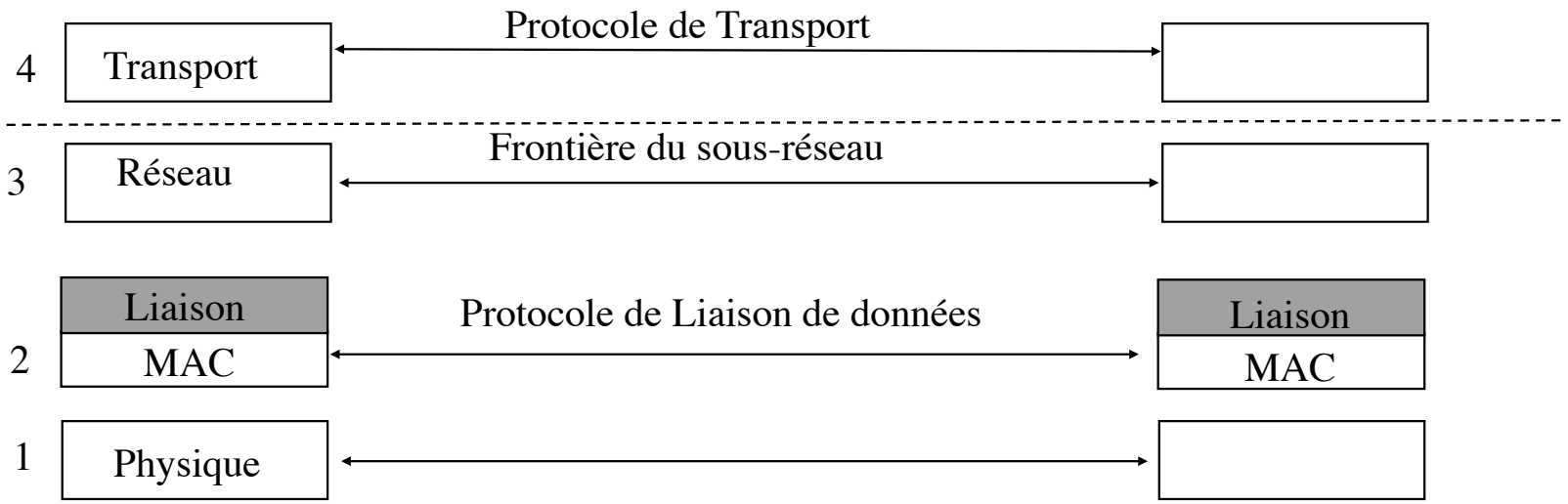
La couche liaison de données

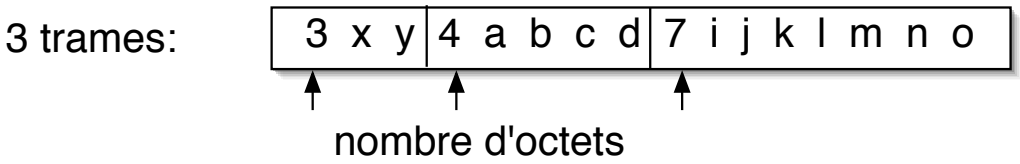


Services fournis à la couche réseau

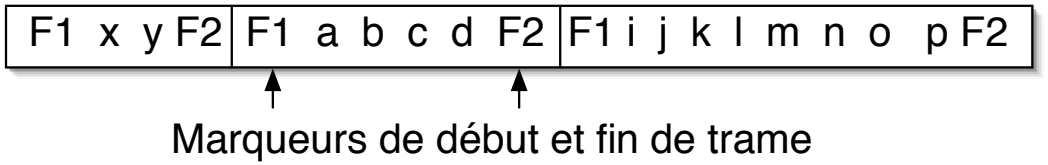
- **Découpage en trame**
- **Un transfert fiable: La détection/correction des erreurs**
- **Contrôle de flux et récupération d'erreur**
- **Accès multiples à un support (fait par la sous couche MAC : Medium Access Control)**
  - **Un exemple : le protocole Ethernet**
- **Une grande partie de ces problématiques 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**

Notion de trame

- **But: fixer une unité pour le contrôle d'erreur.**
- **Techniques de découpage en trame**
  - **1- Compter les caractères**



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



  - **3- Changer le codage utilisé dans la couche physique**

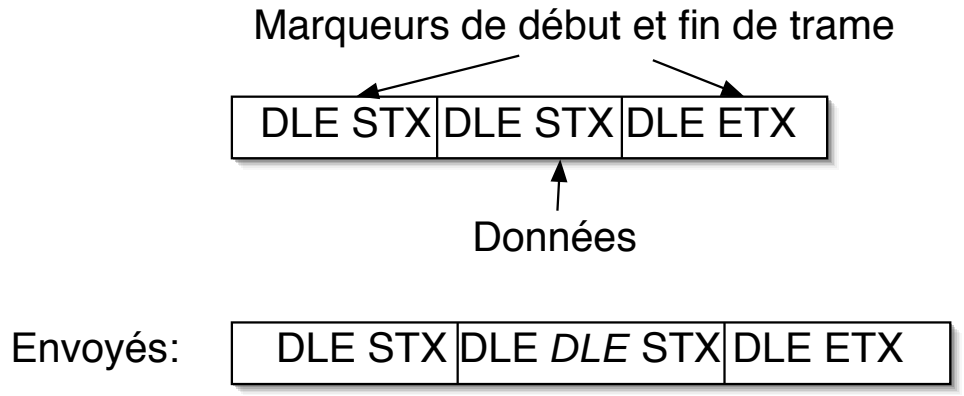
Problème de Transparence (1)

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 effectue une violation du codage Manchester pour détecter la fin de trame**

Le contrôle d'erreur

- **Les données peuvent être modifiées (ou perdues) pendant le transport**
  - **Un service primordial pour de nombreuses applications**
  - **Exemple : le transfert de fichier**
- **La détection d'erreur:**
  - **Comment se rendre compte de la modification/pertes des données à l'arrivée des trames ?**
- **La correction d'erreur, deux techniques :**
  - **Comment corriger à l'arrivée les données erronées ? : La *correction***
  - **Faire en sorte que l'émetteur, renvoie les trames erronées/perdues: la *récupération d'erreurs***

La détection/correction d'erreur

- **Idée: rajouter de l'information aux données permettant de détecter/corriger les erreurs à l'arrivée**
- **Même technique pour le stockage de données**
- **Exemple de détection: Le code de parité**
  - **On rajoute un bit à 1 ou 0 suivant la parité du nombre de bits à 1 dans les données. Le récepteur vérifie la valeur de ce bit de parité.**
  - **Exemple:**
    - » **Données: 1 0 0 0 1 1 Bit de parité: 1**
    - » **Données: 1 0 0 1 1 1 0 1 1 Bit de parité : 0**
- **Qualité de la détection d'erreur ?**
  - **Est-on capable de détecter 1 seul bit erroné ? 2 bits ?...**

Contrôle d'erreur : un modèle d'étude.

- **Mot de code**

  Si une trame contient m bits de données et r bits de contrôle, on appelle mot du code le mot formé par les m + r bits. On pose n = m + r.

- **Distance de Hamming**

  Étant donné deux mots de n bits m1 et m2, le nombre de bits dont ils diffèrent est appelé leur distance de Hamming (notée Disth).

- **Distance de Hamming du code complet**

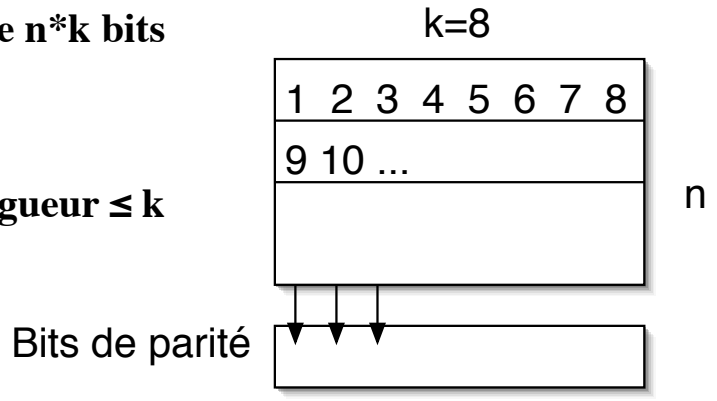
  h = { Min Disth(x1, x2) ; x1 et x2 M }

  M est l'ensemble des 2^m mots de codes possibles si on admet que les r bits de contrôle sont calculés en fonction des m bits de données.

Détection d'erreur

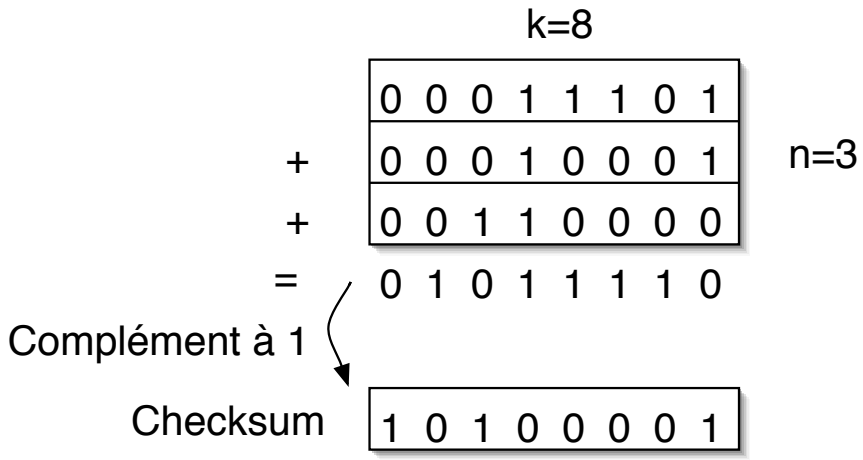
- **Propriété:**
  - **Pour détecter (à coup sûr) x erreurs il suffit que la distance de Hamming h ≥ x + 1**
  - **En effet ainsi s'il y a x erreurs on ne pourra pas "retomber" sur un code existant (différent forcément de x+1 bits)**

- **Exemple de code détecteur :**
  - **Bit de parité**
    - **m = 2, r =1 : M = {000, 011, 101, 110 }**
    - **h = 2 mais détecte aussi tous les erreurs dont le nombre est impair**
  - **Bit de parité par colonne:**
    - **Trame considérée comme une matrice n*k bits**
    - **1 bit de parité par colonne**
    - **h=2**
    - **détection des rafales d'erreurs de longueur ≤ k**



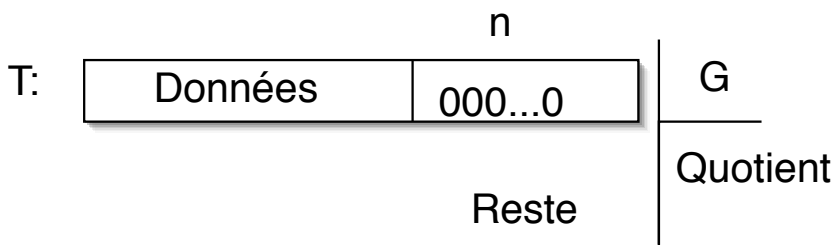
Détection d'erreur par "checksum"

- **Données considérées comme n mots de k bits**
- **Bits de contrôles = complément à 1 de la somme des n mots**
- **A la réception la somme des n mots de données plus le checksum ne doit pas contenir de 0**
- **h=2 mais détection aussi des rafales d'erreur de longueur ≤ k**
- **Utilisé dans UDP, TCP**



Détection d'erreur par CRC (Cyclic redundancy Code)

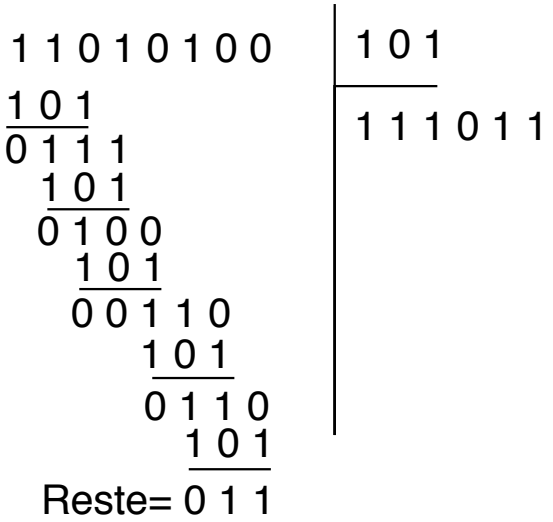
- Basée sur des calculs de division de polynôme à coefficient dans [0, 1]
- Exemple: 1 0 1 0 1 correspond à x^4+x^2+1
- Division de $x^3+x^2+1= (x^2+1)* (x + 1) + (x)$: Reste =x, Quotient = x+1
- Arithmétique polynomiale modulo 2 (sans retenue): soustraction et addition sont équivalentes à un ou-exclusif bit à bit
- On se donne un polynôme générateur G de degré n qui détermine le nombre de bits de contrôle



- T= Quotient*G + Reste donc (T+Reste)/G = 0
- La trame envoyée E= (Données, Reste) est divisible par G, il suffit à l'arrivée de calculer la division de E par G. Si le reste est non nul il y a une erreur

Code détecteur par CRC

- **Exemple: 6 bits de données: 110101 , Polynôme générateur 101 : x^2+1**



- **E= 110101 11**
- **Ne dépend pas de la taille des données**
- **Calcul coûteux mais souvent fait par hard : ou exclusif successifs au fur et à mesure que la trame arrive.**
- **On peut avec n=16 détecter toutes les erreurs simples et doubles, toutes les erreurs comportant un nombre impair de bits et tous les paquets d'erreur de longueur ≤ 16 et, avec une très bonne probabilité, les paquets d'erreurs de longueur supérieure .**
- **Exemple: Ethernet utilise un champs CRC à 16 bits.**

Correction d'erreur

- **Propriété:**
  - **Pour corriger x erreurs il suffit que la distance de Hamming h ≥ 2x + 1**
  - **En effet s' il y a x erreurs, le code erroné reste ainsi le plus proche du code juste, on peut donc le retrouver. Les autres ont encore x+1 différences.**
- **Exemple de code correcteur**

  m = 2, r = 3

  M = {00111, 01100, 10000, 11011}, h = 3, on corrige une erreur
- **Problème :**
  - **Quelle est la valeur minimale de r permettant de corriger les erreurs simples dans des trames de m bits de données et r de contrôle ?**
  - **Il faut que l'ensemble possible des mots erronés (différant de 1 bit) pour chaque mot juste soit inférieur au nombre de mots possibles, pour que l'on ne "retombe" pas sur un autre mot erroné ou un original (quand il y a une erreur) c'est à dire:**
    - **$(n+1)\ 2^m <= 2^n$ soit $(m + r + 1) <= 2^r$**

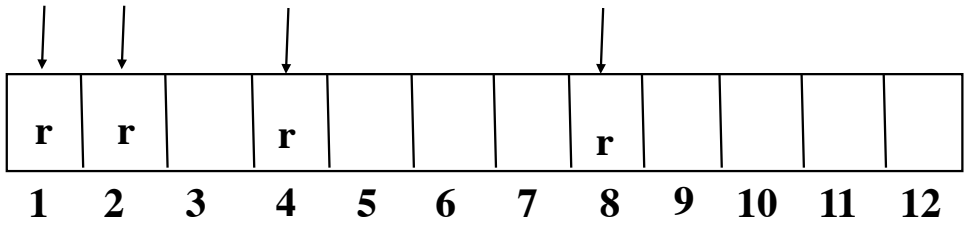
      2^m mots justes avec (n+1) différents (n erronés et le juste) pour chacun
  - **Les codes d'erreur "coûtent" chers**
    Par exemple: 4 bits sont nécessaires pour corriger une erreur sur 8 bits de données.

Le code correcteur de Hamming

- Les bits de données qui servent au calcul d'un bit de contrôle de numéro c sont ceux tel que c apparaît dans la décomposition en puissance de 2 de leur numéro.
- Exemple: 7 = 1 + 2+ 4 donc 7 apparaît dans le calcul de 1, de 2 et de 4

  1 calculé de telle façon que (1, 3, 5, 7, 9, 11, …) parité paire
  2 calculé de telle façon que (2, 3, 6, 7, 10, 11,…) parité paire
  4 calculé de telle façon que (4, 5, 6, 7, 12, 13,…) parité paire
  ...
- Valable pour un nombre quelconque de bits de donnée.
- Nombre de bit de contrôle minimal pour n bits de données
- A destination on recalcule les bits de contrôle. La somme des numéros des bits de contrôle erronés donne le numéro du bit qui porte l'erreur.

Les bits de contrôle sont les bits de numéro= 2^n



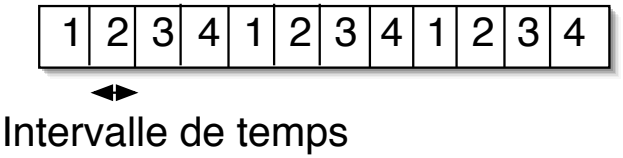
La sous couche MAC: Le contrôle d'accès multiple au canal

- **Réseaux particuliers à diffusion: bus, radio**
  - **Intéressant pour leur faible coût**

- **Problème :**
  - **Réseau à diffusion implique un support unique pour n émetteurs/récepteurs**
  - **Il existe différentes solutions pour réaliser ces accès multiples:**
    - » **Par partage strict du support**
      - **Le support est "divisé", soit dans le temps, soit physiquement**
    - » **Par accès aléatoire :**
      - **On parle quand on veut**
    - » **Par accès séquentiel**
      - **On parle à tour de rôle**

Accès multiple: solutions par partage

- **Multiplexage en fréquence: Découpage et allocation permanente des plages de fréquences.**
  - **Exemple la radio FM**
  - **Problème de l'allocation des plages de fréquences**
- **Multiplexage temporel: Découpage dans le temps de l'allocation de la totalité de la bande passante à chaque entité**



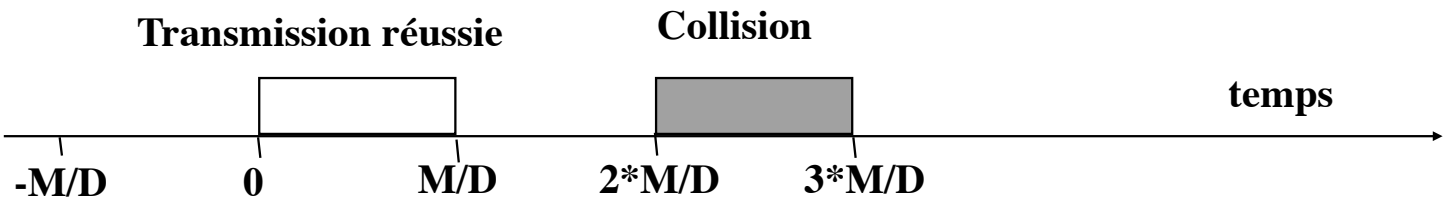
- **Peu efficace pour des échanges sporadiques**

Accès multiple: solutions par accès aléatoires

- **Une seule fréquence, possibilités de collisions**
- **Protocoles ALOHA (bonjour en hawaïen)**
  - **Réseau radio entre iles hawaïennes puis Ethernet en 73 et normalisation en 80**
- **ALOHA PUR**
  - **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 ré-émettre leur paquet ultérieurement.**
  - **La ré-émission a lieu immédiatement avec une probabilité p. Sinon l'émetteur attend la durée d'émission d'une trame puis émettra avec une probabilité p.**
  - **On peut montrer que l'on arrive à une efficacité de 18% du débit total dans le meilleur des cas**

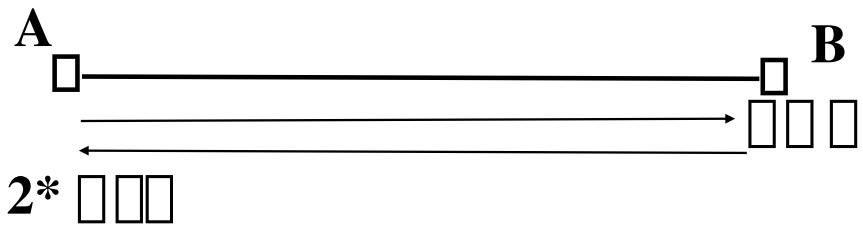
Aloha discrétisé

- **Le temps est discrétisé en intervalles de durée *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 câble est libre**
  - **CD (Collision detection): une machine "écoute" le câble 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**
- **Il y a encore des collisions !**
- **Détection des collisions**
  - **une station qui veut détecter les collisions doit observer le câble pendant une durée de 2* □ (avec □= temps de propagation du signal d'un bout à l'autre du câble) sinon l'efficacité s□effondre : les erreurs devront être rattrapées par les couches supérieures**
  - **T=2* □est appelé la tranche canal**



Stratégie de résolution des collisions

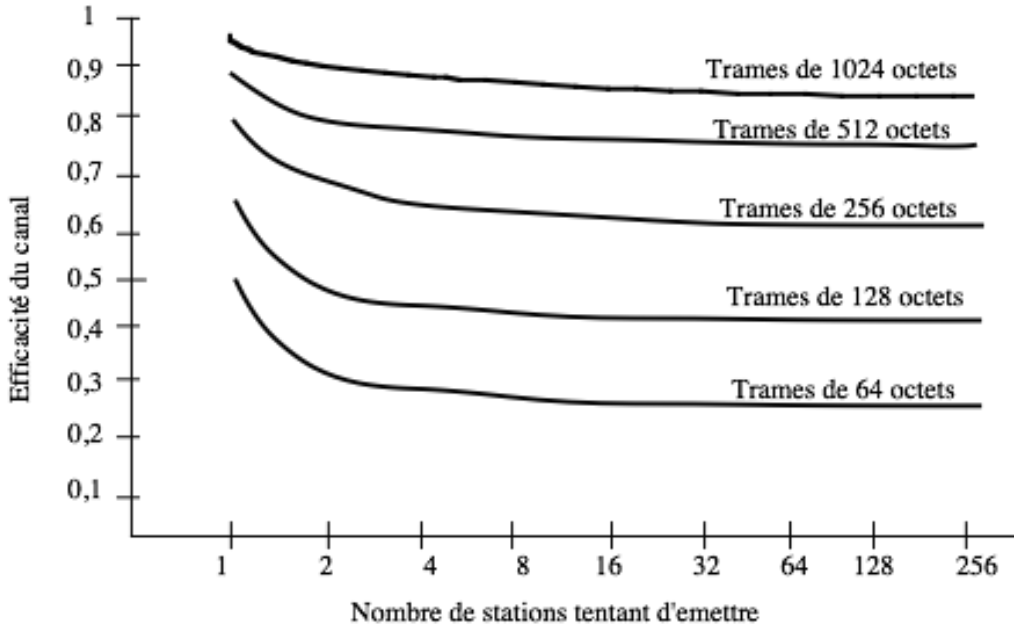
- Une station ayant détectée une collision se met en attente pendant une durée aléatoire X dont l'intervalle de définition double à chaque nouvelle collision jusqu'à une valeur maximum au-delà de laquelle elle abandonne
- Exemple pour 2 stations, au premier coup:
  - » soit 2 stations tirent 0 ou 1 -> 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
- Sinon on passe à une chance sur 4 au deuxième coup ... etc
- 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

Le protocole Ethernet

- **Construire trame à émettre; NbTentative=0**
- **Répéter:**
  - **Tant que câble occupé faire attendre**
  - **Tant que pas de collision faire transmettre**
  - **Si collision détectée et NbTentative <16 faire**
    - » **Arrêter transmission + émission brouillage**
    - » **Tirage aléatoire du temps d'attente X en fonction du nombre de tentative**
    - » **Attendre X**
    - » **NbTentative= NbTentative+1**
- **Jusqu'à transmission complète ou NbTentative =16**
-
- **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, ..., $(2^i - 1) * T$ }**
  - **On s'arrête à i = 10 pour l'intervalle des durées et à 16 pour le nombre d'essais**

Efficacité du protocole Ethernet

- Soit Tprop le temps maximal de propagation sur le support (Longueur du support/ vitesse), Soit Témis le temps d'émission moyen d'une trame (Longueur trame/débit)
- Si l'on définit l'efficacité E comme le temps moyen d'émission de trame sans collision
- Avec un nombre important de station voulant émettre, on arrive approximativement à E= 1/(1+(5*Tprop/Témis))
  - Tprop tend vers 0 , E tend vers 1
  - Témis devient grand / Tprop, E tend vers 1
- Efficacité en fonction de la taille des trames et du nombre de station tentant d'émettre avec Tprop=50 microsec et un débit de 10 mégabits/s



Caractéristiques d'Ethernet

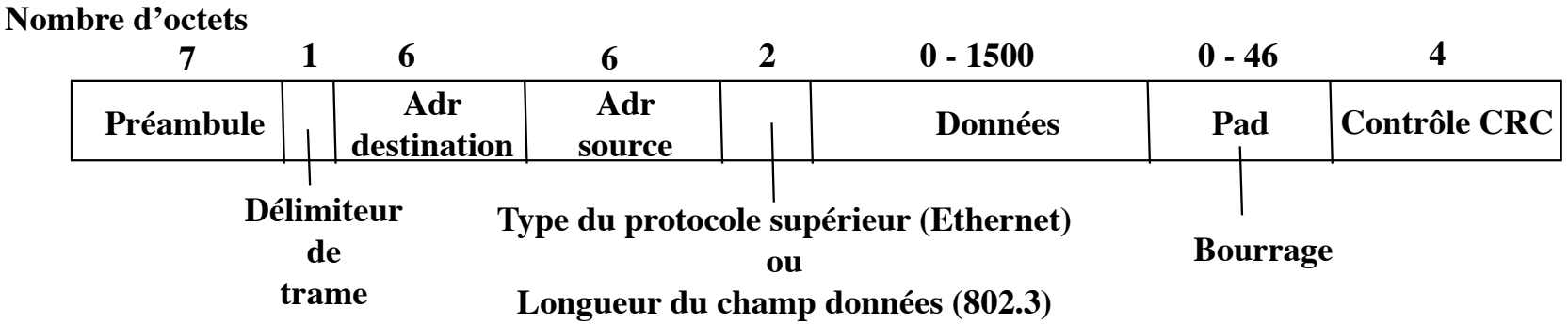
- **Normes 802.3**
  - **10 base 2 : câble coaxial fin**
  - **10 base T, 100 base T: paires torsadées**
  - **100 base FX : fibre optique**

- **Ethernet 10 méga bit/s**
  - **Longueur maximale du support= 2500 m , vitesse de propagation= 10^8 m/s**
  - **Temps supposé maximal sur le réseau: Tranche canal : T = 2 * □ = 50. 10^{-6} s**
  - **Longueur minimale des trames 50. 10^{-6}*10. 10^6 arrondie à 512 bits= 64 octets**

- **Ethernet 100 Mégabit/s**
  - **On garde taille minimale des trames 64 octets**
  - **Tranche canal 5,12 microsecondes**
  - **Réduction de la longueur du support (250 m)**

- **Ethernet 1 gigabit/s existe, bientôt Ethernet 10 gigabit/s**

La trame Ethernet



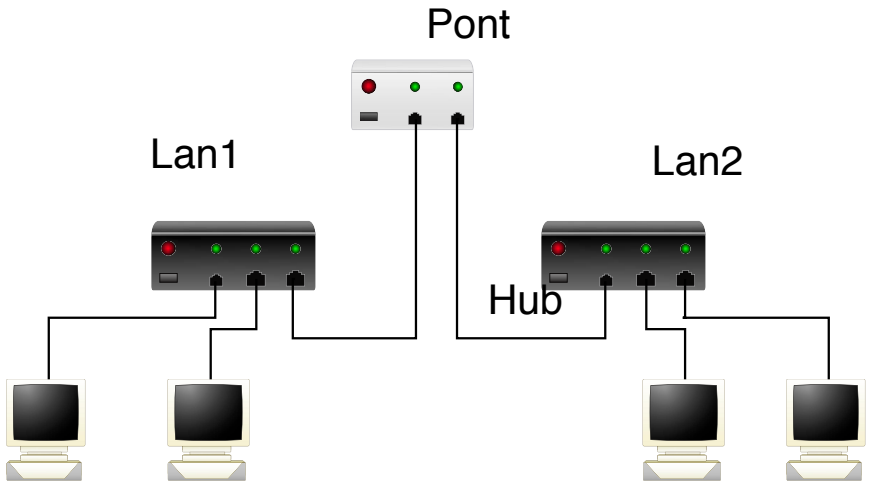
- Préambule : 7 octets 10101010, pour la synchronisation bit du récepteur
- Marqueur de début : 1 octet 10101011
- Adresse source et destination :
  - FF:FF:FF:FF:FF:FF adresse broadcast
- Champ longueur ou protocole supérieur: 2 octets.
  - La longueur d'une trame doit être au minimum de 64 octets de 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 : il s'agit de 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 de station en station à tour de rôle (topologie d'anneau).**
  - **Recevant le jeton, une station qui veut émettre retire le jeton et le remplace par son message. Quand elle a fini d'émettre la station qui a retiré le jeton en ré-émet un qu'elle transmet à sa voisine. Une station ne désirant pas émettre retransmet le jeton à sa voisine.**
- **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 ?**
- **Anneau ou bus à jeton (norme 802.5 et 802.4)**
- **En désuétude par rapport à Ethernet (même pour les réseaux industriel temps-réel)**

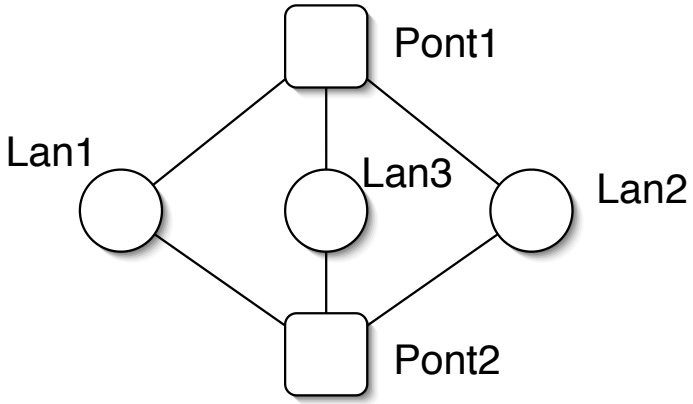
Les ponts

- **Connexion de LAN hétérogènes**
- **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**



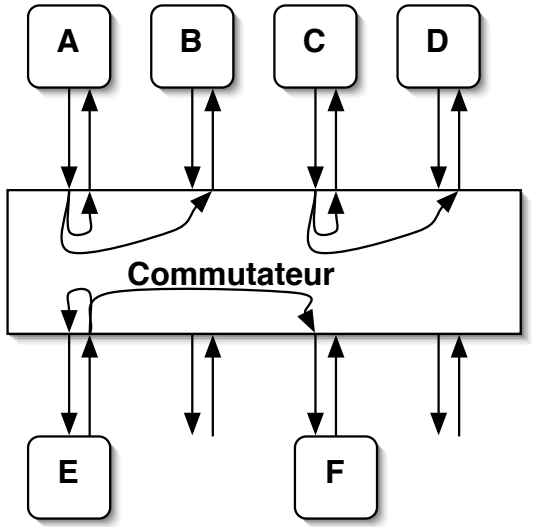
Fonctionnement des ponts

- **Filtrage des trames au vue des adresses Ethernet destinations**
  - **Mémorisation des trames**
  - **Analyse de l'entête Ethernet**
  - **Réémission sur un seul port**
- **Gestion d'une table : adresses Ethernet appartenant à chaque réseau avec durée de vie limitée**
- **Auto-apprentissage des adresses Ethernet grâce à l'adresse source des premiers paquets. Les premiers paquets sont réémis sur tous les ports.**
- **Algorithme de l'arbre recouvrant (spanning tree) pour éliminer les problèmes de boucle et de multiplication de trames**



Le commutateur (ou switch) Ethernet

- Pont haute performance (nombreux ports, filtrage)
- Utilisé à la place des hubs pour augmenter les performances
- Existe avec différents débits sur les ports
  - Par exemple: 1 port à 100 mégabit/s et 10 ports à 10 mégabit/s
- Mémorisation des trames dans le switch
  - Possibilités de plusieurs flux en parallèle
  - Il n'y a plus de collisions



Commutateur Ethernet

- **Possibilités d'émettre et de recevoir en même temps**
- **Algorithme Ethernet change : pas de détection de porteuse, de détection de collision**
- **Carte Ethernet particulière dite "full duplex"**
- **Saturation possible du commutateur (mémoire pleine)**
  - **Suppression des trames par le commutateurs (reprise des pertes par couche supérieure)**
  - **Fausse trame générée par le commutateur dans le cas du half duplex pour limiter les émissions**
  - **Trame de stop/start comprise par les cartes Ethernet dans le cas du full-duplex**
- **Commutateur à la volée (cut-through switching):**
  - **pour diminuer le temps de mémorisation/réémission, le commutateur commence à réémettre dès qu'il a reçu l'adresse destination**
  - **Intéressant seulement quand la file d'attente de sortie est vide**