

Réseaux : 2° Partie



Modèle de référence : il décrit les fonctionnalités nécessaires à la communication et l'organisation de ces fonctions.

Le modèle est essentiellement une architecture en couches définies et délimitées avec les notions de service, de protocole et d'interface.

- *Un service est une description de fonctionnalités.*
- *Un protocole est un ensemble de messages et de règles d'échanges réalisant un service.*
- *Une interface (« point d'accès au service » dans la norme) est le moyen concret d'utiliser le service.*

Les couches inférieures sont plutôt orientées communication et sont souvent fournies par un système d'exploitation.

Les couches supérieures sont plutôt orientées application et plutôt réalisées par des bibliothèques ou un programme spécifique.



Bibliographie

- "TCP/IP sous Linux" de JF Bouchaudy - Formation Tsoft © Ed. Eyrolles
- "TCP/IP Administration de réseau" de Craig Hunt © Ed. O'Reilly
- "Les protocoles TCP/IP et Internet" d'Eric Lapaille © NetLine 1999
- "Technique des réseaux locaux sous Unix" de L. Toutain © Ed. Hermes
- "Pratique des réseaux locaux d'entreprise" de JL Montagnier © Ed. Eyrolles
- "Transmission et Réseaux" de S. Lohier et D. Present © ED. DUNOD
- Les sites www.frameip.com, fr.wikipedia.org, www.w3.org, etc ...

© Copyright 2010 tv <tvaira@free.fr>

Permission is granted to copy, distribute and/or modify this document under the terms of the **GNU Free Documentation License**, Version 1.1 or any later version published by the Free Software Foundation; with no Invariant Sections, with no Front-Cover Texts, and with no Back-Cover.

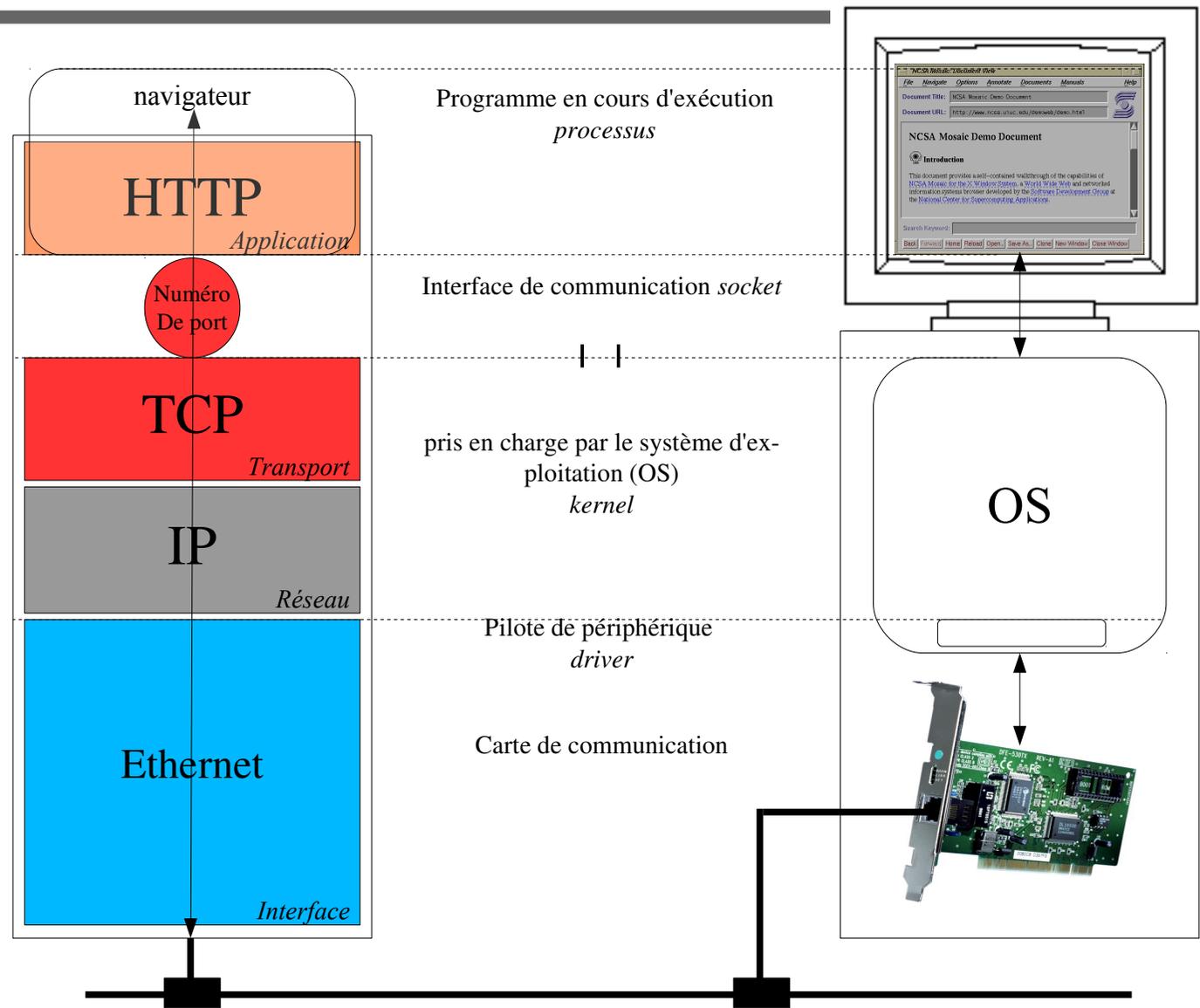
You can obtain a copy of the GNU General Public License :

write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA



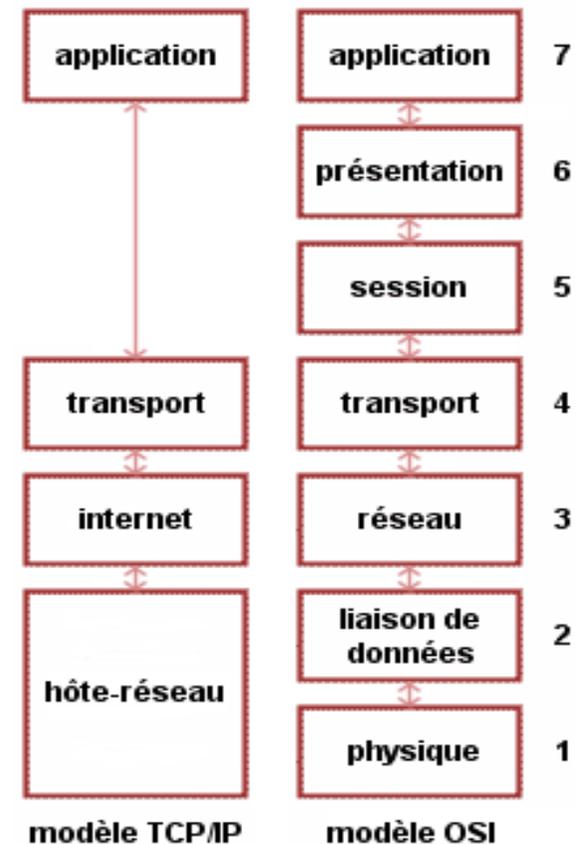
Rappels : Modèle de référence (I)

- Un modèle de référence est utilisé pour décrire la structure et le fonctionnement des communications réseaux
- Le modèle DoD (*Department of Defense*) ou « TCP/IP » est composé de 4 couches
- En raison de son apparence, la structure est très souvent appelé **pile** ou **pile de protocoles**.



Rappels : Modèle de référence (II)

- Un modèle de représentation développé par l'**ISO** (*International Standards Organization*) est souvent utilisé pour décrire la structure et le fonctionnement des communications réseaux : le modèle **OSI** (*Open Systems Interconnect Reference Model*).
- Le modèle OSI contient **7 couches ou niveaux** qui définissent les **fonctions** des protocoles de communication qui vont de l'interface physique à l'interface avec les applicatifs utilisant le réseau.
- Critiques du modèle OSI :
 - Ce n'était pas le bon moment : trop tôt ou trop tard ?
 - Ce n'était pas la bonne technologie : trop complet et trop complexe
 - Ce n'était pas la bonne implémentation : trop lourd et trop lent
 - Ce n'était pas la bonne politique : trop normalisé et trop bureaucratique



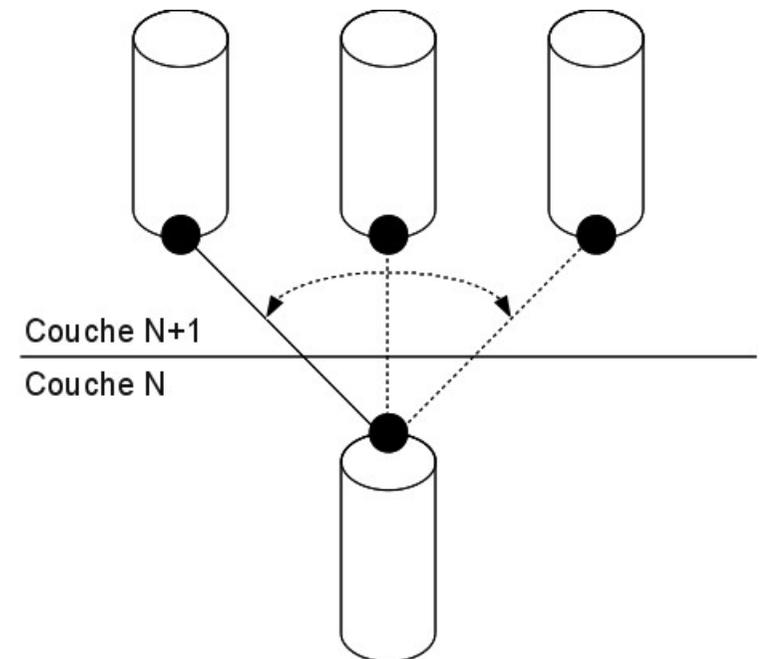
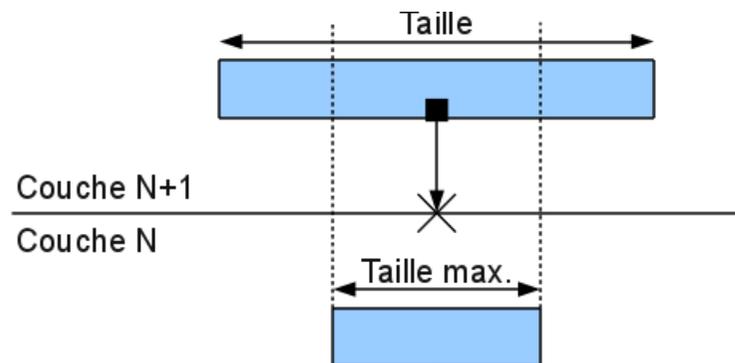
0.a. Les fonctions communes

- Fiabilisation des communications : l'un des rôles majeurs des couches est la construction d'une connexion exempte d'erreurs de transmission.
 - Cela signifie que les données transmises sont reçues sans corruption, perte, réordonnancement et duplication.
 - Cela implique qu'au moins une couche, et en pratique plusieurs, fasse de la détection d'erreur, de la correction d'erreur ou de la retransmission de données et du contrôle de flux.



0.b. Les fonctions communes

- Fonctions de transformation : le modèle définit aussi une série de mécanismes standards de manipulation de commandes ou de données, utilisées pour la réalisation d'un service.
- Multiplexage et démultiplexage de connexion, Éclatement et recombinaison
- Segmentation, fragmentation et réassemblage



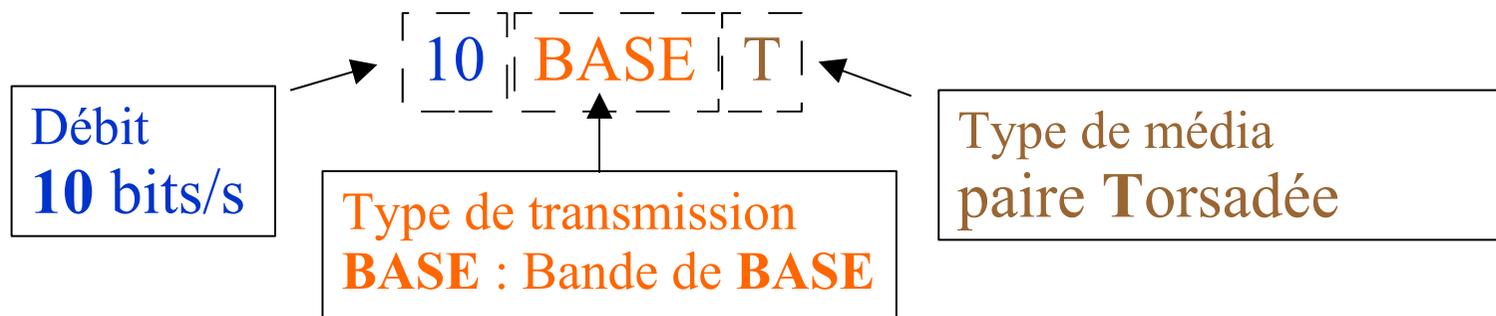
1. La couche Physique

- Son rôle est la transmission des **bits** sur le support physique.
- Ces fonctions principales sont :
 - fournir les caractéristiques électriques, mécaniques et fonctionnelles ;
 - définir le mode d'exploitation (semi-duplex, duplex intégral, //, série, ...) de la liaison (point à point ou multipoint) ;
 - assurer la compatibilité des interfaces qui réalisent les fonctions de codage, modulation et amplificateur du signal.
- ◆ Exemples : V24, X21, RS232, *Transceiver*, MAU, HUB, Répéteur, Prise DB25 et DB9, RJ45 et RJ11, etc ...



2. Ethernet (couche physique)

- Mise au point dans les années 80 par XEROX, Intel et DEC, l'architecture Ethernet permet l'interconnexion de matériels divers avec de grandes facilités d'extension.
- Les différentes normes :
 - Ethernet à 10 Mbits/s : 10BASET, 10BASE5, 10BASE2 et 10BASEF (802.3)
 - FastEthernet à 100 Mbits/s : 100BASET (802.3u)
 - Gigabit Ethernet (Gig-E) à 1 Gbits/s : 1000BASE-LX, 1000BASE-SX, 1000BASE-CX, 1000BASE-LH (802.3z) et 1000BASE-T (802.3ab)
 - Décagigabit Ethernet à 10 Gbits/s (10 Gigabit Ethernet) ...



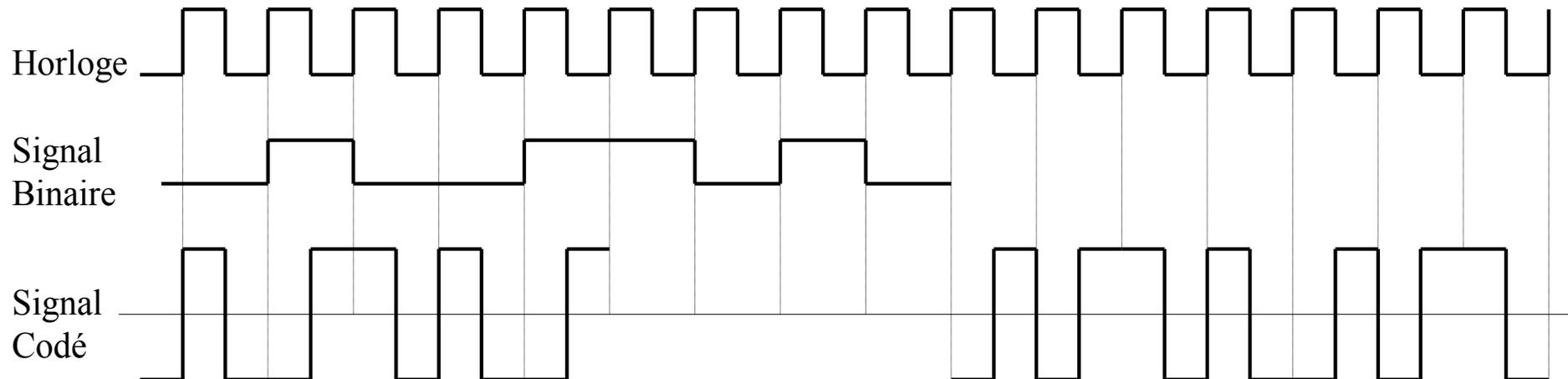
3. Transmission bande de base (BdB)

- La transmission en bande de base est une **transmission numérique** (sans transposition de fréquence par modulation).
- Les codages numériques sont utilisés pour plusieurs raisons :
 - la récupération du signal d'horloge facilitée par les transitions pour chaque bit transmis
 - le spectre d'un signal binaire est concentré sur les fréquences basses (les plus affaiblies)
 - les perturbations subies par un signal sont proportionnelles à la largeur de sa bande de fréquence.
- Les codages en bande de base vont donc essentiellement avoir pour rôle de diminuer la largeur de bande du signal binaire, de transposer celle-ci vers des fréquences plus élevées et d'utiliser les transitions du signal afin d'assurer une **transmission synchrone** (qui permettront au récepteur de synchroniser son horloge).
- Quelques codages utilisés sur Ethernet :
 - 10BASET : codage manchester (ou exclusif entre les DATA et l'Horloge)
 - 100BASET : codage 4B/5B MLT3 (**paires torsadées**)
 - 100BASEFX : codage 4B/5B NRZI (fibre optique), le GigaBit utilise le codage 8B/10B



4. Codage Manchester pour le 10 Mbps

- Le codage Manchester est utilisé sur le 10BASET :



Principe : Une opération XOR (ou exclusif) est réalisée entre l'horloge et les données, d'où une transition systématique au milieu de chaque bit du signal binaire.

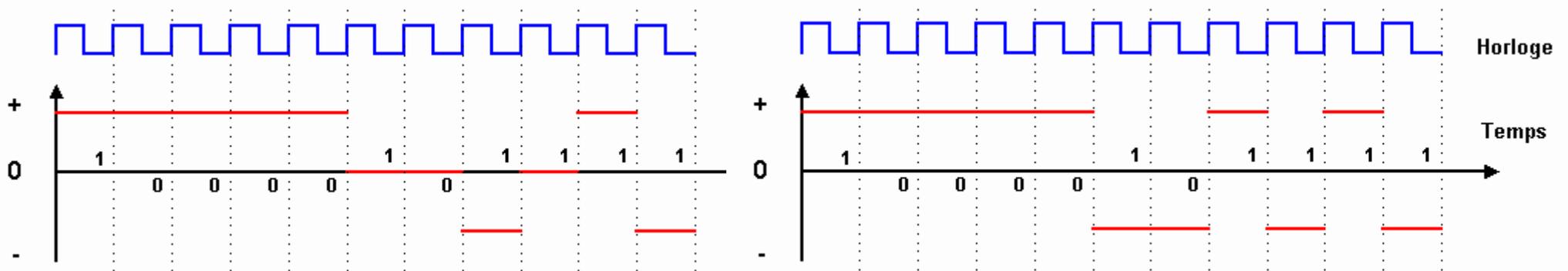


5. Codage NRZI/MLT3 pour le 100 Mbps

Dans ce codage MLT3, seuls les 1 font changer le signal d'état en prenant successivement sur trois états : +V, 0 et -V (le codage Non Retour à Zéro Inversé n'utilise que 2 états). Les 0 sont codés en conservant la valeur précédemment transmise.

Le principal avantage du codage MLT3 est de diminuer fortement la fréquence nécessaire pour un débit donné grâce à l'utilisation de 3 états. Par contre, les longues séquences de 0 peuvent entraîner une perte ou un déphasage de l'horloge du récepteur. Pour éviter cela, on mettra en place un codage 4B/5B.

Exemples : **MLT3 (Fast Ethernet 100BaseTX)** et **NRZI (Fast Ethernet 100BaseFX)**

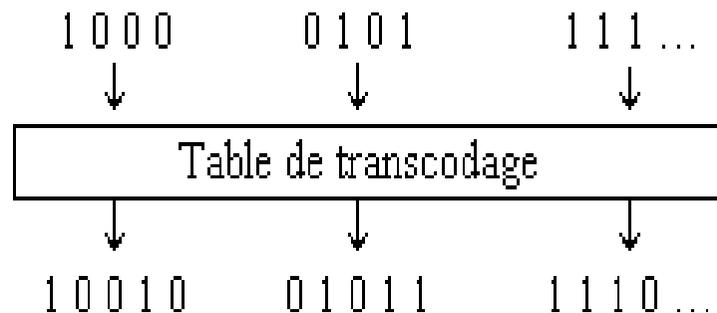


6. Transcodage nB/mB

Les longues séquences de 0 peuvent entraîner une perte ou un déphasage de l'horloge du récepteur.

Pour éviter cela, on met en place un codage 4B/5B. Il consiste à coder, à l'aide d'une table de correspondance, une série de 4 bits en 5 bits (par exemple, la séquence 0000 sera codé 01010) : il faudra toujours au minimum 2 transitions pour 5 bits et on n'aura jamais plus de 2 zéros consécutifs)

On utilise le 4B/5B pour Fast Ethernet et 8B/10B pour Gigabit Ethernet.



7. La couche Liaison

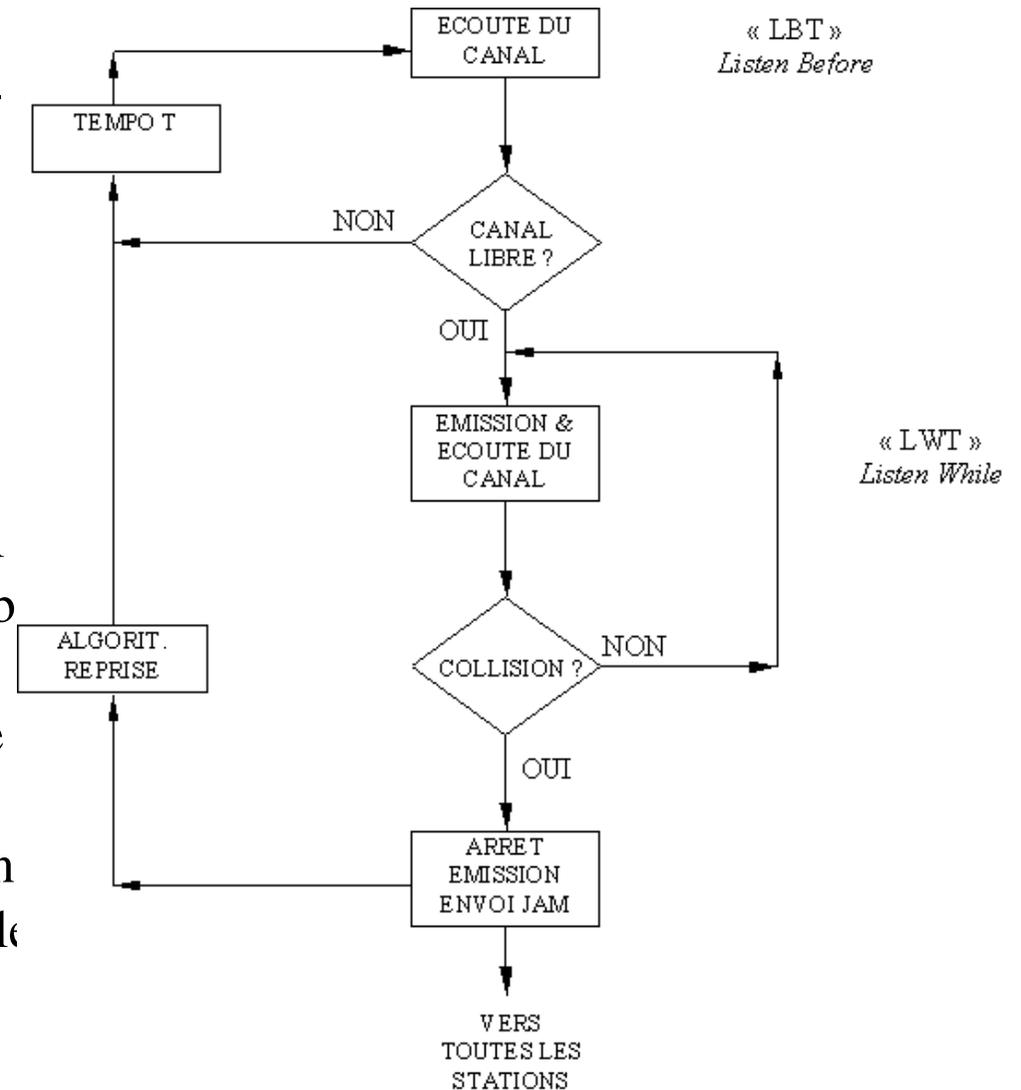
- Elle est responsable de l'acheminement sans erreurs des **trames** sur la ligne physique.
- Ces fonctions principales peuvent être :
 - établir et libérer les connexions ligne ;
 - assurer la mise en trames et la synchronisation ;
 - détecter et corriger les erreurs de transfert ;
 - gérer le contrôle de flux .
- La couche Liaison est découpée en deux sous couches appelées **MAC** (*Medium Access Control*) et **LLC** (*Logical Link Control*).
- ◆ Exemples : Ethernet 802.3 (ISO), Ethernet_II (DoD), Token Ring 802.5, PPP, LLC 802.2, *switch*, WIFI 802.11g, ...



8. Ethernet : sous-couche MAC

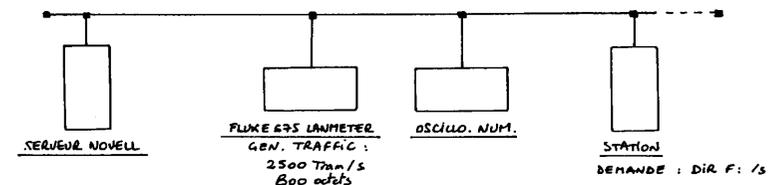
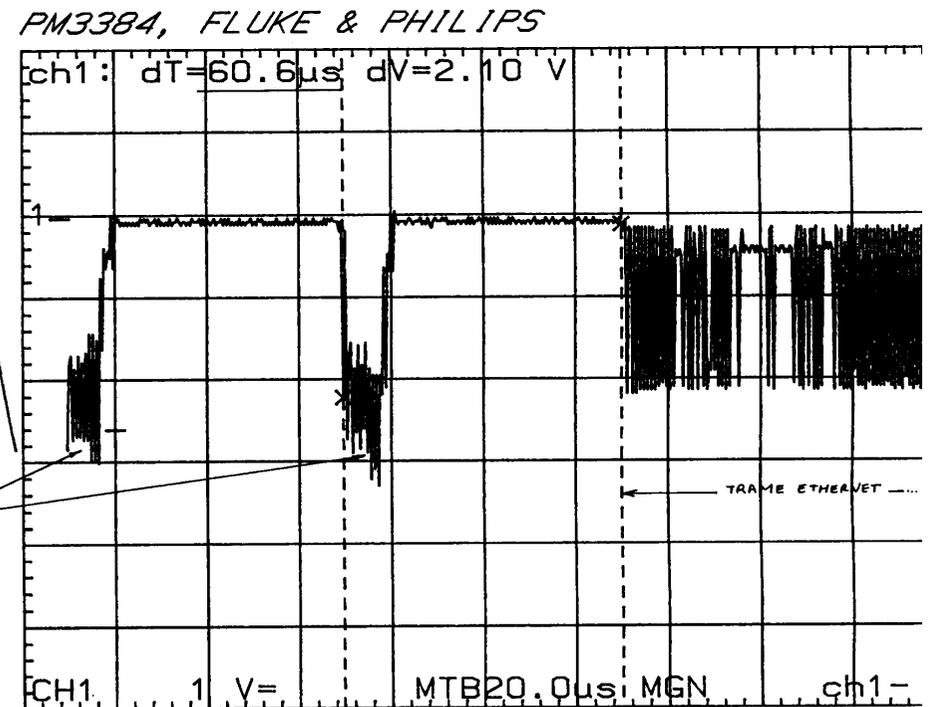
- La sous-couche **MAC** (*Medium Access Control*) d'Ethernet gère l'accès au support selon les principes de **CSMA/CD** (*Carrier Sense Multiple Access / Collision Detection*) :

Sur ce type de réseau, il est possible que 2 ou plusieurs stations détectent le support libre, décident de transmettre en même temps et ce qui provoque une collision : cette situation pose problème. Le réseau Ethernet a décidé de s'en accommoder en mettant en place un mécanisme de détection et reprise de collision (arrêt de la transmission des stations impliquées, attente d'un temps aléatoire et reprise de la procédure normale).



9. Ethernet : CSMA/CD

- La détection de collision se fait par écoute du support. Lorsque la tension sur le câble est plus élevée que la tension maximale pouvant être générée par un seul *transceiver*, une collision est détectée.
- On ne peut prévoir la présence et le nombre de collisions qui vont exister sur ce type de réseau.
- Donc on qualifie ce type de réseau de **probabiliste** (ou à accès aléatoire).
- Un réseau **déterministe** serait un réseau de type Token Ring ou bus CAN par exemple.



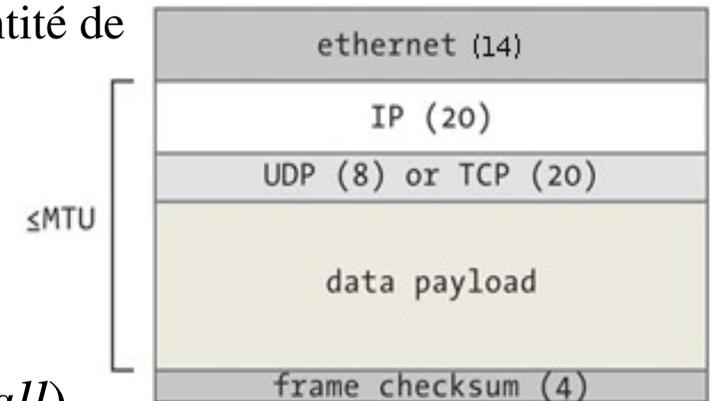
10. La couche Réseau

- Son rôle est de permettre l'acheminement de **paquets** (indépendamment les uns des autres) dans n'importe quel réseau jusqu'à destination.
- La fonction principale de cette couche est de réaliser le routage.
- Cette couche assure aussi la fragmentation qui est le découpage d'un paquet de données en paquets plus petits (fragments), pour passer à travers un lien de plus faible MTU (*Maximum Transmission Unit*). Le MTU définit la taille maximale (en octets) d'un paquet pouvant être encapsulé dans une trame de la couche inférieure.
- Dans le modèle « TCP/IP », comme aucune connexion n'est établie, les paquets peuvent arriver dans le désordre ; le contrôle de l'ordre de remise est éventuellement la tâche des couches supérieures.
- ♦ Exemples : IP (*Internet Protocol*), IPX (*Internet Packet eXchange*), routeur, ...



11. La couche Transport

- Elle est responsable du transport des **messages** complets de bout en bout (soit de processus à processus) au travers du réseau.
- Fonctions :
 - gérer les adresses de transport (les numéros de port dans le modèle DoD) ;
 - gérer le transport en établissant et libérant une connexion (TCP) ou non (UDP) ;
 - gérer les messages (segments) : segmentation des données, réassemblage des données, contrôle du séquençement (TCP) ;
 - gérer la qualité de service : acquittement et retransmission sur absence, d'acquittement, contrôle de flux (fenêtre coulissante), détection des erreurs (TCP) ;
 - gérer le MSS (*Maximum Segment Size*) qui désigne la quantité de données (en octets) qui pourra être encapsulé dans un paquet non fragmenté.
- ◆ Exemples : TCP (*Transmission Control Protocol*), UDP (*User Datagram Protocol*), SPX (*Sequenced Packet eXchange*), pare-feu (*firewall*), ...



12. Les couches Session – Présentation et Application

- **La couche Session** établit une communication entre émetteur et récepteur en assurant l'ouverture et la fermeture des sessions (des communications) entre usagers, définit les règles d'organisation et de synchronisation du dialogue entre les abonnés. Exemple : RPC (*Remote Procedure Call*), *firewall stateful*, ...
- **La couche Présentation** met en forme les informations échangées pour les rendre compatibles avec l'application destinatrice, dans le cas de dialogue entre systèmes hétérogènes. Elle peut comporter des fonctions de traduction, de compression, d'encryptage, ... etc. Exemple : XDR (*eXternal Data Representation*).
- ✓ *Remarque* : ces deux couches ne sont présentes dans le modèle TCP/IP, tout simplement parce qu'elles sont apparues inutiles ou que très rarement utiles. Le modèle OSI dépouillé de ces 2 couches ressemble fortement au modèle TCP/IP.
- **La couche Application** va apporter les services de base offerts par le réseau. Elle ne contient pas l'"application" de l'utilisateur, mais réalise l'interface pour les fonctions de communication avec les applicatifs. Exemples : FTP (*File Transfert Protocol*), HTTP (*HyperText Transfert Protocol*), etc ...



13. Notions d'API

- Les services offerts par une couche constituent l'interface à la couche supérieure.
- Cette interface est implémentée sous forme de bibliothèque de fonctions soit une **API** (*Application Program Interface*).
- Le développeur utilisera donc concrètement une interface pour programmer une application TCP/IP grâce par exemple :
 - à l'API Socket BSD sous Unix/Linux ou
 - à l'API WinSocket sous Windows
- Une *socket* est un point de communication par lequel un processus peut émettre et recevoir des informations. Ce point de communication devra être relié à une adresse IP et un numéro de port.
- Une *socket* est communément représentée comme un point d'entrée initial au niveau TRANSPORT du modèle à couches dans la pile de protocole.



14. L'interface *socket*

- En programmation, on utilisera l'interface *socket* pour réaliser des applications réseaux :

TCP (*Transmission Control Protocol*) est un protocole de transport fiable, en mode connecté (RFC 793).

UDP (*User Datagram Protocol*) est un protocole souvent décrit comme étant non-fiable, en mode non-connecté (RFC 768), mais plus rapide que TCP.

