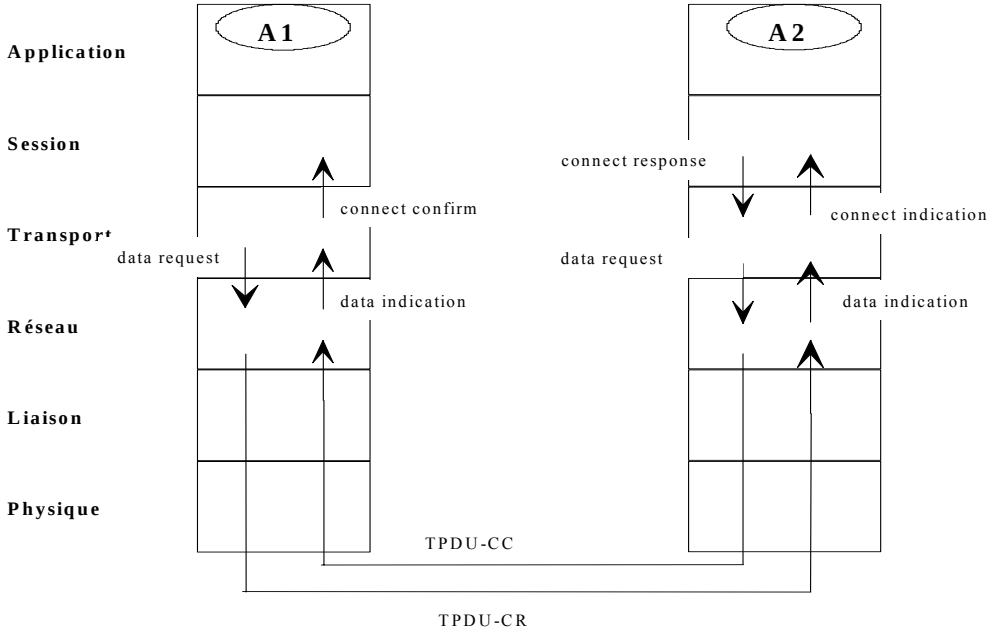



*Figure 6-10 Connexion transport établie*

Le niveau transport étant établi, on peut maintenant établir le niveau session (Figure 6-11).

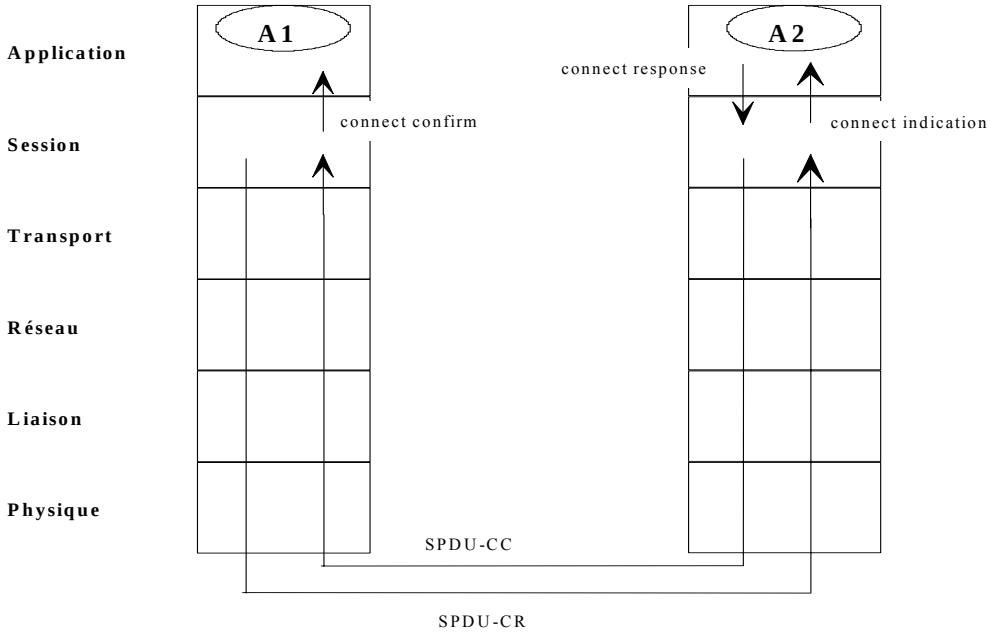



*Figure 6-11 Connexion session établie*

La connexion au niveau session étant établi, les deux applications sont en mesure d'échanger des données (Figure 6-12).

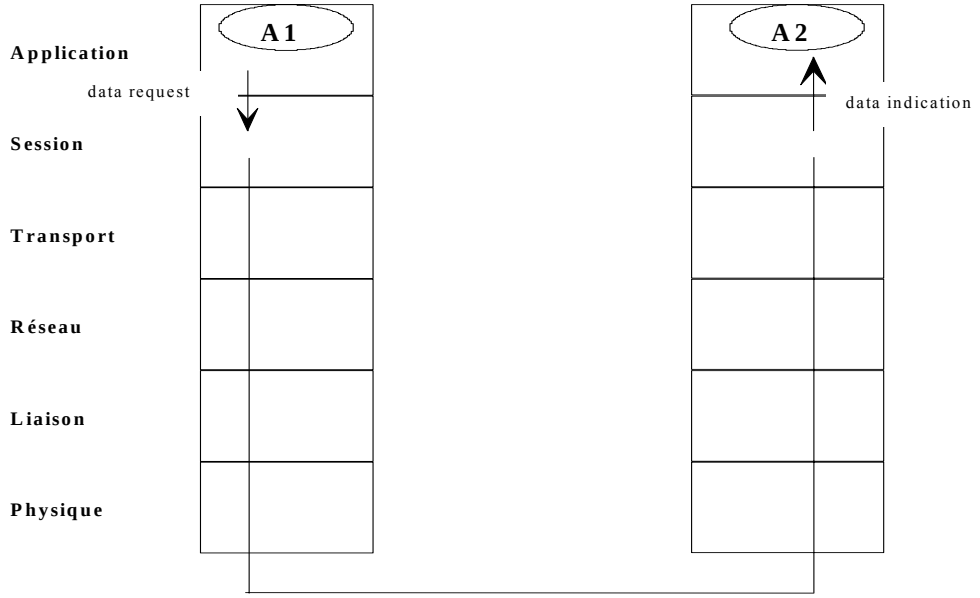



*Figure 6-12 Echange de données entre les applications*

# 7. Architecture TCP/IP

## 7.1 Historique

En 1971, le Network Working Group met au point le protocole de communication entre ordinateurs pour le réseau ARPANET appelé Network Control Protocol ou NCP. Bob Kahn, acteur majeur de l'ARPANET, envisagea d'utiliser NCP dans le cadre d'un projet de commutation de paquets par radio : ce protocole apparaissant mal adapté, il décida, en collaboration avec Vinton Cerf, chercheur à Stanford, de réaliser un nouveau protocole mieux adapté au contexte de ce projet et permettant de relier les réseaux. Un premier papier sur TCP/IP (Transmission Protocol, Internet Protocol) fût publié par ces deux chercheurs en Septembre 1973 lors d'une conférence de l'International Network Working Group (INWG). Dans les années 80, il devient un standard de l'interconnexion des réseaux, et constitue le cœur de la pile de protocoles utilisés dans Internet (c'est en janvier 1983 que le réseau ARPANET adopte le protocole TCP/IP à la place du protocole NCP.).

## 7.2 Le découpage en couches

Dans l'architecture TCP/IP, la couche physique et la couche liaison ne sont pas spécifiées : tous types de voies peut être envisagés (Ethernet, X25.2, une liaison téléphonique, SLIP, etc.). En plus de ces 2 couches, on trouve uniquement 3 couches dans TCP/IP, ce qui fait une architecture TCP/IP à 5 couches :

- protocoles réseau : IP, ICMP, ARP, RARP,
- protocoles transport: TCP ou UDP,
- couche application : avec les services de base tels que courrier électronique, transfert de fichiers...

Les deux premières couches permettent le dialogue entre les couches hautes de différentes applications. La figure suivante (Figure 6-1) fournit la correspondance entre les modèles TCP/IP et OSI.

<table>
<tr><td>OSI</td><td colspan="3">TCP/IP</td></tr>
<tr><td>7<br>6<br>5</td><td colspan="3">Application<br><br>telnet,ftp, ... DNS, ...</td></tr>
<tr><td>4</td><td colspan="2">TCP</td><td>UDP</td></tr>
<tr><td rowspan="3">3</td><td colspan="2" rowspan="2"></td><td>ICMP<br>IP</td></tr>
<tr><td>ARP</td><td>RARP</td></tr>
<tr><td colspan="3">Interface</td></tr>
<tr><td>2</td><td colspan="3">Liaison : Ethernet, X25-2</td></tr>
<tr><td>1</td><td colspan="3">Physique</td></tr>
</table>

*Figure 7-1 Correspondances entre les modèles TCP/IP et OSI.*

## 7.3 Protocole IP

Les informations suivantes concernent IP version 4.

Les fonctions du protocole IP (Internet Protocol) incluent :

- l'acheminement des datagrammes vers les machines distantes (routage)
- la fragmentation et le réassemblage des datagrammes

L'unité de données de transmission pour le protocole IP est le datagramme (Figure 7-2). IP est un protocole de remise non fiable en mode non-connecté : il n'échange aucune information de contrôle (message d'établissement de connexion, fermeture d'une connexion, ...) afin d'établir une connexion de bout en bout avant de transmettre les données et les datagrammes peuvent être perdus, dupliqués, retardés, altérés, remis dans le désordre. Si un protocole des couches supérieures requiert un service orienté connexion, il devra prendra soin d'établir la connexion. Le protocole IP est implanté sur toutes les machines (hôtes et routeurs) connectées à un réseau TCP/IP.

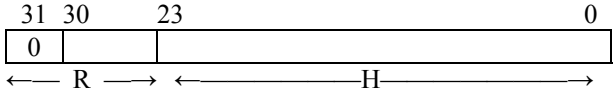
### 7.3.1 Adressage IP

L'adresse IP est codée sur 4 octets notés sous la forme décimale, comme par exemple 147.210.94.100. Elle est composée de deux parties : la première composante correspond au numéro de réseau (R) et la seconde composante correspond à un numéro d'hôte (H) de ce réseau. Les réseaux sont classés en trois catégories selon leur importance en fonction du nombre de machines qui les composent.

Les 32 bits de l'adresse IPv4 sont vus comme une suite de quatre nombres compris entre 0 et 255.

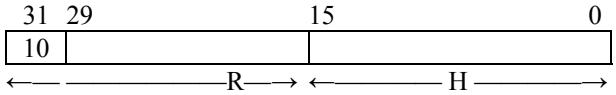
- Réseau classe A de grande taille

Le premier des quatre nombres est réservé au numéro de réseau, et les 3 autres correspondent au numéro de la machine dans le réseau, c'est la partie locale.



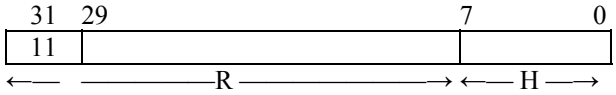

- Réseau classe B de taille moyenne

Les deux premiers nombres sont réservés pour la partie réseau, et les deux derniers pour la partie locale.




- Réseau classe C de petite taille

Les trois premiers nombres sont réservés pour la partie réseau, et le dernier pour la partie locale.




Remarque :
Les valeurs 0 et 255 ont parfois une signification particulière et ne peuvent être utilisées dans certains cas pour constituer une adresse IP.

| Classe de réseau | Intervalle des adresses utilisées | |
|---|---|---|
| A | 0.0.0.0 → 127.255.255.255 | |
| B | 128.0.0.0 → 191.255.255.255 | |
| C | 192.0.0.0 → 223.255.255.255 | |
| D | 224.0.0.0 → 239.255.255.255 | Adresse de diffusion (mode multi-casting d'IP) |
| divers | 240.0.0.0 → 255.255.255.255 | |

Considérons la machine d'adresse IP *147.210.94.1*. Elle fait partie d'un réseau de classe *B* d'adresse *147.210* et l'adresse de la machine au sein du réseau est *94.1*.
La table *hosts* fournit pour chaque adresse IP, la liste des noms d'hôte associé à cette adresse. Sur les machines UNIX, la table *hosts* est enregistrée dans le fichier */etc/hosts* dont voici un extrait :

```
sem:~  cat /etc/hosts
#ident "@(#) hosts   1.1        2/7/91 common"
#ident "@(#)common:hosts        1.3 11/28/89 16:01:56"
# cmd-inet:etc/hosts    1.1
#
# Internet host table
#
0.0.0.0         anyhost
127.0.0.1       localhost

147.210.94.1    minuit
147.210.94.92   sem
147.210.94.99   thor happy
147.210.94.100  noe
147.210.94.200  pentium
# terminaux X Couleur (salle 004 : sem )
147.210.94.101  aigle
…
# divers
147.210.94.254  agsbdxiii
# imprimantes laser
147.210.94.91   labas

147.210.94.128 pizza
```

Sur les machines UNIX qui ont leur propre disque, le fichier */etc/resolv.conf* contient le nom du domaine et l'adresse IP.

```
sem:~  cat /etc/resolv.conf
domain iuta.u-bordeaux.fr
nameserver 147.210.94.1
```

### 7.3.2 Sous-réseau

Pour diverses raisons (performances, découpage en service, répartition géographique…), on est amené à délimiter des sous-réseaux à l'intérieur d'un même réseau. Le masque de sous-réseau permet une subdivision

plus fine de l'adresse IP, en introduisant un champ supplémentaire appelé sous-réseau : le nombre de bits composant ce champ détermine le nombre de sous-réseaux disponibles.
Par exemple : prenons un réseau d'adresses 1.2.3.*. Pour constituer 2 sous-réseaux, on prendra 2 bits pour identifier la partie sous-réseau : 01 et 10 (les valeurs 'tout à 1' et 'tout à 0', c'est-à-dire 00 et 11, étant non autorisées). Les sous-réseaux auront alors les adresses suivantes : 1.2.3.128 et 1.2.3.64, et le masque de sous-réseau est alors : 255.255.255.192. L'adresse IP 1.2.3.130 correspond à une machine du sous-réseau 1.2.3.128 : cette adresse est obtenue simplement en effectuant un ET logique entre l'adresse et le masque.

| 1 | 2 | 3 | 130 |
|---|---|---|---|
| 0 0 0 0 0 0 0 0 1 | 0 0 0 0 0 0 1 0 | 0 0 0 0 0 0 1 1 | 1 0 0 0 0 0 1 0 |
| A.R | | | S.R \| A.M. |

ET¶

| 255 | 255 | 255 | 192 |
|---|---|---|---|
| 1 1 1 1 1 1 1 1 1 | 1 1 1 1 1 1 1 1 | 1 1 1 1 1 1 1 1 | 1 1 0 0 0 0 0 0 |

EGAL¶

| 1 | 2 | 3 | 128 |
|---|---|---|---|
| 0 0 0 0 0 0 0 0 1 | 0 0 0 0 0 0 1 0 | 0 0 0 0 0 0 1 1 | 1 0 0 0 0 0 0 0 |

(A.R, S.R et A.M désignent respectivement adresse réseau, sous-réseau et adresse machine.)

### 7.3.3 Structure du datagramme IP

La figure suivante fournit la structure du datagramme IP.

| 0 | 8 | 16 | 24 31 |
|---|---|---|---|
| Version \| Long. Ent. | Type de service | Longueur totale | |
| Identification | | Flags | Offset (fragment) |
| Durée de vie | Protocole transporté | Checksum | |
| Adresse IP source | | | |
| Adresse IP destinataire | | | |
| Options | | | bourrage |
| Données (Segment de niveau supérieur) | | | |

*Figure 7-2 Structure du datagramme IP.*

Signification et taille des différents champs :

**Version** (4 bits) : Numéro de version du protocole
**Long. Ent.** (4 bits) : Longueur de l'en-tête en mots de 32 bits (les options font partie de l'en-tête),
**Type de service** (8 bits) : Priorité au délai, au débit, à la fiabilité,
- 00 pas de service,
- 08 priorité au délai,
- 10 priorité au débit,
- 20 priorité à la fiabilité,
- ...

**Longueur totale** (16 bits) : Longueur totale, en octets, entête et données comprises,
**Identification** (16 bits) : Identifiant du datagramme
**Flags** (3 bits) : Gestion de la fragmentation
- 000 dernier fragment
- 001 fragment à suivre
- 010 pas de fragment

**Offset** (13 bits) : Décalage de la fragmentation: position relative par rapport au début du datagramme initial, si celui-ci a été fragmenté (exprimé en unité de 8 octets),
**Durée de vie** (8 bits) : Temps écoulé depuis l'émission, exprimé en nombre de sauts,
**Protocole transporté** (8 bits) : Protocole de niveau supérieur
- $01_h$ ICMP
- $06_h$ TCP
- $09_h$ IGP
- $11_h$ UDP
- $1D_h$ Transport ISO Classe 4
- ...

**Checksum** (16 bits) : Total de contrôle,

**Adresse IP source** (32 bits) Adresse IP de l'émetteur
**Adresse IP destinataire** (32 bits) Adresse IP du destinataire
**Options** (n bits) : Facultatives,
**Bourrage** (n bits) : Caractères de 'bourrage' si nécessaire,
**Données** (n bits) : Les données.

Voici un extrait du fichier PROTOCOL qui donne quelques exemples de numéros de protocole (valeur en décimale) :

```
# Copyright (c) 1994 Microsoft Corp.
#
# This file contains the Internet protocols as defined by RFC 1060
# (Assigned Numbers).
#
# Format:
#
# <protocol name>  <assigned number>  [aliases...]   [#<comment>]

ip       0     IP       # Internet protocol
icmp     1     ICMP     # Internet control message protocol
ggp      3     GGP      # Gateway-gateway protocol
tcp      6     TCP      # Transmission control protocol
egp      8     EGP      # Exterior gateway protocol
pup      12    PUP      # PARC universal packet protocol
udp      17    UDP      # User datagram protocol
hmp      20    HMP      # Host monitoring protocol
xns-idp  22    XNS-IDP  # Xerox NS IDP
rdp      27    RDP      # "reliable datagram" protocol
rvd      66    RVD      # MIT remote virtual disk
```

### 7.3.4 Protocole ARP

Au sein d'un réseau local, chaque machine est identifiée par son adresse "physique" (par exemple, dans un réseau local de type Ethernet, chaque machine possède une adresse Ethernet constituée de 6 octets). Chaque machine possède aussi une adresse de niveau 3, comme par exemple une adresse IP si ce réseau local forme un réseau IP.

Il faut donc mettre en correspondance les deux types d'adresse. Cela peut se faire de deux façons:

- par correspondance directe lorsque l'adresse physique peut être incluse dans l'adresse de niveau 3,
- lorsque cela n'est pas possible (cas des adresses Ethernet avec IP) il faut gérer sur chaque machine une table de correspondance, qui doit évoluer dans le temps.

Le protocole ARP (Address Resolution Protocol) donne une solution à cette deuxième approche :

Une machine qui souhaite connaître l'adresse physique d'une autre machine , dont elle ne connaît que l'adresse IP diffuse à toutes les machines du réseau un message où elle demande à son correspondant de s'identifier. Seule la machine concernée répond à ce message en indiquant sa correspondance adresse physique/adresse IP. Chaque machine peut donc mettre à profit ce type de message pour mettre à jour dynamiquement une table en mémoire cache.

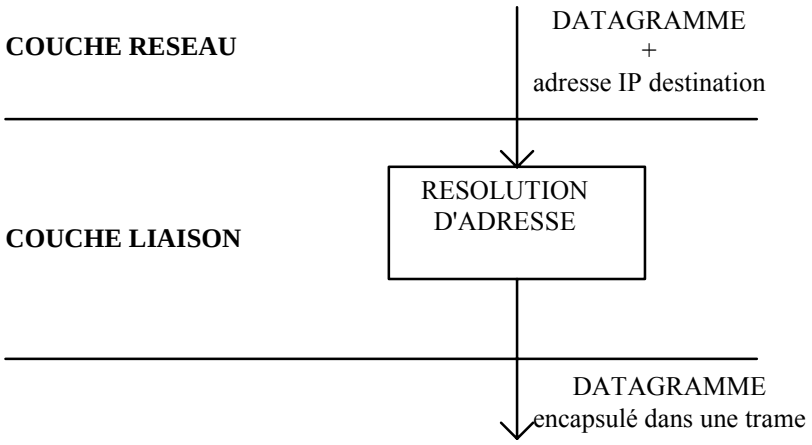



*Figure 7-3 Conversion adresse IP en adresse physique*

### 7.3.5 Routage des datagrammes IP

L'acheminement ou routage correspond à la sélection du routeur à utiliser pour la transmission de données. IP détermine le routage approprié pour chaque paquet. Pour transmettre un datagramme entre 2 machines faisant parties d'un même réseau, il n'y a pas de problème particulier, on utilise la résolution d'adresse physique citée plus haut. C'est ce qu'on appelle la remise directe.
Lorsque émetteur et destinataire ne font pas partie du même réseau local, on est conduit à traverser une ou plusieurs routeurs. Un routeur possède nécessairement au moins deux connexions physiques et deux adresses IP. Un datagramme transite de routeur en routeur jusqu'à ce que l'une d'elles puisse effectuer une remise directe au destinataire.
Machines et routeurs doivent utiliser des tables de routage. Une table de routage est un ensemble de couples (adresse réseau destinataire, adresse IP de routeur)

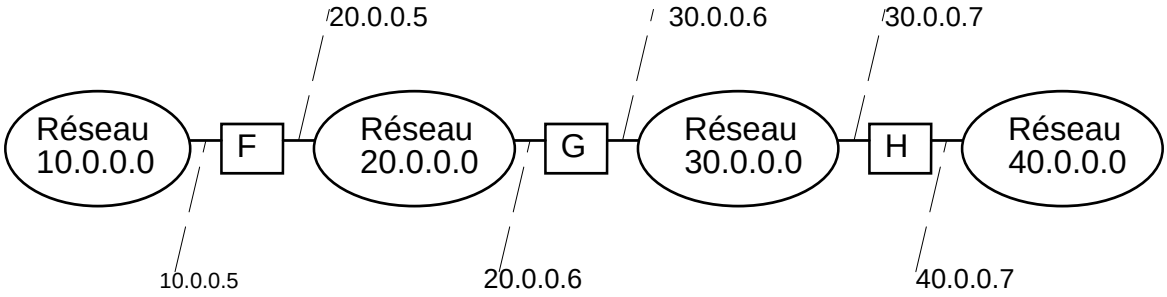



*Figure 7-4 Réseaux interconnectés*

| Pour atteindre les machines situées sur le réseau | router vers cette adresse |
|---|---|
| 20.0.0.0 | Remise directe |
| 30.0.0.0 | Remise directe |
| 10.0.0.0 | 20.0.0.5 |
| 40.0.0.0 | 30.0.0.7 |

*Figure 7-5 Table de routage de G*

On peut également recourir à une "route par défaut": toutes les destinations non répertoriées sont routées vers un routeur donné.
L'adresse IP du prochain routeur ("de prochain saut") ne peut être conservée dans le datagramme, qui ne possède que les adresses émetteur et récepteur du datagramme, et n'est pas modifiée en cours de transfert. Elle est en fait traduite en adresse physique.

### 7.3.6 Fragmentation des datagrammes.

Chaque type de réseaux possède une unité de transfert maximale (MTU : Maximum Transfert Unit) qui correspond à la taille du plus grand paquet que le réseau est capable de transférer. Comme le datagramme doit être acheminé sur différents réseaux, il est possible que sa taille soit trop importante pour un type de réseau traversé, et sa division en éléments de plus petites tailles doit être effectuée par un routeur. Cette opération porte le nom de fragmentation.
Les fonctions de flux, de contrôle d'erreur, et de séquencement sont reportées à des protocoles du niveau supérieure.

### 7.3.7 Interface

Cette partie définit comment transporter les datagrammes IP en fonction du type de liaison (Ethernet, X25, FDDI, liaison série : Point_to_Point_Protocol).

### 7.3.8 Protocole RARP.

Le protocole RARP (Reverse Address Resolution Protocol) effectue l'opération inverse du protocole ARP : à partir d'une adresse Ethernet, il fournit l'adresse IP.

### 7.3.9 ICMP

ICMP (Internet Control Message Protocol) est un protocole de même niveau que IP. Il sert essentiellement, lorsque cela est possible, à avertir l'émetteur d'un datagramme IP d'un problème survenu lors de son acheminement.
Les principaux cas d'utilisation de ICMP sont:
- demande d'écho (utile pour la mise au point et exploité par la commande ping),
- réponse d'écho,
- destination inaccessible (le datagramme a été détruit),
- demande de limitation de débit (une routeur est proche de la congestion),
- demande de modification de route (une routeur sait qu'il existe une meilleure route),
- boucle de routage,
- durée de vie expirée (ou durée de réassemblage expirée).

## 7.4 Protocole TCP

TCP est un protocole de transfert fiable en mode connecté. Son usage est rendu nécessaire du fait que IP est un protocole de remise non fiable. TCP est très proche du transport ISO classe 4. Il fournit un service circuit virtuel et utilise un système d'acquittement pour le contrôle d'erreur. L'unité de données de transmission pour le protocole TCP est le segment (Figure 7-6).

### 7.4.1 Principes

- Les messages sont considérés comme des flots d'octets (les numéros de séquence sont des numéros d'octet et non de message). Cette notion n'est pas surprenante sous UNIX.
- TCP met en oeuvre un mécanisme d'anticipation, par "fenêtre glissante"
- La taille de fenêtre d'émission est variable dans le temps, et elle est contrôlée par le récepteur. Si celui-ci ne peut accepter de nouveaux messages, il demande à l'émetteur de réduire sa taille de fenêtre.
- On acquitte par rapport au dernier octet de la séquence complète reçue (s'il y a un "trou", on n'acquitte pas la suite).
- TCP utilise un mécanisme de retransmission sur temporisateur: Si un segment n'a pas été acquitté à l'expiration d'un certain délai, le segment est retransmis. Le délai est adaptatif : si tout se passe bien, le délai augmente ; en cas de problème, il se réduit.

### 7.4.2 Mécanisme de port

Le port est l'interface entre la couche supérieure (application) et TCP. Un numéro de port permet d'identifier un dialogue de niveau applicatif en cours sur une connexion TCP. Un port peut être vu schématiquement comme une file d'attente pour les messages à destination d'une application. Toutefois, cette notion est un peu plus subtile : une connexion est définie par ses deux extrémités, chaque extrémité étant elle-même un couple (adresse IP, numéro de port). Ainsi une messagerie peut gérer plusieurs dialogues simultanément à partir d'un même numéro de port.
Il existe des numéros de port réservés (<1024) et d'autres à la libre disposition des utilisateurs. Dans ce cas, la machine qui est à l'origine de la demande de connexion doit d'abord émettre une demande du type "quel port puis-je utiliser pour un transfert de fichiers?".
Voici un extrait du fichier SERVICES qui donne quelques exemples de numéros de port associés aux applications les plus classiques :

```
# Copyright (c) 1994 Microsoft Corp.
#
# This file contains port numbers for well-known services as defined by
# RFC 1060 (Assigned Numbers).
#
# Format:
#
# <service name>  <port number>/<protocol>  [aliases...]   [#<comment>]
#

echo                 7/tcp
echo                 7/udp
netstat             15/tcp
ftp-data            20/tcp
ftp                 21/tcp
telnet              23/tcp
smtp                25/tcp    mail
time                37/tcp    timserver
time                37/udp    timserver
```

```
rlp                39/udp    resource      # resource location
name               42/tcp    nameserver
name               42/udp    nameserver
whois              43/tcp    nicname       # usually to sri-nic
domain             53/tcp    nameserver    # name-domain server
domain             53/udp    nameserver
nameserver         53/tcp    domain        # name-domain server
nameserver         53/udp    domain
finger             79/tcp
hostnames         101/tcp    hostname      # usually from sri-nic
iso-tsap          102/tcp
x400              103/tcp                  # ISO Mail
x400-snd          104/tcp
pop               109/tcp    postoffice
pop2              109/tcp                  # Post Office
pop3              110/tcp    postoffice
NeWS              144/tcp    news
biff              512/udp    comsat
exec              512/tcp
route             520/udp    router routed
timed             525/udp    timeserver
uucp              540/tcp    uucpd         # uucp daemon
maze             1666/udp
nfs              2049/udp                  # sun nfs
```

### 7.4.3 Structure du segment TCP

La figure suivante fournit la structure du segment TCP.

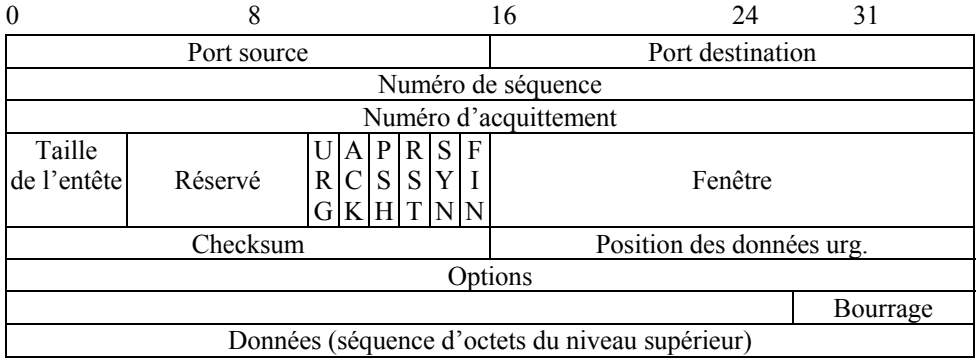



*Figure 7-6 Structure du segment TCP.*

Signification et taille (en nombre de bits) des différents champs :

**Port Source (16)** : Numéro du port source,
**Port Destination (16)** : Numéro du port destination,
**Numéro de séquence (32)** : Numéro de séquence du premier octet de ce segment,
**Numéro d'acquittement (32)** : Numéro du prochain octet attendu,
**Taille de l'entête (4)** : Longueur de l'en-tête en mots de 32 bits (les options font partie de l'en-tête),
**Réservé (6)** : Réservé pour une prochaine utilisation...
**Bit URG (1)** : Le pointeur de données urgentes est valide,
**Bit ACK (1)** : Le numéro d'acquittement est valide,
**Bit PSH (1)** : Données à envoyer de suite (push),
**Bit RST (1)** : Provoque la rupture anormale de la connexion
**Bit SYN (1)** : Demande l'établissement de la connexion,
**Bit FIN (1)** : Demande la fin de la connexion,
**Fenêtre (16)** : Taille de fenêtre demandée à l'émission,
**Checksum (16)** : Total de contrôle,
**Position des données urg. (16)** : S'il y en a, fournit la position relative des dernières données urgentes,
**Options (n)** : Facultatives
**Bourrage (n)** : Caractères de bourrage (si nécessaire),
**Données (n)** : Séquence d'octets du niveau supérieur.

Les ports destination et source destination identifient les programmes d'application aux deux extrémités.

Tous les segments TCP ont la même structure. Ce sont les bits URG, ACK, PSH, RST, SYN et FIN qui permettent, de façon éventuellement cumulative, de préciser la ou les fonctions du segment.
-Le champ FENETRE permet d'interagir sur la taille de la fenêtre émission de l'autre extrémité.
Lors de la connexion (bit SYN), les extrémités déterminent les numéros de séquence initiaux. Ces numéros ne peuvent pas être 0 en raison de la possibilité de réinitialisation (RST): il y aurait risque d'ambiguïté entre le flot "avant" réinitialisation et le flot "après". Le push est la remise forcée de données à la couche application.
-Le champ POINTEUR D'URGENCE permet de repérer dans le flot de données la position de données urgentes (qui doivent "doubler" les autres données) lorsque le bit URG est positionné.
-Le champ OPTION permet entre autres la négociation de la taille de segment à la connexion.

### 7.4.4 Exemple de scénario : demande de connexion

Vous pouvez à titre d'exercice dépouiller l'échange suivant de segments TCP.

```
Emission:     BA BA 00 17 12 34 00 00 00 00 00 60 02 10 00
              24 33 00 00 02 04 04 00

Reception:    00 17 BA BA AB CD 00 00 12 34 00 00 60 12 10 00
              65 40 00 00 02 04 04 00

Emission:     BA BA 00 17 12 34 00 01 AB CD 00 01 50 10 10 00
              16 33 00 00

Reception:    00 17 BA BA AB CD 00 01 12 34 00 01 50 18 10 00
              33 17 00 00 6C 6F 67 69 6E 3A

Emission:     BA BA 00 17 12 34 00 01 AB CD 00 07 50 10 10 00
              16 33 00 00
```

## 7.5 Protocole UDP

Par rapport à TCP, UDP (User Datagram Protocol) est un protocole "mince" qui n'apporte que très peu de valeurs ajoutées par rapport à IP : il offre un service de remise non fiable en mode sans connexion (les messages UDP peuvent être perdus, déséquencés, dupliqués ou retardés). UDP utilise un mécanisme de "port" identique à TCP. Il est utilisé par des applications qui n'exigent pas tous les contrôles effectués par TCP.

## 7.6 La couche application

Plusieurs applications sont devenues des standards dans le monde Internet. On peut citer, par exemple, FTP (File Transfert Protocol) pour le transfert de fichier, Telnet pour la connexion à un ordinateur distant, DNS (Domain Name Service) pour le protocole de serveurs de noms. Ces applications, basées sur le modèle Client/Serveur, dialoguent entre elles par l'intermédiaire de ports d'un protocole de la couche inférieure (Figure 7-7).

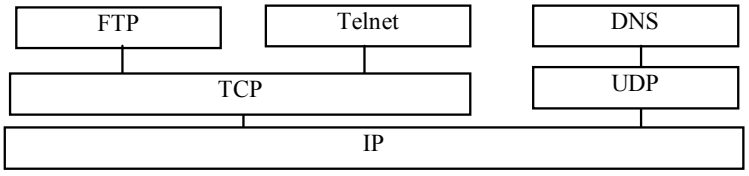



*Figure 7-7 Une application utilise UDP ou TCP*