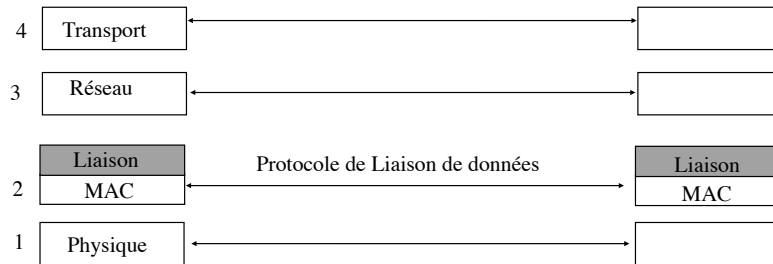


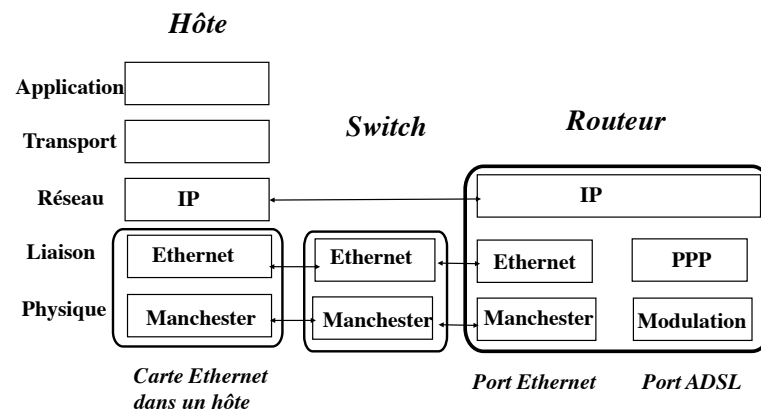
La couche liaison de données



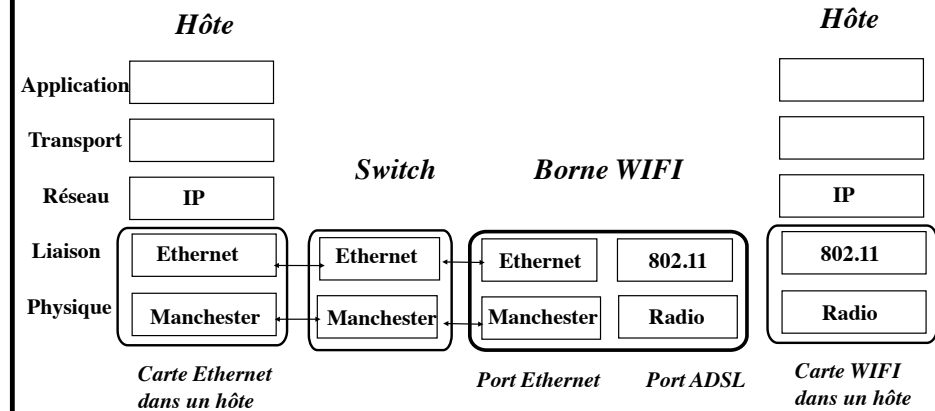
Services fournis à la couche réseau

- Découpage en trame
- Un transfert fiable: La détection/correction des erreurs
- Contrôle de flux et récupération d'erreur
- Accès multiples à un support (fait par la sous couche MAC : Medium Access Control)
 - Un exemple : le protocole Ethernet
- Une grande partie de ces problématiques est en fait souvent réalisée dans la couche transport (en particulier pour les réseaux locaux)
- Dans Ethernet la couche Liaison se résume à la couche MAC , à la délimitation des trames et à la détection des erreurs
 - C'est un service sans connexion, ni acquittement
- Le reste (contrôle de flux et récupération d'erreur) est fait dans TCP

Equipements et couches protocolaires



Equipements «Niveau 2»



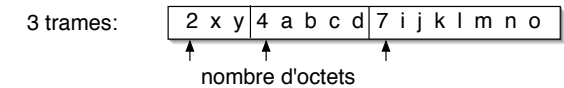
Découpage en morceaux des données

- Paquet (ou trame pour le niveau Liaison de donnée)
- But du découpage en paquet
 - » Unité pour la détection/correction d'erreur
 - » Efficacité pour la récupération d'erreur: on ne renvoie que le paquet perdu ou erroné
 - » Permet de répartir la bande passante entre les différents utilisateurs: pas de monopolisation du réseau trop longue par un seul utilisateur
- Découpage effectué éventuellement par les protocoles des couches supérieures
- Exemple Ethernet 1500 octets maximum dans une trame

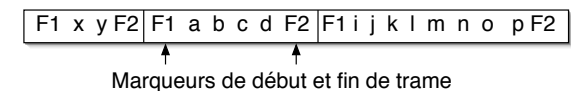
Notion de trame

- Techniques de découpage en trame

1- Compter les caractères



2- Utiliser des marqueurs de début et de fin de trame

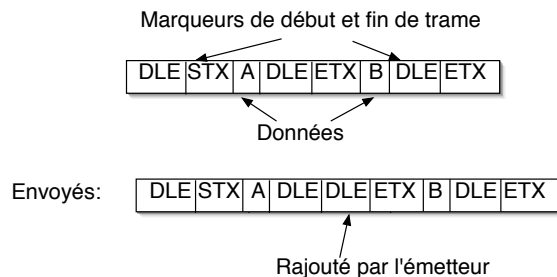


3- Emettre un silence entre les trames (au niveau physique)

Problème de Transparence (1)

- **Confusion des délimiteurs de trame et des données**
- Les données peuvent contenir les délimiteurs de trames
- Exemple 1 : caractères de délimitation

Les caractères DLE STX et DLE ETX délimite le début et la fin des trames. Pour assurer la transparence des données l'émetteur rajoute un DLE devant tout DLE des données



Transparence (2)

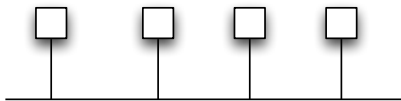
- **Exemple 2 : Utilisation de fanions**
 - HDLC est une procédure de liaison de données et qui utilise comme délimiteur de début et fin de trame la séquence de bits : 01111110
 - Pour assurer la transparence l'émetteur ajoute systématiquement un 0 après toute séquence de 5 bits à 1 rencontrée dans la trame
 - Le récepteur effectue l'opération inverse, c'est à dire qu'il retire le 0 qui suit chaque séquence de 5 bits à 1.
- **Méthodes hybrides:**
 - Certains protocoles combinent plusieurs de ces méthodes pour limiter les risques de confusion en cas d'erreur. Que se passe-t-il si dans la 1ère méthode le nombre d'octets est erroné à l'arrivée ?
 - **Exemple :** Ethernet utilise un marqueur de début de trame et arrête le codage Manchester à la fin de trame (silence inter-trame de 9,6 microseconde pour le 10 mégabit/s)

La sous couche MAC: Le contrôle d'accès multiple au canal

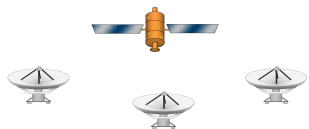
- Réseaux particuliers à diffusion:

- Intéressant pour leur faible coût de déploiement

- bus



- Satellite radio



- Wifi radio



La sous couche MAC: Le contrôle d'accès multiple au canal

- Problème :

- Réseau à diffusion implique un support unique pour n émetteurs/ récepteurs
- Il existe différentes solutions pour réaliser ces accès multiples:
 - » Par partage statique du support
 - Le support est “divisé” de façon statique, soit dans le temps, soit en fréquence (voir multiplexages dans le cours Couche Physique)
 - » Par partage dynamique du support
 - Par accès aléatoire :
 - “On émet quand on veut”
 - Par accès séquentiel
 - “On émet à tour de rôle”

Intérêt des méthodes à accès statiques

- Monopolisation d'une partie de la bande passante
- Adaptées pour:
 - Nombre de stations actives réduites et fixes
 - Trafics prévisibles et peu variables
- D'où l'intérêt des méthodes à accès dynamique: la couche MAC à proprement parler des réseaux locaux à diffusion
 - où le nombre de stations générant du trafic varie dans le temps
 - les trafics générés sont sporadiques

- L'allocation du support est fait en fonction des demandes

Une émetteur «seul» va pouvoir utiliser toute la bande passante

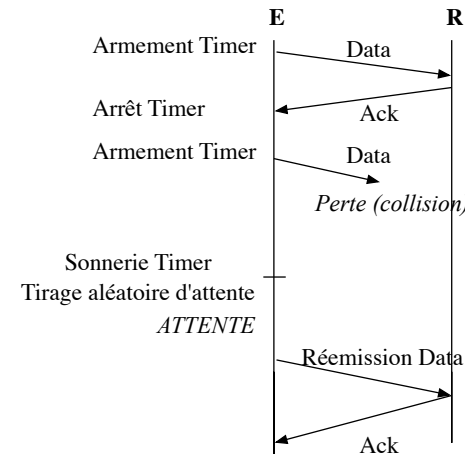
Accès multiple: solutions par accès aléatoires

- Accès du support à la demande
- Une seule fréquence, possibilités de collisions si plusieurs émissions en même temps
- Ethernet et WiFi (Wireless Fidelity), utilise ce principe
- Premiers protocoles: ALOHA (bonjour en hawaïen)
 - Réseau radio entre îles hawaïennes puis Ethernet en 73 et normalisation en 80

Historique: ALOHA PUR (1972)

- Une station voulant émettre un paquet d'information sur le réseau, commence immédiatement à le transmettre. Évidemment, si deux émetteurs ou plus émettent en même temps, il y a collision et les émetteurs devront re-émettre leur paquet ultérieurement
- La ré-émission a lieu si un acquittement n'est pas reçu au bout d'un temps fixe
- Cette re-émission a lieu immédiatement avec une probabilité p . Sinon l'émetteur attend la durée d'émission d'une trame avant de re-émettre

Exemple ALOHA



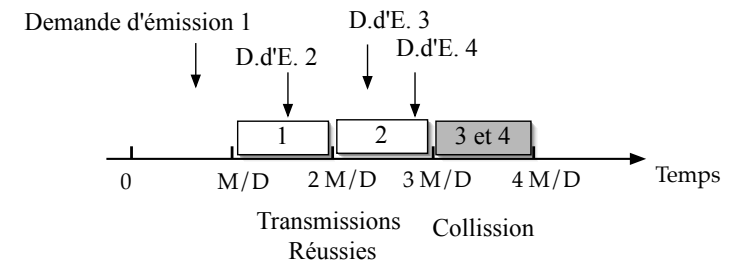
Algorithme ALOHA

Suivant évènement

- Si demande d'émission :
 - Emettre paquet
 - Armer timer
 - Si réception Acquittement
 - Arrêter Timer
 - Si Sonnerie Timer
 - Attendre un temps aléatoire
 - Emettre paquet (mémoire dans un buffer)
- On peut montrer que l'on arrive à une efficacité de 18% du débit total dans le meilleur des cas

Aloha discrétisé

- Le temps est discrétisé en intervalles de durée fixe M/D
 - M : nombre de bits des messages, D : débit
 - M/D = temps d'émission d'une trame
- Les horloges de toutes les stations sont synchronisées
- Les messages ne peuvent être transmis qu'en début d'intervalle
- L'efficacité est ainsi doublée (0,37)



Vers un protocole avec détection d'activité et de collision

- **Pour augmenter les performances:**

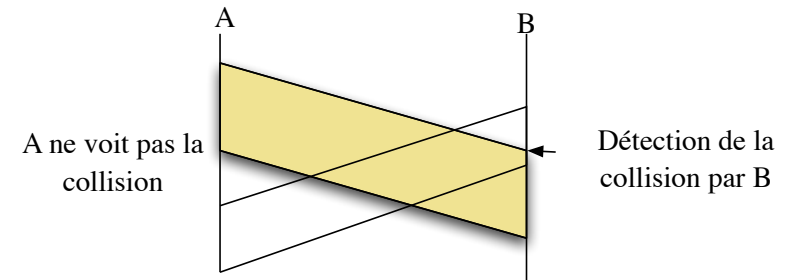
- CSMA (Carrier Sense Multiple Access) : on sait détecter si le support est libre, on émet bien sûr que si il est libre
- CD (Collision detection): une machine "écoute" le support pendant qu'elle émet, s'il y a une différence par rapport à ce qu'elle émet, c'est qu'il y a une collision

» La transmission est alors tout de suite arrêtée

- **Il y a encore des collisions !**

- A cause du temps de propagation sur le support qui n'est pas nul

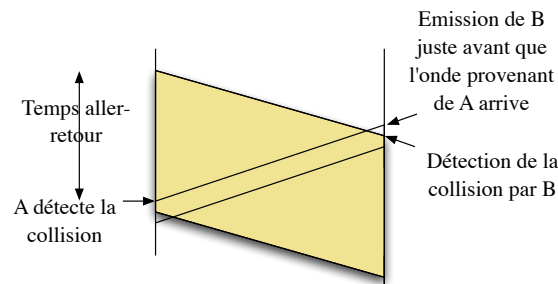
Détection des collisions



- A ne voit pas la collision car l'onde provenant de B n'arrive en A qu'après qu'il ait fini d'émettre.
- Comment être sûr de détecter une collision si elle doit avoir lieu pendant l'émission d'un paquet ?
- Cela permettrait d'éviter les acquittements et d'augmenter l'efficacité

Détection des collisions

– Pire cas:



- $T=2* \tau$ est appelé la tranche canal (avec τ = temps de propagation du signal d'un bout à l'autre du câble)
- une station qui veut détecter les collisions doit émettre pendant une durée au moins égale à $2* \tau$
- Cela induit pour un débit fixe **une taille minimale de paquet**

Stratégie de résolution des collisions

- Une station ayant détectée une collision se met en attente pendant une durée aléatoire X dont l'intervalle de définition double à chaque nouvelle collision jusqu'à une valeur maximum au-delà de laquelle elle abandonne
- On limite ainsi le temps moyen d'attente en diminuant la probabilité qu'il y ait une nouvelle collision
- Sans connaître le nombre de station en concurrence au départ ce protocole s'adapte à celui ci
- Exemple pour 2 stations, au premier coup, tirage dans $\{0, T\}$: la station n'attend pas, ou la station attend T
 - » soit 2 stations tirent 0 ou T -> nouvelle collision
 - » Soit 2 stations 0 et T ou T et 0-> plus de collision
 - » Donc 1 chance sur 2 pour avoir à nouveau une collision
- Si nouvelle collision tirage dans $\{0, T, 2T, 3T\}$... etc

Stratégie de résolution des collisions

- Tirage aléatoire:
 - première collision on tire aléatoirement X dans $\{0, T\}$
 - deuxième collision on tire X dans $\{0, T, 2*T, 3*T\}$
 - i ème collision on tire X dans $\{0, \dots, (2^i - 1) * T\}$
 - On s'arrête à $i = 10$ pour l'intervalle des durées et à 16 pour le nombre d'essais
- Donc l'attente des stations est non déterministe
- Il existe un protocole déterministe (plus compliqué), il implique que l'on connaît le nombre et l'identification des stations du réseau

Ebauche du protocole Ethernet

- **Suivant évènement**
- **Si demande d'émission d'un paquet**
 - NbTentative=1
 - Tant que le support est occupé attendre
 - Emettre le paquet
- **Si détection de collision**
 - Arrêter l'émission du paquet
 - NbTentative++;
 - si NbTentative < 16 faire
 - » Tirage aléatoire du temps d'attente X en fonction du nombre de tentative
 - » Armer un timer de re-émission de durée X
- **Si sonnerie timer de re-émission**
 - Tant que le support est occupé attendre
 - Emettre le paquet

Efficacité du protocole Ethernet

- Soit T_{prop} le temps maximal de propagation sur le support (Longueur du support/vitesse), Soit T_{emis} le temps d'émission moyen d'une trame (Longueur trame/débit)
- Si l'on définit l'efficacité $E = T_{emis} / (T_{emis} + T_{att})$ où T_{att} est le temps moyen de contention (émission avec collisions et attente)
- On arrive en approximant à $T_{att} = 5,4 * T_{prop}$ avec un nombre important de machines
- Donc $E = T_{emis} / (T_{emis} + 5,4 * T_{prop})$
- $E = 1 / (1 + (5,4 * T_{prop}) / T_{emis})$

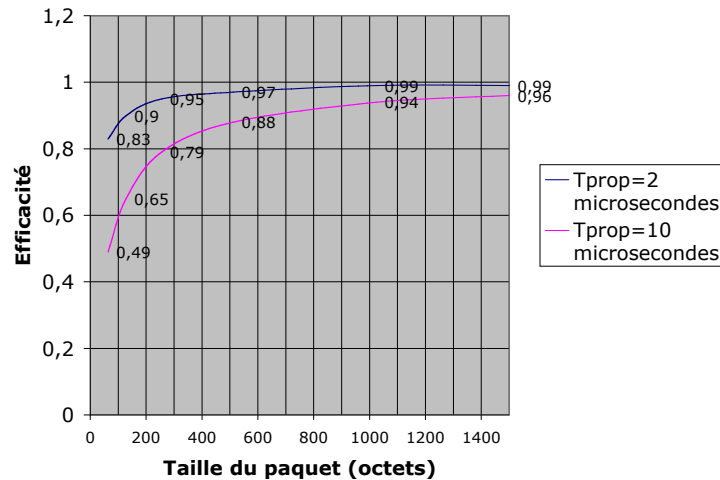
Efficacité du protocole Ethernet

- $E = 1 / (1 + (5,4 * T_{prop} / T_{emis}))$
- **Si T_{prop} tend vers 0, E tend vers 1**
 - Moins le réseau est grand, moins il y a de chance d'y avoir des collisions
- **Si T_{emis} devient grand par rapport au T_{prop} , E tend vers 1**
 - A T_{prop} constant, plus les paquets sont grands moins il y a de collisions
 - Il ne peut y avoir une collision que pendant l'émission du début du paquet (pendant T_{prop})
 - Après l'émission du début du paquet, il ne peut plus y avoir de collisions

Efficacité du protocole Ethernet

Efficacité en fonction de la taille des trames (64 à 1500 octets) :

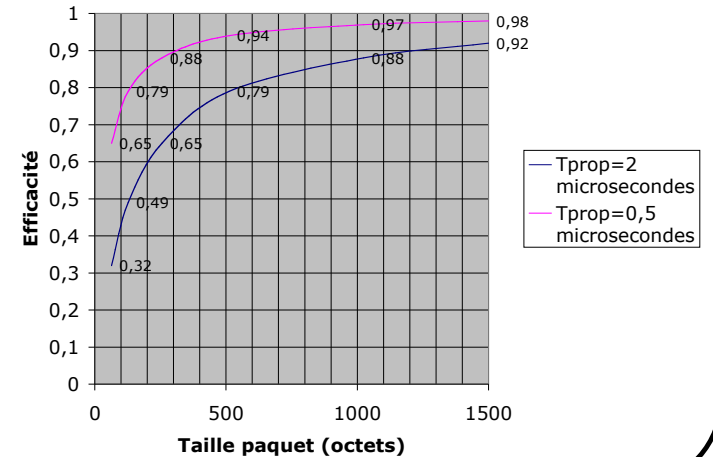
- 10 Mégabits/s
- Temps de propagation de 2 et 10 microsecondes (200m et 1000m)



Efficacité du protocole Ethernet

Efficacité en fonction de la taille des trames pour :

- 100 Mégabits/s
- Tprop de 0,5 et 2 microsecondes (50m et 200m)



Caractéristiques d'Ethernet 10 Mégabit/s

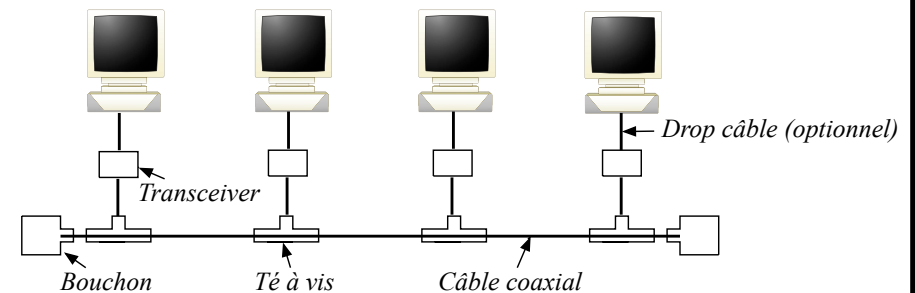
Normes 802.3

- **10 base 5** : câble coaxial épais, prise *vampire*, n'existe plus
 - 2500 m , segment 500m, répéteurs, 100 stations
- **10 base 2** : câble coaxial fin, Tés à vis et bouchons
 - 1000 m , segment 200m, répéteurs, 30 stations
- **10 base T** : paires torsadées, 2 paires en half-duplex
 - 400 m, Segment 150 m, hubs ou commutateurs, 1024 stations
- **10 base FX** : 2 fibres optiques
 - 2000 m, Segment 2000 m, 1024 stations

Taille minimale des trames

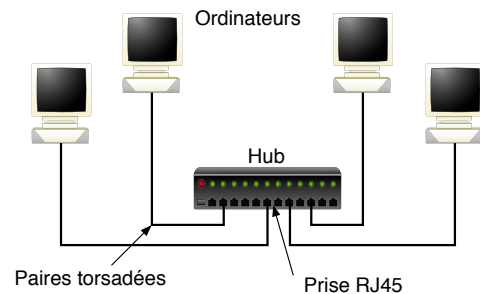
- Vitesse de propagation de l'ordre de 10^8 m/s, temps de traversée des répéteurs, hubs ...
- Tranche canal supposée (et devant être respectée suivant les différents supports): $T = 2 * \tau = 50. 10^{-6}$ s
- Longueur minimale des trames $50. 10^{-6} * 10. 10^6$ arrondie à 512 bits= 64 octets

Ethernet sur câble coaxial fin



Réseau Ethernet en paire torsadée

- Câble UTP5: 1 paire utilisée en entrée, 1 paire en sortie
- **Concentrateurs (Hub)** puis **commutateurs (switch)** centralisés dans une armoire de brassage pour faciliter l'administration et la maintenance
- Réseau en étoile mais à diffusion: le hub retransmet sur toutes les paires de sorties ce qu'il reçoit sur une paire d'entrée



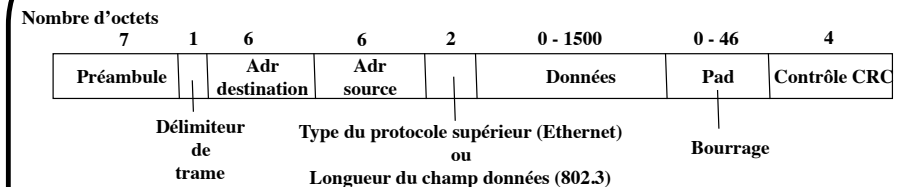
Fast Ethernet

- **Ethernet 100 Mégabit/s**
 - On garde taille minimale des trames 64 octets
 - Tranche canal 5,12 microsecondes
 - Réduction de la longueur du support (2 brins de 100 m)
- **Plusieurs Normes:**
 - 100 base TX : 2 paires utilisées dans câble UTP5
 - 100 base T4: 4 paires torsadées utilisées dans câble UTP5
 - 100 base FX : 2 fibres optiques
 - Codage 4B5B et NRZI
 - Maximum de 2 km en full duplex

Giga Ethernet

- **Ethernet 1 Gigabit/s**
 - Réduction à 20 m si on veut garder la longueur minimale des trames à 64 octets
 - On garde 200 m mais taille minimal des trames à 640 octets (en fait 512 octets)
 - Utilisation de 10% pour des paquets de 64 octets de données (on verra plus loin qu'aujourd'hui cette contrainte n'existe plus pour les réseaux en paire torsadée, il n'y a plus de collision dans les commutateurs)
 - Normalisé sur paire torsadée, câble coaxial et fibre optique
- **Ethernet 10 Gigabit/s**
 - 10 GBase T normalisé en 2006 sur câble catégorie 6 (55 m) ou 6a (100 m)
 - Plusieurs normes sur fibre optique (jusqu'à 40 km)
- **Ethernet 40 et 100 Gigabit/s**
 - Normalisée en 2010 sur fibre optique

La trame Ethernet



- Préambule : 7 octets 10101010, pour la synchronisation bit du récepteur
- Marqueur de début : 1 octet 10101011
- Silence inter-trame de 9,6 microsecondes, soit l'équivalent de 12 octets
- Adresse source et destination :
 - FF:FF:FF:FF:FF:FF adresse broadcast
- Champ longueur ou protocole supérieur: 2 octets.
 - La longueur d'une trame doit être au minimum de 64 octets de l'adresse destination au champ contrôle.
- Pad : si la longueur des données à transmettre est insuffisante ce champ contient des octets de remplissage quelconques.
- Contrôle : 4 octets servant au contrôle d'erreur (par CRC)

Accès multiple: Solutions d'allocation sans collision

- **Principe à base de jeton**

- Un message particulier appelé jeton, passe d'ordinateur en ordinateur à tour de rôle (topologie d'anneau).
- Recevant le jeton, un ordinateur a le droit d'émettre (un temps borné). Quand il a fini d'émettre, il transmet le jeton à son voisin.
- Il ne peut pas y avoir de collision
- Il faut connaître son voisin et avoir une topologie en anneau (physique ou virtuel)

Accès multiple: Solutions d'allocation sans collision

- **Problèmes**

- Si on ne dispose pas d'une topologie en anneau il faut la recréer, au moyen d'un protocole (dit protocole d'anneau virtuel : bus à jeton).
- Qui génère le premier jeton, le recrée en cas de disparition et en assure l'unicité ?
- Plus compliqué que le principe d'Ethernet

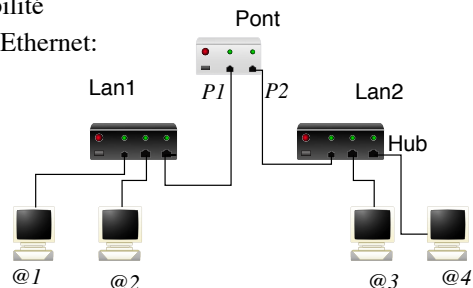
- **Intéressant pour des contraintes temps réel, le tour de parole est régulier et à temps borné**

- **Anneau ou bus à jeton (norme 802.5 et 802.4)**

- En désuétude par rapport à Ethernet (même pour les réseaux industriel temps-réel)
- Le protocole **USB** utilise un algorithme de ce type mais dans ce cas particulier c'est l'ordinateur qui distribue le jeton à tour de

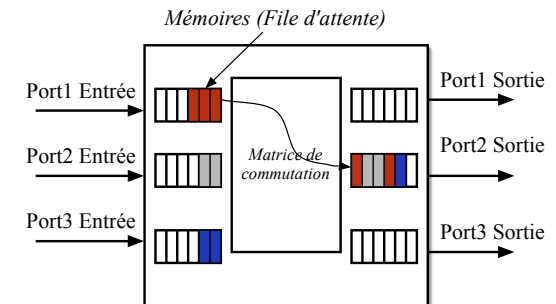
Les ponts

- Connexion de LAN hétérogènes :
 - Exemple: Point d'accès WIFI = Pont Ethernet/WIFI
 - Le pont doit changer l'entête *Liaison de donnée* quand les paquets passent d'un réseau à l'autre
- Optimisation de la charge
- Le pont fait office de filtre et il y a donc moins de charge sur chaque LAN
- Augmentation des performances : domaines de collisions restreints
- Augmentation de la fiabilité
- Exemple Pont Ethernet/Ethernet:



Fonctionnement d'un commutateur (switch) Ethernet

- **Pont Ethernet/Ethernet haute performance**
- **Utilisé à la place des hubs pour augmenter les performances**



Fonctionnement des commutateurs Ethernet

- **Mémorisation des trames dans le switch (files d'attentes)**
- **Filtrage des trames au vue des adresses Ethernet destinations**
 - Analyse de l'entête Ethernet
 - Re-émission sur un seul port
- **Gestion d'une table : adresses Ethernet /interfaces**
 - Durée de vie limitée
 - **Auto-apprentissage des adresses Ethernet grâce à l'adresse source des premiers paquets.**
- Les premiers paquets sont re-émis sur tous les ports
- Possibilités de cascader plusieurs switches

Les commutateurs

- Exemple d'une table gérée dans des switches
- Tables une fois remplies:
- @ représente une adresse Ethernet

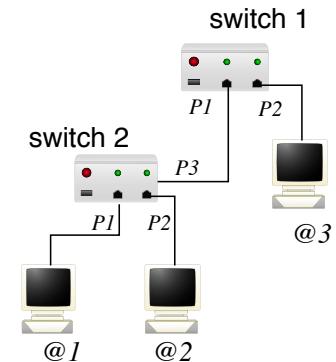
• Switch1
Adresse Ethernet / Interface

- @1, @2 P1
- @3 P2

• Switch2

Adresse Ethernet / Interface

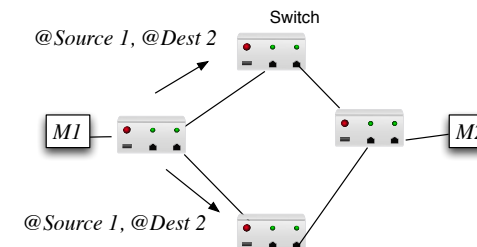
- @1 P1
- @2 P2
- @3 P3



Algorithme d'auto-apprentissage

- A la réception d'une trame (@source, @destination) sur une interface P faire
 - Mémoriser la trame
 - si @source est dans la table (@, P) alors
 - » re-armer le timer
 - » sinon la rajouter et armer un timer
 - si @source est dans la table avec une interface différente alors
 - » modifier la table avec P et re-armer le timer
 - si correspondance (@destination, interface) est dans la table alors
 - » re-émettre la trame sur interface
 - » sinon re-émettre sur toutes les interfaces
 - Si timer associé à (@, interface) sonne alors
 - » Supprimer (@, interface) de la table

Problème de boucle



- Le premier paquet émis depuis M1 vers M2 sera re-émis par le switch1 vers les deux switch 2 et 3 (broadcast car table vide)
 - Ces paquets sont reçus par les deux switches 2 et 3 qui le renvoie vers le switch 4
 - le switch 4 re-émet en broadcast ces paquets.
 - Des paquets tournent sans arrêt sur la boucle
- Le réseau explose ... on parle de **tempête** de broadcast

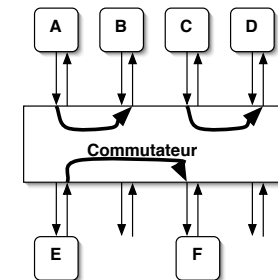
Le problème survient dès qu'il y a une boucle dans le réseau

Algorithme de l'arbre recouvrant (spanning tree)

- Trouver un arbre recouvrant un graphe quelconque pour éliminer les boucles
- Une fois l'arbre déterminé, les ponts désactivent certains ports
- Construction de l'arbre:
 - Basé sur les identificateurs des ponts (adresse Ethernet ou autres)
 - » Racine : identificateur le plus grand
 - Echange de paquets spéciaux (BGPDU: Bridge Protocol Data Unit) entre les ponts contenant les informations permettant de déterminer la racine et le chemin vers la racine
 - Mise a jour périodique de ces informations pour d'éventuelles modifications de topologie

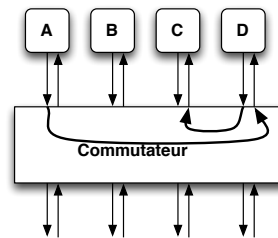
Performances d'un commutateur Ethernet

- Les trames ne sont réémises que vers le destinataire
- Il n'y a plus de collision
- Possibilités de plusieurs flux en parallèle
 - Si le switch suit le rythme au niveau commutation
- Exemple : Sur un switch 100 Mégabit; 3 flux à 100 mégabits/s !



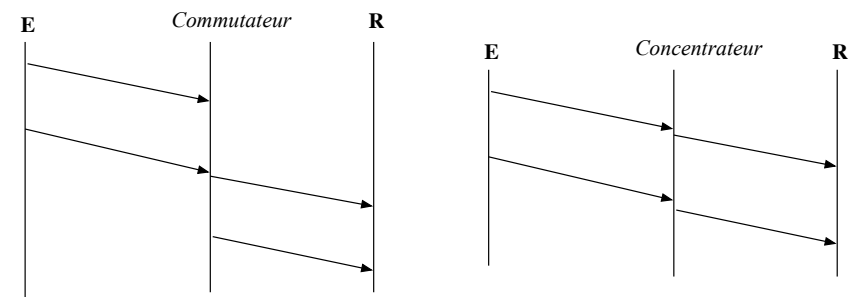
Commutateur Ethernet Full Duplex

- Possibilités d'émettre et de recevoir en même temps sur les deux paires torsadées (on parle de Full-duplex)
- Donc sur du 100 base T on peut atteindre 100 mégabit/s en émission et 100 mégabit/s en réception
- En Full duplex, l'algorithme Ethernet change :
 - Carte Ethernet particulière dite "full duplex"
 - pas de détection de porteuse,
 - pas de détection de collision,
 - plus de taille minimale de trame
 - on peut augmenter fortement la longueur des brins



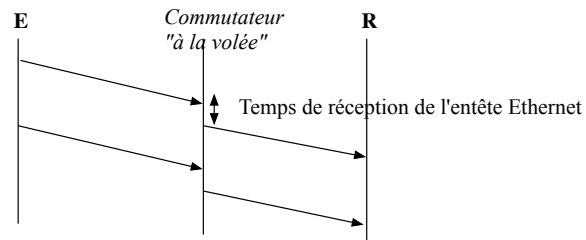
Amélioration de la commutation Ethernet

- Stocker et renvoyer (Store and Forward)
 - Reçoit une trame entièrement, vérifie sa validité et la retransmet sur le bon port après consultation de sa table
 - Le commutateur rajoute au moins un temps d'émission à la latence
 - Temps de traversée : 120 micro secondes pour une trame de 1500 octets à 100 Mégabit/s



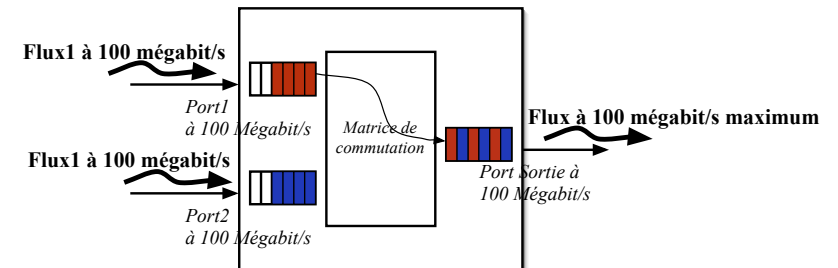
Amélioration de la commutation Ethernet

- **Commutateur à la volée (cut-through switching):**
 - pour diminuer le temps de mémorisation/re-émission, le commutateur commence à re-émettre dès qu’il a reçu l’adresse destination
 - Intéressant seulement quand la file d’attente de sortie est vide
 - Temps de traversée : 20 micro secondes pour une trame de 64 octets (10 mégabit/s)



Saturation des commutateurs

- File d’attente (mémoire) pleine
- Suppression des trames par le commutateur
- Exemple:



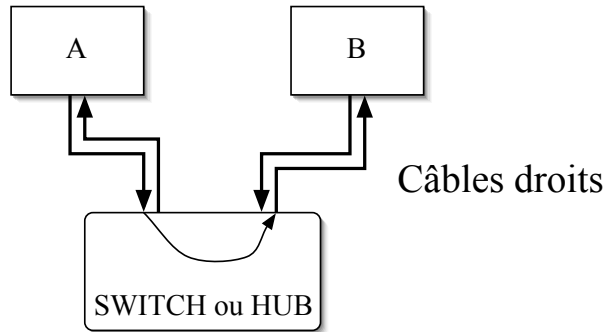
Saturation des commutateurs

- **Reprise des pertes par couche supérieure: dégradation forte des performances**
- **Introduction de contrôle de congestion au niveau liaison de donnée:**
 - Fausse trame générée par le commutateur dans le cas du half-duplex pour limiter les émissions
 - Trame de “pause” comprise par les cartes Ethernet dans le cas du full-duplex

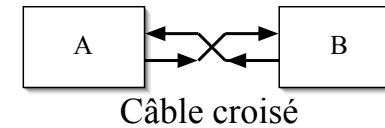
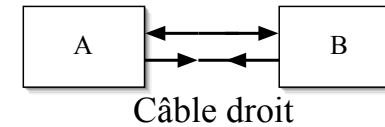
Auto-négociation

- **Différentes options possibles:**
 - Transmission (Full et half-duplex)
 - Débits et technologie (10 base T, 100 base T, 100 base T4, ...)
- Les cartes Ethernet et commutateurs effectuent maintenant une négociation au moment du branchement pour décider de ces options
- Négociation possible seulement sur paire torsadé et fibre optique
- Adaptation aussi dans certains cas au type de câble (croisés ou droits)

Pour finir Des histoires de câbles



Connexion de deux ordinateurs



- De la même façon il faut utiliser un câble croisé pour interconnecter deux Hubs ou deux Switchs
- Dans les cartes et switch récents le croisement (ou non) se fait automatiquement dans l'équipement

Avant le 1er TP: les adresses IP

- **IPv4:** sur 4 octets donnés en décimal séparés par des points
- **Exemple:** 201.123.45.10
- **Deux parties distinctes:**
 - Une partie désigne le **réseau local**
 - Une partie désigne la **machine** dans le réseau local
- **Différentes classes suivant les besoins (définies par les 4 premiers bits du 1er octet):**
 - classe A: (0.....) ; partie machine: 3 derniers octets (R.M.M.M)
 - classe B: (10.....) ; partie machine: 2 derniers octets (R.R.M.M)
 - classe C: (110.....) ; partie machine: le dernier octet (R.R.R.M)
 - Classe D: (1110.....) Multicast (G.G.G.G)
- **Exemple:**
 - 195.0.0.4 et 195.0.0.5 : deux adresses de classe C de deux machines appartenant au même réseau local 195.0.0.0
 - On désigne un réseau en mettant la partie machine à 0
- **Partie machine à 0 interdit** pour ne pas confondre une adresse de réseau et une adresse de machine

Les adresses IP

- **Exemple:**
 - adresse 160.1.2.3 : 160= 10100000 donc classe B
 - Réseau 160.1.0.0 ; adresse de machines possibles: 160.1.0.1 à 160.1.255.254
- **Adresses particulières:**
 - 127: boucle locale (*Loopback*); interface virtuelle
 - 0.0.0.0 : Utilisé dans protocole d'apprentissage d'adresse
 - *Broadcast*: Pour s'adresser à toutes les machines d'un réseau local
 - Partie machine ne comporte que des 1(en binaire)
 - Exemple: 192.0.0.255
 - 255.255.255.255
 - Broadcast Ethernet : FF:FF:FF:FF:FF:FF (en Héxa)

Les adresses IP sans classe

- Depuis le début des années 1990 :

- Sans classe (*classless*) pour économiser des adresses
- Adresse réseau / Nombre de bit de la partie réseau
- Permet de limiter le gaspillage des adresses
- Exemple: 192.0.0.64 / 26, les deux bits de poids fort du dernier octet font aussi partie de la partie réseau
- On parle de **PREFIXE** pour désigner la partie réseau

- **IPV6**

- Adresses sur 16 octets données en Hexadécimal:
2001:660:5301:8000:0:0:0:1aed/64
- 6,67. 10¹⁷ adresses au millimètre carré de la surface terrestre !
- Même principe de préfixe que IPv4
- Saturation des adresses IPv4 (2012)
- Co-habitation IPV6-IPV4 à l'aide de tunnels

Le masque de réseau (*Netmask*)

- Comment calculer la partie réseau d'une adresse IP de machine ?

- **Netmask**

- Composé de bits à 1 sur la partie réseau et de bit à 0 sur la partie machine
- Exemples de *netmask*:
 - Classe A (/8): 255.0.0.0; Classe C (/24): 255.255.255.0
 - Réseau/ 26 : 255.255.255.192; Réseau /28: 255.255.255.240
- Pour calculer la partie réseau (donc l'adresse du réseau) à partir d'une adresse de machine il suffit de faire le ET bit à bit avec le Netmask

- Exemple:

- 160.1.64.69/27, Netmask= 255.255.255.224
- 1010 0000. 0000 0001. 0100 0000. 0100 0101
- ET 1111 1111. 1111 1111. 1111 1111. 1110 0000
- 1010 0000. 0000 0001. 0100 0000. 0100 0000= 160.1.64.64

- On peut aussi calculer l'adresse Broadcast à l'aide du Netmask

- Adresse Réseau *Ou-booléen* Complément-booléen du Netmask