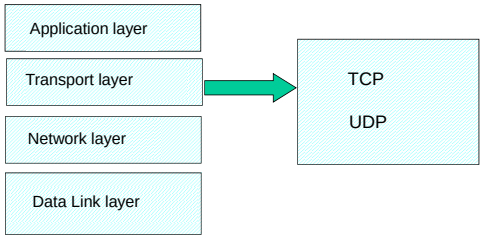
La couche Transport

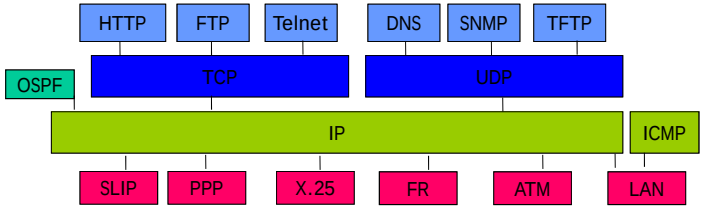


La couche transport

- Son rôle est de permettre aux entités paires sur les ordinateurs source et destination de soutenir une conversation **comme s'ils étaient reliés par une liaison point à point** : s'occuper de la qualité de transport
- Deux protocoles de bout en bout ont été définis pour cette couche :
  - TCP : Transmission Control Protocol
  - UDP : User Datagram Protocol

Identification des protocoles

- À l'instar de l'OSI avec la notion de SAP, chaque unité protocolaire de TCP-IP identifie le protocole ou l'application supérieure
  - Numéro de port
  - Identifiant de protocole
  - Ethertype



Référence de transport et socket

- La couche transport est chargée de mettre en place une connexion (avec l'ordinateur dst) dans une laison (logique) **point à point.**
- Elle ne s'occupe donc pas de trouver la route entre les deux correspondants, ce qui est du ressort de la couche Internet.
- Elle doit s'assurer **au minimum** que les données transmises à son interlocuteur sont bien dirigées vers la bonne **application**.

Principe de communication inter-applications

- Sur toute machine, il tourne toujours plusieurs programmes, appelés **processus**.
- Plusieurs processus peuvent attendre des données en provenance du reste du réseau.
  - Par exemple, il peut tourner à la fois un serveur FTP et un serveur HTTP (Web).
- Un processus qui tourne sur une machine et qui attend des données en provenance du réseau s'appelle :
  - un **démon** pour Unix (traduction anglaise du mot daemon)
  - ou **service** pour Windows.
  - **Nous utiliserons définitivement le mot démon.**
- Nécessité d'avoir un Lien entre processus distants leur permettant de :
  - communiquer entre eux,
  - identifier les éventuels flux simultanés vers plusieurs applications, ou différents niveaux applicatifs ou système

Principe de communication inter-applications

- Ce démon « écoute » le réseau en attendant que quelqu'un s'adresse à lui.
- La couche transport doit donc indiquer **à quel démon elle doit envoyer les données**.
- La « localisation » du démon dans le système s'appelle un **port** et il est identifié par un **numéro**.
- **A chaque démon présent dans un système donné correspond un numéro de port,** qu'il faut impérativement mentionner pour avoir une chance de joindre ce démon.
- La notion de ports autorise le multiplexage des connexions sur une même machine
- C'est la couche transport qui attribue le numéro de port à une connexion

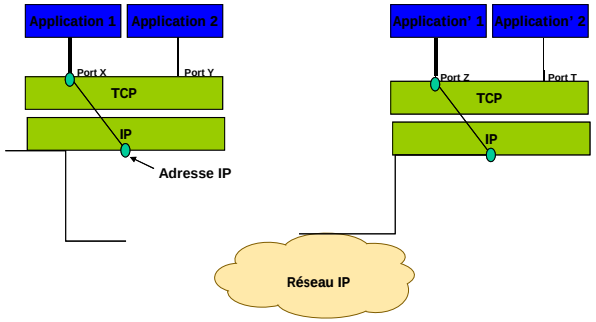
Socket / ports

- **Définition :**
  - La combinaison adresse IP/numéro de port est parfois appelée une **socket**.
  - **Une socket décrit une connexion entre deux machines.**
  - Certains numéros de port sont fixes et toujours attribués à la même application. Ce sont des ports « bien connus » (**well known ports).** Les well known ports sont gérés par l'IANA.
  - **Exemples :** FTP : 21 ; Telnet : 23 ; SMTP (mail) : 25 ; HTTP : 80 ; POP (récupération courrier) : 110

Socket / ports

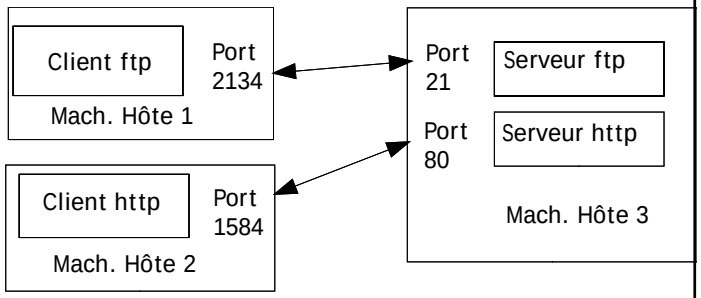
- En général, les **processus serveurs** sont identifiés par leur numéro de port bien connu **compris entre 1 et 1023**.
  - Une exception célèbre est constituée par le serveur X-Windows (responsable de l'affichage graphique sur les machines Unix) qui écoute par exemple sur le port 6000.
- La partie **client** possède un numéro de port **éphémère**. **Ces numéros de port sont compris entre 1024 et 5000**. et **changent à chaque connexion**.
- La liste des numéros de port est répertoriée dans le fichier **/etc/services** sous Unix.

Principe de communication inter-applications



Pour établir la communication, l 'émetteur doit fournir
- ❑ L'identificateur du système distant 'adresse réseau'
- ❑ L'identificateur du protocole utilisé
- ❑ L'identificateur du processus d'extrémité 'numéro de port'

Schéma d'une connexion client / serveur



Données de la connexion

- ❑ De manière symétrique, pour se connecter à un **serveur** donné, il faut utiliser un **client**.
- ❑ Le client va ouvrir une connexion sur l'ordinateur client en utilisant également un **numéro de port.**
- ❑ Pour pouvoir établir une connexion client/serveur, il faut donc **au moins** 4 paramètres :
  - L'adresse IP du **client**
  - Le numéro de port du **client**
  - **L'adresse IP du serveur**
  - Le numéro de port du **serveur**
- ❑ L'association : {protocole, @IP destination, port source, @IP source} est désigné sous le terme **SOCKET**

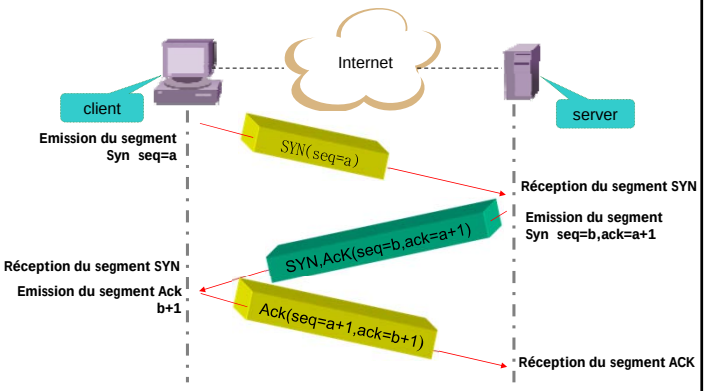
La couche Transport : TCP

- ❑ Protocole TCP :
  - Protocole responsable de l'adressage de niveau 4
  - Protocole de transport en mode connecté qui résout tous les problèmes résiduels du déplacement de l'information
  - Il garantit donc plusieurs choses :
    1. **Le séquencement** des paquets.
    2. **Le contrôle de pertes** : la destination devra recevoir tous les paquets envoyés.
    3. **Le contrôle des doublons** (c'est à dire que le destinataire ne doit pas recevoir deux fois le même paquet).
    4. **Le contrôle de flux** afin d'éviter la saturation et la congestion du destinataire.
    5. **La gestion d'un circuit virtuel** qui en fait un protocole orienté connexion.
- ❑ *TCP préférera couper une connexion plutôt que de violer une des trois premières garanties.*
  - *La contrepartie à ces garanties est un temps d'acheminement nettement plus long qu'UDP.*

Etablissement de la connexion

- Cette opération se passe en trois temps : 3 ways handshaking
  1. La machine émettrice émet un premier paquet avec le flag SYN à 1 (activé) pour indiquer qu'elle souhaite se synchroniser avec la machine réceptrice. Elle indique également son numéro de séquence courant Seq_{emet} = x
  2. La machine **réceptrice** accuse réception du paquet (ACK x+1), elle envoie à son tour son numéro de séquence courant Seq_{emet} = y. Le flag SYN est toujours à 1.
  3. L'émetteur accuse réception du paquet (ACK y+1). Les deux machines connaissent maintenant leur numéros de séquence respectifs et peuvent se caler dessus. Le flag SYN est à 0 pour cette troisième étape. Le numéro de séquence de l'émetteur est maintenant x+1.

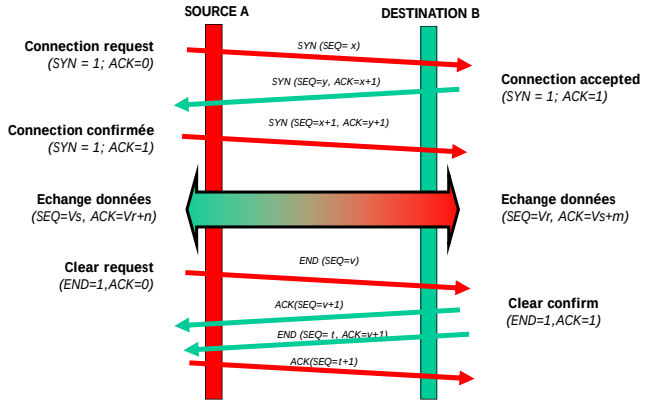
TCP Connexion



TCP : déconnexion

- La connexion peut se **relâcher** de deux façons :
  - **Déconnexion normale**
  - **Clotûre abrupte :** Réinitialisation
    - L'émetteur envoie un paquet avec le Flag RST à 1. Le récepteur n'a pas besoin d'acquitter.
    - On voit apparaître un message du type « connexion reset by peer ».
      - Toutes les ressources telles que les mémoires tampons sont libérées

Gestion d 'une connexion TCP



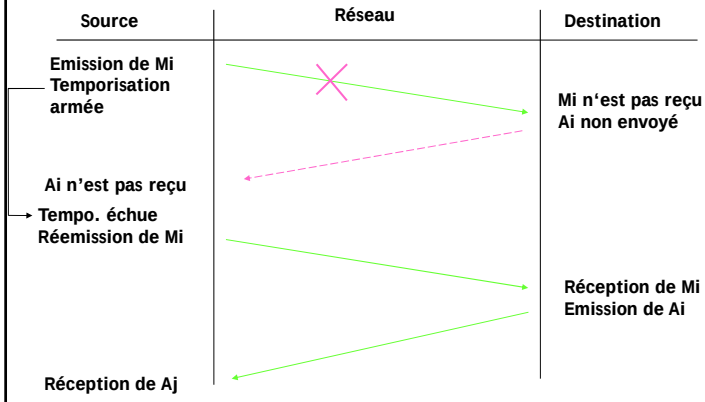
TCP : Segmentation et séquencement

- Segmentation, contrôle de flux
  - Les données transmises à TCP constituent un flot d'octets de longueur variable.
  - TCP divise ce flot de données en segments en utilisant un mécanisme de fenêtrage.
  - Un segment est émis dans un datagramme IP.
- Séquencement
  - Chaque paquet est **numéroté** avant d'être envoyé.
  - La réception des paquets doit se faire **dans l'ordre** pour etre considérée comme valide.
  - Cette contrainte impose à l'émetteur et au récepteur de **synchroniser** leurs numéros de séquence.

TCP : Accusé réception

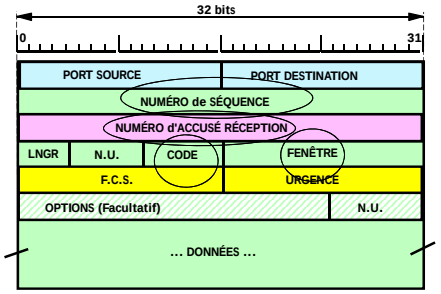
- Contrairement à UDP, TCP garantit l'arrivée des messages, c'est à dire qu'en cas de perte, les deux extrémités sont prévenues.
  - Ce concept repose sur les techniques d'acquittement de message :
    - lorsqu'une source S émet un message Mi vers une destination D,
    - S attend un acquittement Ai de D avant d'émettre le message suivant Mi+1.
    - Si l'acquittement Ai ne parvient pas à S,
      - S considère au bout d'un certain temps que le message est perdu et réémet Mi :
- Le récepteur doit donc obligatoirement **accuser réception des paquets reçus**
  - Lorsque la **destination** reçoit correctement un paquet, elle envoie un **acquittement** à la source
  - on appelle cela un ACK
- **Cet acquittement porte le numéro du paquet que la destination s'attend à recevoir.**

TCP : Acquittements



Le protocole T.C.P.

Protocole de transport fiable en mode connecté



Le segment T.C.P.

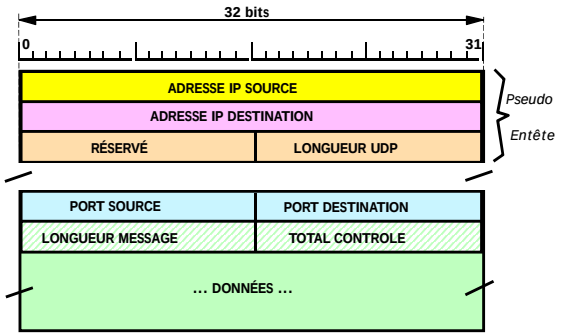
Protocole de transport TCP

- TCP est
  - Un protocole en mode connecté de bout en bout
  - Avec contrôle de flux et de congestion
  - Assure l'intégrité des données (checksum)
  - Assure la transmission (détection des pertes et retransmission)
  - Une connexion TCP est identifiée de manière unique par: adresse IP source, adresse IP destination, port TCP source, port TCP destination
- TCP est utilisé pour sa robustesse:
  - Émulation de terminal, transfert de fichier, web, e-mail…
  - Client serveur

La couche Transport : UDP

- Protocole UDP :
  - Protocole responsable de l'adressage de niveau 4
  - Protocole de transport en mode non connecté qui vise la rapidité et non pas la résolution des problèmes résiduels du déplacement de l'information
    - Les protocoles de niveau supérieurs doivent se charger de la résolution des problèmes
      - Ex : TFTP va corriger les erreurs
- *UDP est un protocole simple de transport qui n'offre aucune garantie du point de vue de la fiabilité. UDP travaille sans connexion.*

Le PROTOCOLE UDP



Message UDP = DATAGRAMME UDP

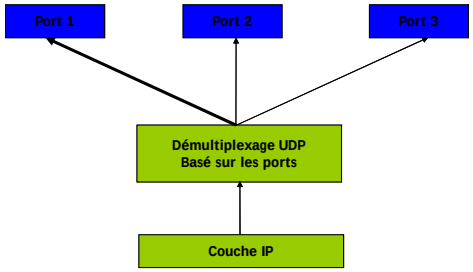
Fonctionnement du protocole UDP

- **UDP reçoit des messages en provenance de diverses applications et les passe à la couche IP.**
- A la réception, **UDP récupère des datagrammes destinés à diverses applications et se charge de les distribuer**.
- Chaque programme a un numéro de port local et un numéro de port distant **avant** de pouvoir envoyer des informations au protocole UDP.
- Lorsque UDP reçoit un datagramme, il vérifie le numéro de port dst. Un message d'erreur peut-être envoyé à la src du datagr par l'intermédiaire du protocole appelé ICMP (à venir).
- **UDP multiplexe** et **démultiplexe** les paquets en provenance et à destination des programmes du système d'exploitation.

UDP : Multiplexage

- UDP reçoit des messages en provenance de diverses applications et les passe à la couche IP.
- A la réception, UDP récupère des datagrammes destinés à diverses applications et se charge de les distribuer.
- UDP multiplexe et démultiplexe les datagrammes en sélectionnant les numéros de ports :
  - une application obtient un numéro de port de la machine locale; tout message émis via ce port ➔ champ PORT SOURCE du datagramme UDP = ce numéro de port,
  - une application connaît (ou obtient) un numéro de port distant afin de communiquer avec le service désiré.
- Lorsque UDP reçoit un datagramme
  - il vérifie que celui-ci est actif et ouvert (associé à une application)
  - le délivre à l'application responsable (mise en queue)
- si ce n'est pas le cas, il émet un message dit ICMP *port unreachable*, et détruit le datagramme.

Fonctionnement du protocole : dé-multiplexage



- MULTIPLEXAGE : UDP vers IP
- DEMULTIPLEXAGE : IP vers UDP

Protocole de transport UDP

- **UDP** (User Datagram Protocol) est un protocole **non fiable sans connexion**
- **UDP est** destiné aux applications qui ne veulent pas du séquencement et du contrôle de flux (de TCP) parce qu'elle souhaitent utiliser leurs propres moyens.
  - Applications du type client/serveur ou demande/réponse
  - Applications de **transmission du son et de l'image, où l'aspect temps réel est plus important que la fiabilité.**
- La couche transport transmet des **paquets** à la couche inférieure.

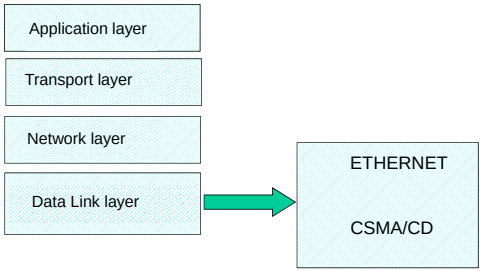
Protocole de transport UDP

- UDP est
  - Un protocole en mode non connecté
  - Sans contrôle de flux et de congestion
  - Peut vérifier l'intégrité des données (checksum)
  - N'apporte pas de garantie de transmission à l'application
  - Un flux UDP est identifiée de manière unique par: adresse IP source, adresse IP destination, port UDP source, port UDP destination
- UPD est un protocole de transport proche de l'IP:
  - Flux temps réel: voix, video…
  - Gestion de réseau SNMP

TCP versus UDP

UDP	TCP
Service sans connexion ; aucune connexion n'est établie entre les hôtes.	Service orienté connexion ; une connexion est établie entre les hôtes.
UDP ne garantit ou n'accuse pas réception des données de livraison ou de séquence.	TCP garantit la livraison à l'aide des accusés de réception et la livraison de données en séquence
Les programmes qui utilisent le protocole UDP doivent fournir une stabilité nécessaire pour le transport des données	Les programmes qui utilisent TCP sont assurés de la fiabilité du transport des données.
UDP est rapide, présente des exigences de faible délai et peut facilement prendre en charge des communications point à point et point à multipoint.	TCP est plus lent, présente des exigences de délai plus élevé et ne peut prendre en charge facilement que les communications point à point.

La couche liaison



La couche Liaison

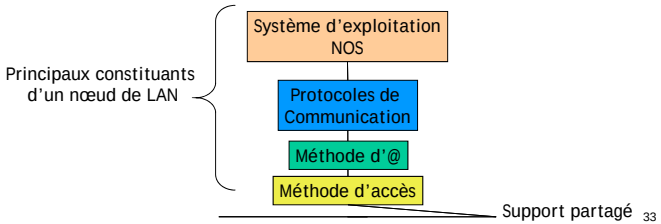
- Rôle : Assurer une transmission des trames sur une liaison point à point => pas de routage à ce niveau.
  - Sécuriser les échanges
  - Gérer l'accès au support de transmission
  - 1er travail, transformer un flot brut de bits sans cohérence apparente en trames, sur lesquelles on pourra travailler efficacement (en particulier, faire une détection d'erreurs ou de pertes, synchroniser le dialogue etc.) :
- Au niveau Internet
  - Utilisation d'un protocole spécifique à la couche 2 utilisé
    - Couche 2 point à point
    - Et couche 2 point – Multipoint

La couche Liaison

- Couche 2 point à point :
  - Transport des données sur des liaisons série
    - Sécuriser les échanges : délimitation des blocs de données
      - » HDLC
      - » SLIP
      - » PPP
- Et couche 2 point – Multipoint
  - Sécuriser les échanges
  - Gérer l'accès au support de transmission
    - » Couche Ethernet

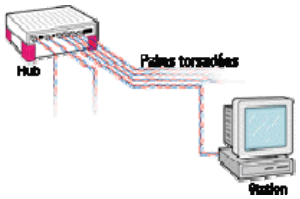
Principes d'Ethernet

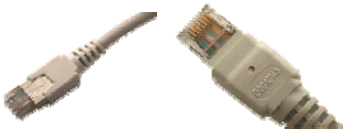
- Support de transmission
  - brin = segment = bus = câble coaxial
  - pas de boucle
  - pas de sens de circulation
- Chaque carte Ethernet possède une adresse unique au niveau mondial (adresse MAC)
- Pas de multiplexage en fréquence ⇒ une seule trame à un instant donné
  - Réception par tous les transceivers du réseau d'une trame émise par une station



Composantes LAN

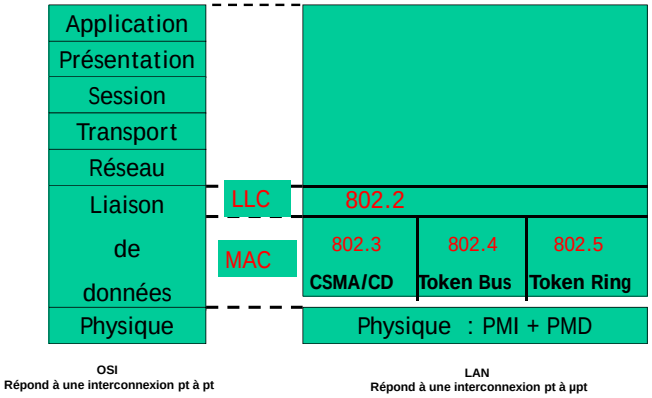
Les hubs





Un connecteur RJ-45

Niveau 1 et 2 des réseaux locaux



Niveau 2 des réseaux locaux

La couche liaison de donnée des réseaux locaux est divisée en deux sous-couches:

- La sous couche LLC (Logical Link control)
- La sous couche MAC (Medium Access Control)

La sous-couche MAC à fait l'objet de trois normes:

- 802.3 : Réseau en bus CSMA/CD
- 802.4 : Token Bus
- 802.5 : Token Ring

Logical Link Control

- But : masquer à la couche réseau les différents types de LAN et donc les hétérogénéités de technologies susceptibles d'exister dans les couches MAC et physiques
  - La couche réseau doit pouvoir se poser sur la couche liaison et bénéficier de ses services sans avoir à se soucier du détail de l'implémentation du service, comme l'exige la structure en couche
- Le rôle de cette sous couche LLC est de gérer les communications avec les couches supérieures.
  - Elle prend en charge :
    - Découpage du flot de bits en trames
    - Contrôle d'erreurs (par CRC le plus souvent) et de pertes
    - Contrôle de flux
    - Eventuellement le séquencement.
- Ses spécification sont données dans le standard IEEE 802.2 et sont reprises dans la norme internationale IS 8802-2 de l'ISO.

Accès au Suport

- Problématique :
  - Soit un amphithéâtre bourré de bavards et qui en plus voudraient que tout le monde entende ce qu'ils ont à dire.
  - Proposer un moyen viable d'attribuer la parole sans aboutir à une anarchie ?
- Il existe deux grandes philosophies pour attribuer la parole, que ce soit dans les amphis, les réseaux ou ailleurs :
  - L 'attribution aléatoire ou à contention :
    - Techniques CSMA : écoute de la porteuse
      - CA (prévention de la collision) : Réseau d'apple partage d'imprimante pour de faibles débits 230,4 kbps
      - CD (détection de la collision) : normalisée par IEEE 802.3 et 8802.3 pour réseau Ethernet et représente 90% des techniques utilisées aujourd'hui
  - et l 'attribution déterministe ou à réservation : dérivées du polling/selecting avec réservation des ressources par échange de jeton (Token) circulant de machine en machine

Adressage Ethernet

- **Diffusion :** broadcast
  - toutes les machines reçoivent, seule la machine destinataire traite l 'info et éventuellement répond
- Chaque station reçoit toutes les données
  - Emetteur d'une trame ?
  - Destinataire d'une trame ?
- **Identification :** IP ? Non.
  - L'IP sert au routage, i.e. niveau couche réseau, Ici on reste au niveau liaison,
    - pas de routage, diffusion => adressage MAC
- Ajout d'un bordereau d'envoi
  - Entête de trame
  - Adresse destination
  - Adresse source
- Notion de trame structurée
  - On appellera cette adresse : l 'adresse Ethernet
- **= >** Lien entre les adresses IP et MAC :

le protocole ARP (Address Resolution Protocol).

Forme de l'adresse Ethernet

- **IEEE** propose deux formats : long (48 bits) plus utilisé et court (16 bits)
- **Forme :**
  - six nombres codés en **hexadécimal** séparés par " : ".
- *Exemple :* 00 : D3 : FF : 17 :1E : 03
- Caractéristiques :
  - 48 bits
  - *Unicité globale : non obligatoire* *(uniquement sur un réseau)*
- 3 premiers octets indiquent le constructeur
- Comme pour IP des adresses spécifiques

Premiers octets : le constructeur

- **Les trois premiers octets de l'adresse indiquent à quel constructeur on a à faire :**
  - 00:00:0C : XX:XX:XX : Cisco
  - 08:00:20 : XX:XX:XX : Sun
  - 08:00:09 : XX:XX:XX : HP
  - 08:00:14 : XX:XX:XX : Excelan
  - AA:00:04 : XX:XX:XX : DEC
  - 00:00:0E : XX:XX:XX : Fujitsu
  - 00:00:0F : XX:XX:XX : NEXT
  - 00:00:10 : XX:XX:XX :SYTEK
  - 00:00:15: XX:XX:XX : DataPoint Co.
  - 00:00:1B: XX:XX:XX : Novell

Adresses Ethernet spécifiques

- Comme pour les adresses IP, les adresses Ethernet peuvent être:
  - Unicast : adresse de station, Adresse individuelle
  - Broadcast : adresse de diffusion généralisée vers toutes les machines.
    - L'adresse de diffusion dans les réseaux Ethernet, comme pour l'adressage IP, est constituée entièrement de 1 :
      - **FF : FF : FF : FF : FF : FF**
    - L'usage abusif de broadcast peut être très pénalisant surtout pour les liens WANs
      - Filtrage au niveau du routeur
  - Multicast : adresse d'un groupe
      - Les plages : 01-00-5E-00-00-00 à 01-00-5E-7F-FF-FF

Trame de données

Norme 802.3

Préambule	SFD	@ Destination	@ Source	Long	Données	Bourrage L<46 oct	CRC
7	1	6	6	2	de 46 à 1500		4 oct

Norme Ethernet

Préambule	SFD	@ Destination	@ Source	Type	Données	Bourrage L<46 oct	CRC
7	1	6	6	2	de 46 à 1500		4 oct

- Débit d'émission / réception : 10 Mb/s
  - 10 bits par µs
- Longueur des trames :
  - 26 octets réservés au protocole
  - Longueur minimale : **72 octets (46 utiles)**
  - Longueur maximale : **1526 octets (1500 utiles)**

La norme 802.3 et CSMA/CD

CSMA/CD: Carrier Sense Multiple Access/Collision Detected

CSMA/CD est une technique basée sur le principe d'écoute et de détection de collision

Le principe d'accès au support est la compétition: un émetteur utilise la voie dès qu'il est prêt à émettre, Il ignore donc les autres émetteurs

=> Risque de collision

Pour limiter le nombre de collision, cette technique utilise le principe d'écoute, c'est à dire un émetteur n'émet que s'il n 'y a pas de transmission en court

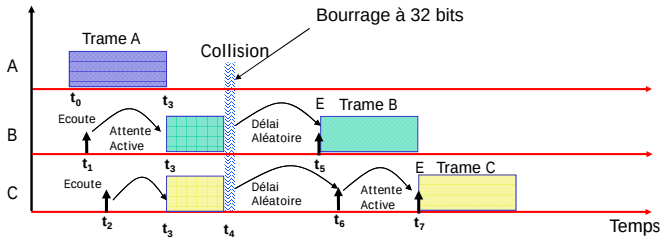
Accès CSMA/CD

- Le protocole Ethernet se base sur la méthode d'accès appelée CSMA/CD (Carrier Sense Multiple Access / Collision Detection) développé par l'Université d'Hawai et convient particulièrement aux topologies en bus.
- Nous allons procéder à une petite analogie avec le monde téléphonique :
  - Lorsque vous désirez entrer en communication avec un interlocuteur :
    1. vous composez le N° de téléphone désiré et espérez que la ligne soit libre;
    2. si vous entendez le signal 'occupé', vous essayez un moment plus tard.
  - CSMA/CD se base sur le même principe:
    1. Chaque nœud du réseau est à l'écoute du réseau (si un paquet lui est destiné, il le lit),
    2. et lorsqu'un équipement désire émettre un paquet, il ne le fait que si personne d'autre n'est train de transmettre ses propres paquets.
    3. Si le réseau est 'occupé', il attend un moment (calculé de façon aléatoire) et essaye à nouveau.
    - .

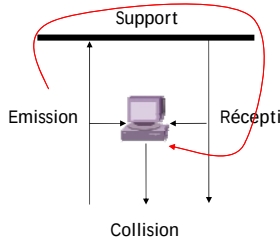
Accès CSMA/CD

- Compte tenu des caractéristiques physiques d'un réseau :
  - un paquet (paquet 1) peut être émis par un nœud mais pas encore détectable par l'équipement désirant émettre;
  - celui- ci transmet son paquet (paquet 2) à l'instant où le 'paquet 1' est détectable :
    - La station 2 compare paquet 1 et paquet 2; puisqu'ils sont différents
      - il en résulte une collision
- En cas de collision, les nœuds impliqués :
  - Emettent un signal pour signaler de façon certaine l'événement à l'ensemble du réseau,
  - Arrêtent leurs émissions
  - Puis essayent d'émettre à nouveau après un délai aléatoire r :
    - utilisation de l'algorithme du BEB (Binary Exponential Backoff).
  - Réécoutent la disponibilité du support puis émission si ce dernier est disponible
- C'est la couche MAC qui s'occupe de ces fonctions pour éviter de remonter jusqu'à LLC et pénaliser les performances du réseau

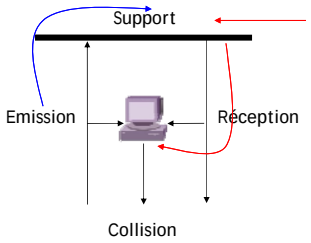
Principe CSMA/CD



Collision



Le signal reçu est identique au Signal émis, il n'y a pas de collision



Le signal reçu est différent du Signal émis, il y a de collision

Collision : si la station envoie et en même temps elle reçoit quelque chose de différent de ce qu'elle envoie

Fenêtre de collision

- Pour éviter les collisions et pouvoir les détecter il faut
  - Eviter les trames trop courtes
  - Limiter la longueur du réseau
- La fenêtre de collision est :
  - Le temps minimal t_{min} pendant lequel une station doit émettre pour détecter la collision la plus tardive que son message est susceptible de subir
  - Si la station émet durant un temps inférieur à t_{min}; elle ne pourra jamais détecter la collision
  - En quelque sorte, c'est comme si on imposait une taille minimale de trame

Collisions discrètes !

Dans l'exemple:

1. DTE2 voit la collision
2. DTE1 ne voit rien
3. DTE2 ré-émet sa trame, puisque collision
4. DTE1 en reçoit une deuxième copie !!!

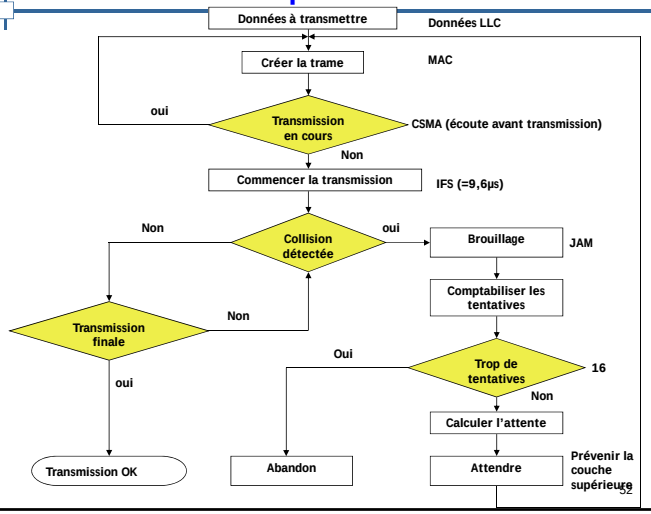
➔ Eviter les collisions discrètes

- Collision !
- DTE2 voit la collision
- DTE1 ne voit rien !

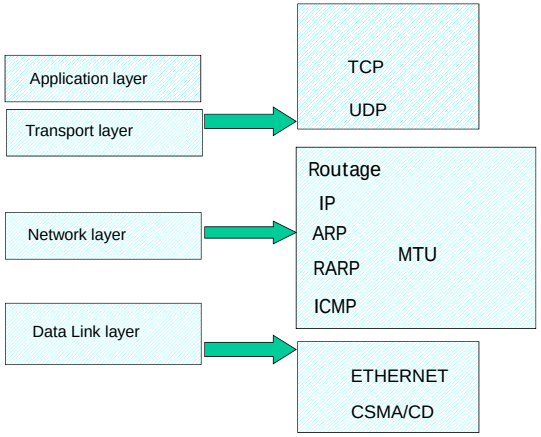
Collisions

- Chaque station doit occuper le réseau un minimum de temps appelé : TIME SLOT ou TRANCHE DE CANAL ou PERIODE DE VULNERABILITE DU MESSAGE
- Time Slot = Temps aller/retour sur la plus grande distance possible (2500 m) = Round Trip Delay
- Pour que la détection de collision soit réalisée, DTE2 doit occuper le canal par une séquence de brouillage, de taille 32 bits à 1, pendant la durée dite JAM INTERVAL (3,2 µs)

Principe CSMA/CD



Architecture TCP-IP



Références Bibliographiques

Ouvrages

- Réseaux, cours et exercices, Andrew Tanenbaum; Dunod
- Réseaux et Télécoms : Claude Servin, Dunod 2003
- Les réseaux : G.Pujolle, 3ème édition, Eyrolles
- TCP/IP, Architecture, protocoles, applications, Douglas Comer; Dunod
- Microsoft Press : Kit de Formation TCP-IP
- Cisco TCP/IP Routing Professional Reference : C.Lewis; Cisco Press
- IP Addressing and Subnetting for New Users : Cisco White paper
- Internet Routing Architectures, Second Edition; Sam Halabi; Cisco Press

Liens Web

- Site officiel IETF : www.IETF.org
- Pascal Nicolas : Support de Cours, Uni d'Angers. www.info.univ-angers.fr/pub/pn
- Olivier Hoarau : introduction aux RL et étendus, www.funix.free.fr

PPTs

- Protocole IP : P.Gautier; ISEN Toulon
- Protocoles UDP-TCP : P.Gautier; ISEN Toulon
- Modèles OSI : P.Gautier; ISEN Toulon
- Cours des réseaux Informatiques : M.Rziza, FSR