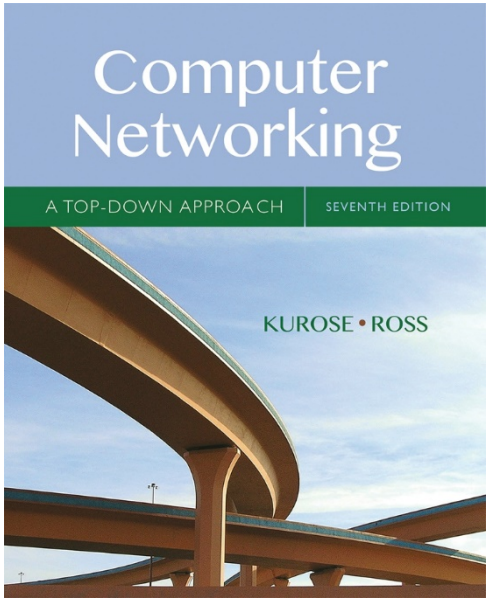
Chapitre V
La couche liaison
(Partie I)



Computer Networking: A Top Down Approach
7ème édition
Jim Kurose, Keith Ross
Addison-Wesley
2017

Couche liaison: plan

5.1 introduction, services

5.2 détection d'erreurs, correction

5.3 protocoles d'accès multiple

5.4 LANs

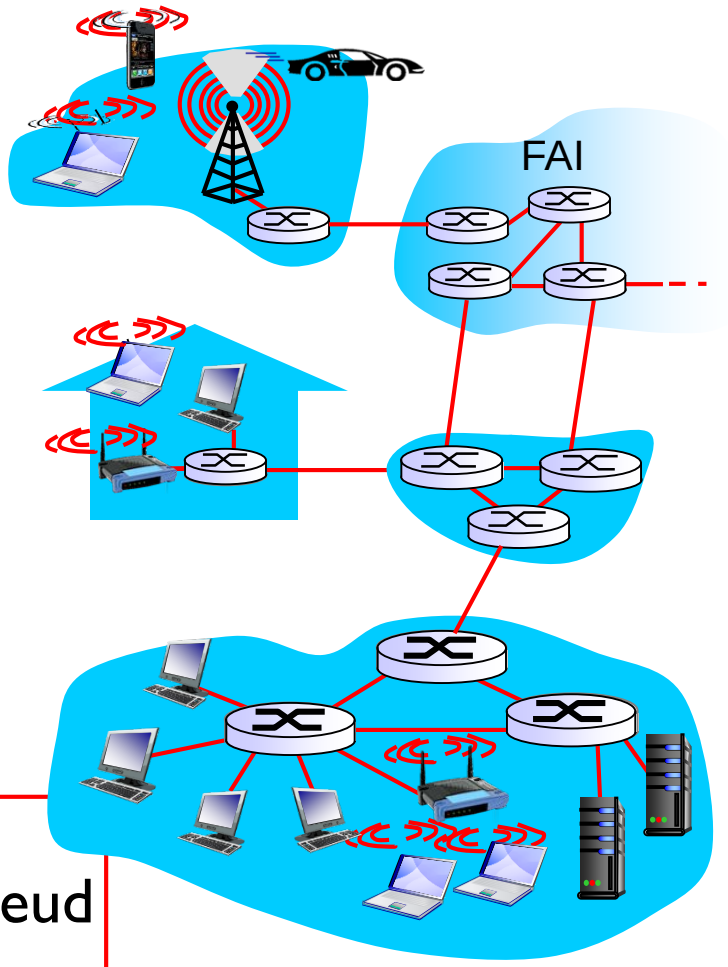
- adressage, ARP
- Ethernet
- commutateurs
- VLANS

5.5 virtualisation: MPLS

Couche liaison: introduction

terminologie:

- hôtes et routeurs: nœuds
- Les canaux de communication qui connectent les nœuds adjacents: liens
  - filaire
  - sans fil
- paquet de couche 2: trame, encapsule un datagramme



la couche liaison est responsable d'acheminer un datagramme d'un noeud au noeud *physiquement adjacent*

La couche liaison

- Les datagrammes sont transférés par différents protocoles de liaison sur différents liens:
  - ex., Ethernet sur un premier lien, frame relay sur les liens intermédiaires, 802.11 sur le dernier lien
- chaque protocole de liaison fournit des services différents
  - ex., fiable ou pas

analogie:

- Voyage de Québec à Lausanne
  - limo: Québec à Montréal
  - avion: Montréal à Genève
  - train: Genève à Lausanne
- touriste = datagramme
- étape de transport = lien de communication
- mode de transport = protocole de couche liaison
- agence de voyage = algorithme de routage

Services de la couche liaison

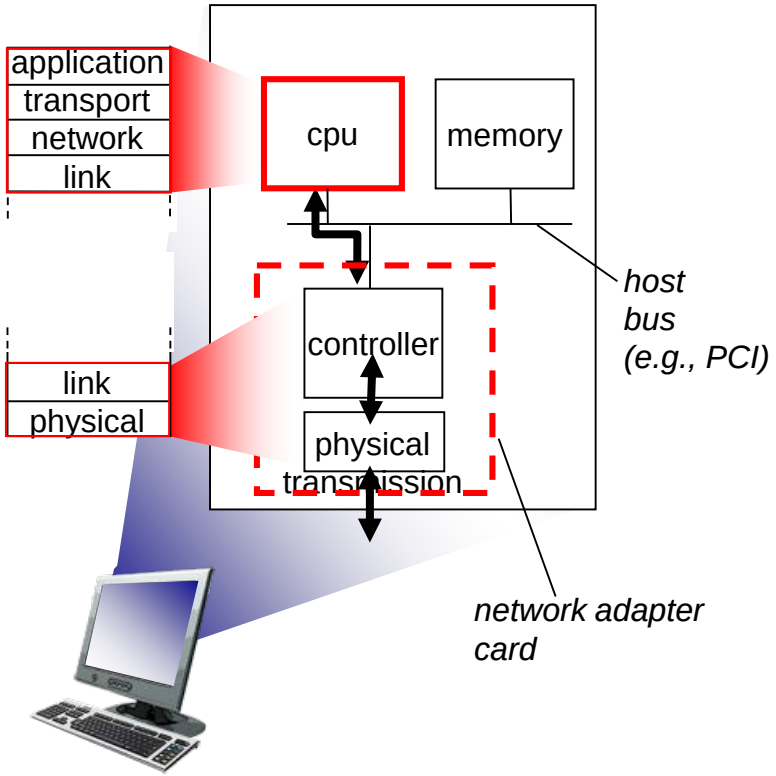
- *Mise en trames, accès au lien:*
  - encapsuler les datagrammes en trames, ajouter l'en-tête, le *trailer*
  - accès au canal si le media est partagé
  - adresses "MAC" dans l'en-tête pour identifier source et destination
    - différentes des adresses IP!
- *transfert fiable entre les nœuds adjacents*
  - similaire au mécanismes de TCP
  - rarement utilisé dans les liens avec taux d'erreurs faible (fibre)
  - liens sans fil: taux d'erreurs élevés

Services de la couche liaison (plus)

- *détection d'erreurs:*
  - erreurs causées par l'atténuation du signal, bruit.
  - le récepteur détecte la présence d'erreurs:
    - demande la retransmission ou élimine la trame
- *correction d'erreurs:*
  - le récepteur identifie *et corrige* les erreurs sur les bits sans demander la retransmission
- *half-duplex et full-duplex*

Couche liaison: implémentation

- dans chaque nœud
- adaptateur ou puce
  - carte Ethernet, carte 802.11; chipset Ethernet
  - implémente couche liaison et physique
- liée aux bus du système
- combinaison de matériel/logiciel



Couche liaison: plan

5.1 introduction, services

5.2 détection d'erreurs, correction

5.3 protocoles d'accès multiple

5.4 LANs

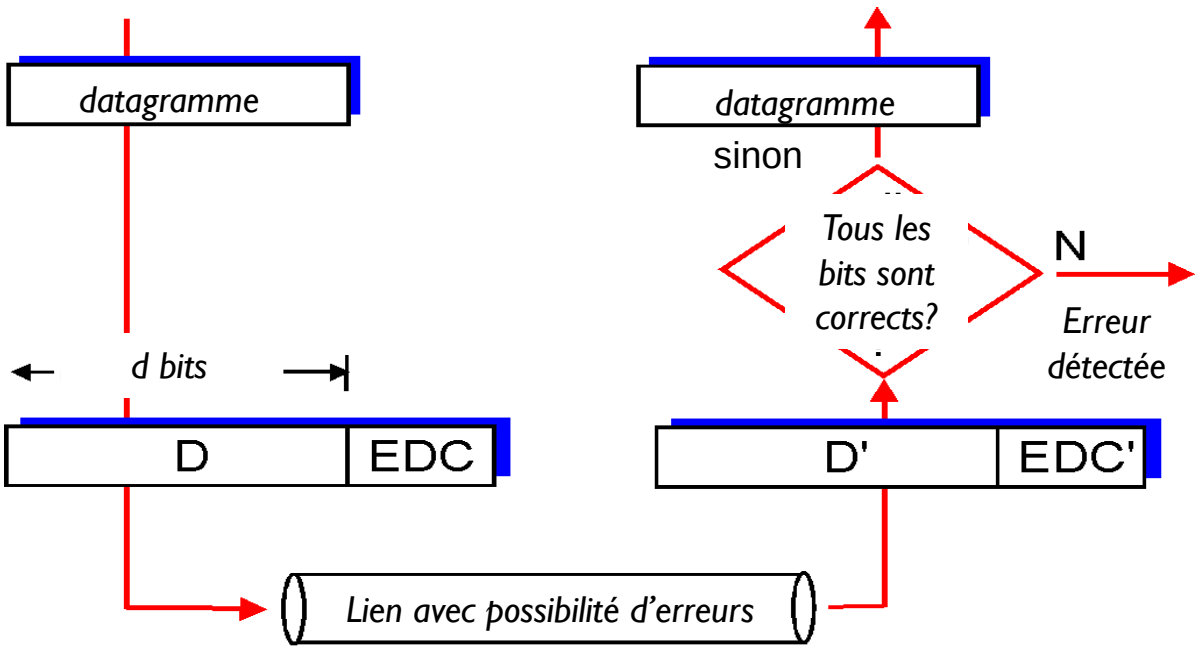
- adressage, ARP
- Ethernet
- commutateurs
- VLANS

5.5 virtualisation: MPLS

Détection d'erreurs

EDC= bits de "*Error Detection and Correction*" (redondance)
D = Données protégées (+ en-têtes)

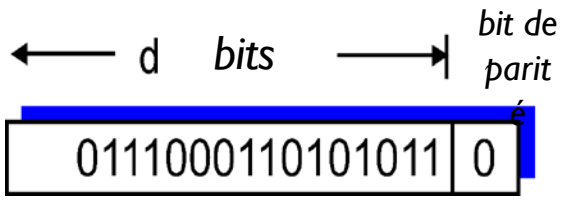
- La détection n'est pas fiable à 100%!
  - le protocole peut ne pas détecter des erreurs, mais rarement
  - si la taille de EDC est grande, alors une meilleure détection et correction



Contrôle de parité

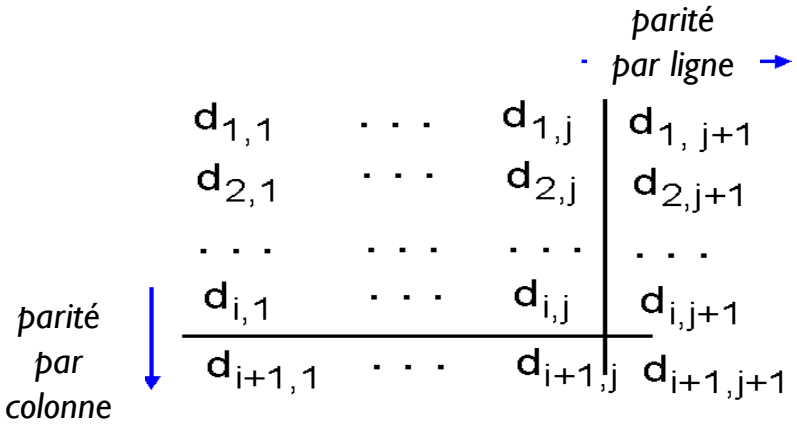
parité avec 1 bit:

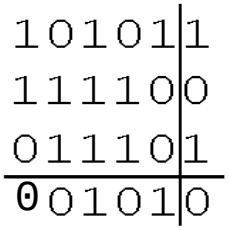
- *détecte si un seul bit est erroné*



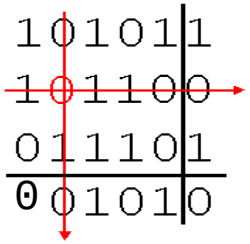
parité à deux dimensions:

- détecte et corrige si un seul bit est érroné





Pas d'erreurs



Erreur de parité

Erreur de parité

Somme de contrôle (rappel)

but: détecter les "erreurs" (ex., bits erronés) dans le paquet transmis (note: utilisé dans la couche transport)

émetteur:

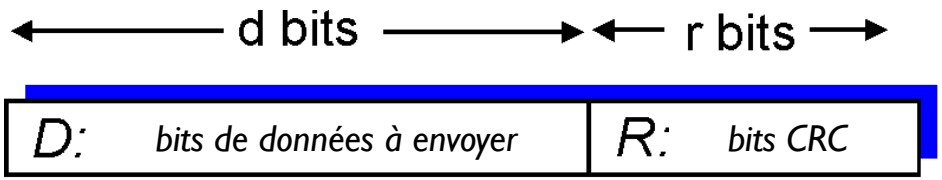
- le segment est considéré comme des entiers de 16-bit
- checksum: complément à 1 de la somme des données du segment
- comme champ de l'en-tête UDP

récepteur:

- calcule le checksum du segment reçu
- vérifie si le checksum calculé est égal à celui reçu:
  - NON – erreur détecté
  - OUI – pas d'erreurs détecté. *toutefois il peut en y avoir?*

Vérification de redondance cyclique

- codage plus puissant pour la détection d'erreur
- les données, D, sont traitées comme un nombre binaire
- l'émetteur et le récepteur se mettent d'accord sur un pattern de r+1 bit (appelé générateur), G
- but: choisir un CRC, R, de r bits tel que
  - <D,R> est divisible par G (modulo 2) (pas de reste)
  - le récepteur divise <D,R> par G. Si il y a reste: erreur détectée!
  - peut détecter toutes les erreurs en rafale de r+1 bits
- très utilisé en pratique (Ethernet, 802.11 WiFi, ATM)



$$D * 2^r \quad \text{XOR} \quad R$$

CRC exemple

on veut:

$(D \cdot 2^r)$ XOR $(R) = nG$

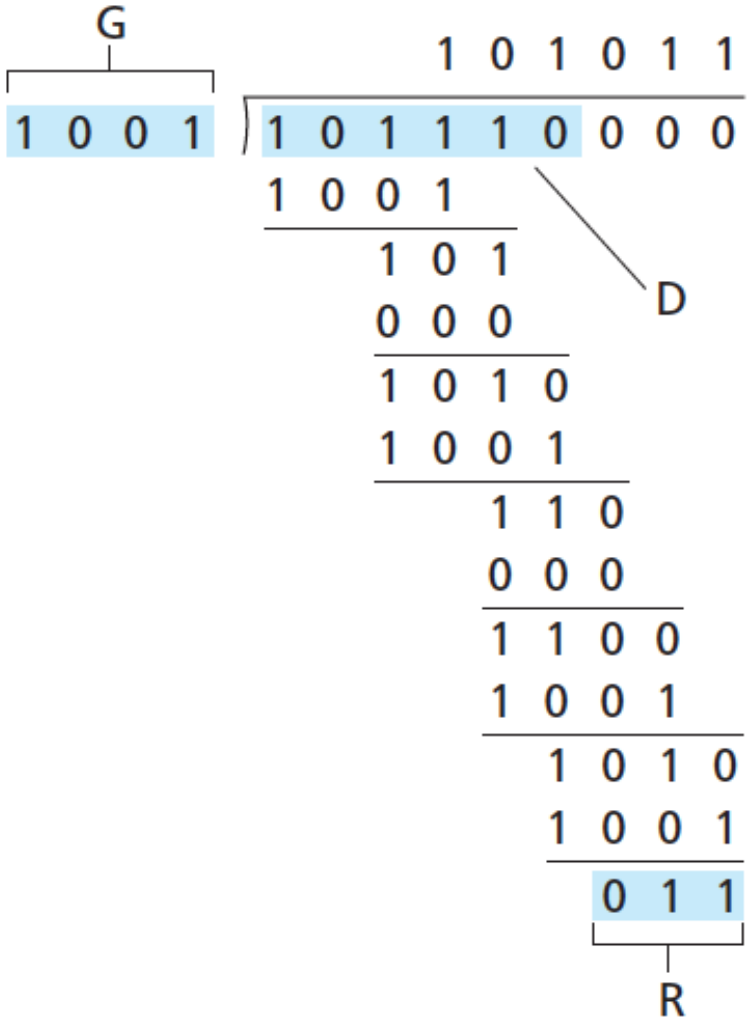
ce qui est équivalent à:

$D \cdot 2^r = (nG)$ XOR (R)

c.-à-d.:

si on divise $D \cdot 2^r$ par G, le reste est R :

$$R = \text{reste}\left[\frac{D \cdot 2^r}{G}\right]$$



Couche liaison: plan

5.1 introduction, services

5.2 détection d'erreurs, correction

5.3 protocoles d'accès multiple

5.4 LANs

- adressage, ARP
- Ethernet
- commutateurs
- VLANS

5.5 virtualisation: MPLS

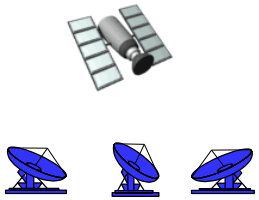
Liens à accès multiple, protocoles

deux types de “liens”:

- point-à-point
  - PPP (ex. accès avec un modem)
  - point-à-point entre un switch Ethernet et un hôte
- *diffusion (média partagé)*
  - Ethernet
  - 802.11 (réseaux locaux sans fil)

câble partagé (ex., Ethernet)

sans fil (ex., 802.11 WiFi)



Satellite

un party (air)

Protocoles d'accès multiple

- un seul canal de diffusion partagé
- si plusieurs transmissions simultanées ➔ interférence
  - *collision* si un nœud reçoit plusieurs signaux en même temps

protocole d'accès multiple

- algorithme distribué qui détermine comment les nœuds se partage le canal
- la communication de l'info concernant le partage doit utiliser le même canal
  - pas de canal hors-bande pour la coordination

Un protocole idéal

Soit: un canal de diffusion de *R* bps

desiderata:

1. quand un seul nœud transmet, il le fait avec un débit *R*.
2. quand *M* nœuds transmettent, chacun transmet avec un débit moyen de *R*/*M*
3. totalement décentralisé:
   - pas de nœud coordonnateur
   - pas de synchronisation
4. simple

Protocoles MAC : taxonomie

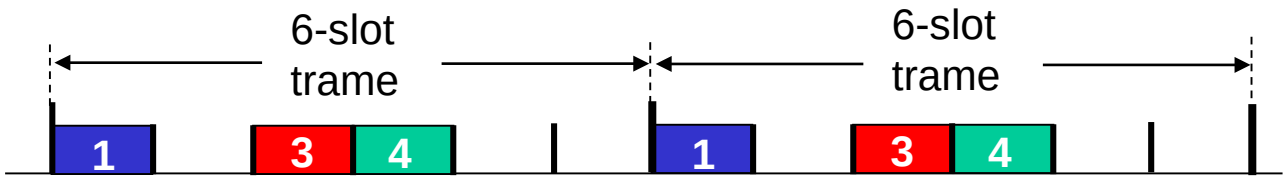
trois types:

- *partitionnement du canal*
  - diviser le canal en petites "morceaux" (temps, fréquence, code)
  - allouer un morceau à un nœud pour une utilisation exclusive
- *accès aléatoire*
  - canal non divisé, possibilité de collisions
  - "récupérer" après les collisions
- *"par tour"*
  - les nœuds transmettent par tour, mais les nœuds ayant plus de données utilisent le canal plus longtemps

Protocoles à partitionnement de canal: TDMA

TDMA: *time division multiple access*

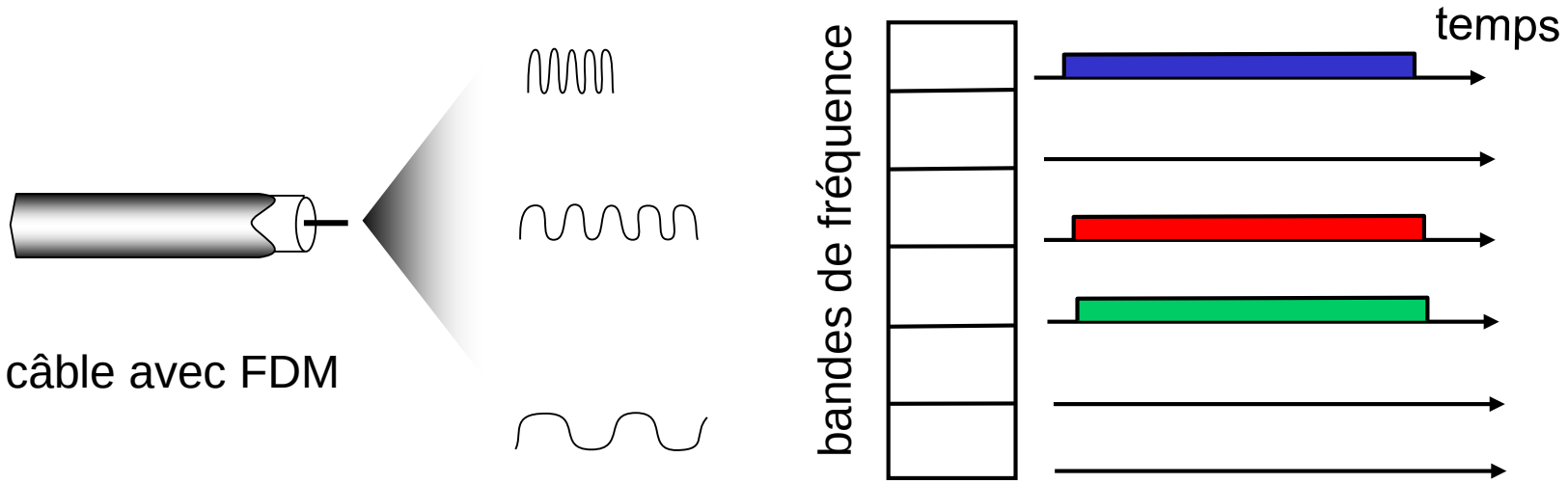
- chaque nœud est assigné un intervalle de temps (slot) fixe pour transmettre un paquet à chaque tour (trame)
- possibilité d'avoir des intervalles inutilisés



Protocoles à partitionnement de canal: FDMA

FDMA: *frequency division multiple access*

- le spectre du canal est divisé en plusieurs bandes de fréquence
- chaque nœud est assigné une bande fixe
- possibilité d'avoir des bandes inutilisées



Protocoles à accès aléatoire

- quand un nœud veut transmettre un paquet
  - il l'envoie au débit total du canal R.
  - pas de coordination entre les nœuds
- plusieurs nœuds qui émettent ➜ "collision",
- le protocole spécifie:
  - comment détecter les collisions
  - comment récupérer après une collision (ex., via retransmissions "retardées")
- exemples de protocoles:
  - *slotted* ALOHA
  - ALOHA
  - CSMA, CSMA/CD, CSMA/CA

Slotted ALOHA

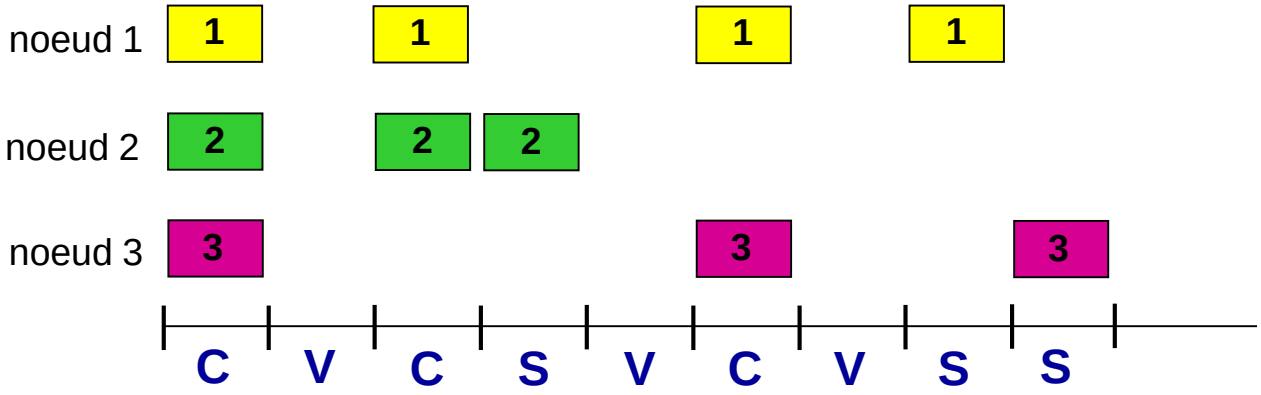
hypothèses:

- trames de tailles égales
- temps est divisé en slots
- transmission au début du slot
- synchronisation
- Tous les nœuds détectent la collision

fonctionnement:

- quand un nœud a une nouvelle trame, il transmet au prochain slot
  - *si pas de collision:* émettre une nouvelle trame au prochain slot
  - *si collision:* retransmettre la trame dans les prochains slots avec probabilité p jusqu'au succès

Slotted ALOHA



Avantages:

- si un seul nœud alors il utilise tout le débit du canal
- totalement décentralisé: synchronisation au niveau des slots seulement
- simple

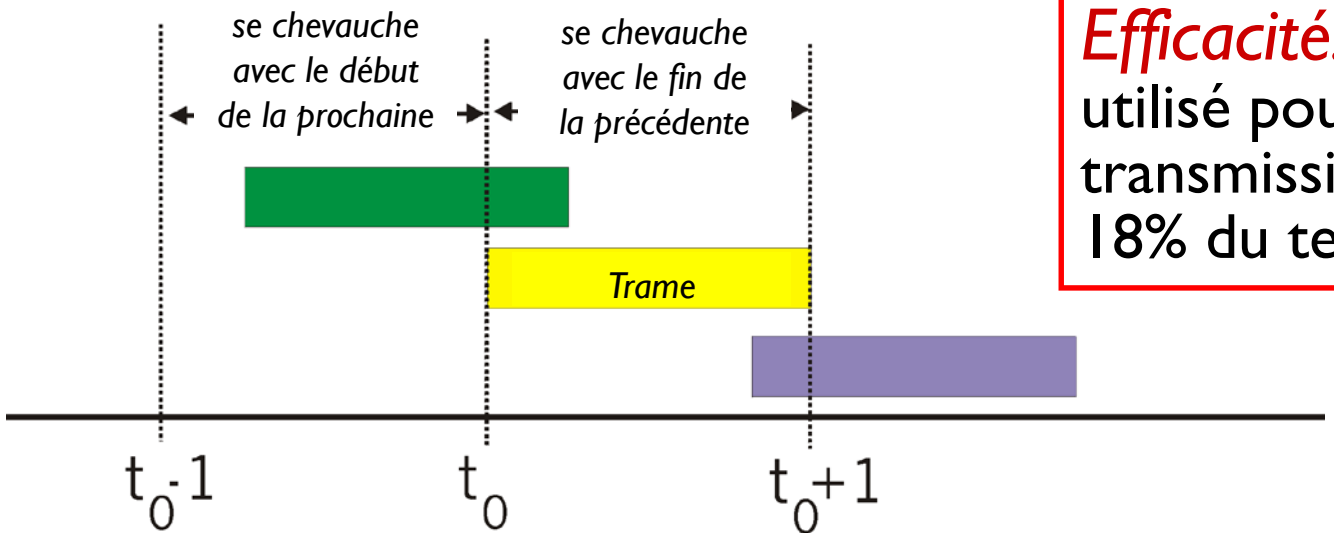
Inconvénients:

- collisions, slots perdus
- slots inoccupés
- besoin de synchronisation

Efficacité: le canal est utilisé pour des transmissions durant 37% du temps!

(*unslotted*) ALOHA pure

- *unslotted* Aloha: plus simple, pas de synchronisation
- quand une trame arrive
  - transmettre immédiatement
- la probabilité de collision croît:
  - trame envoyée à t_0 tombe en collision avec des trames envoyées dans $[t_0-1, t_0+1]$



Efficacité: le canal est utilisé pour des transmissions durant 18% du temps!

CSMA (*carrier sense multiple access*)

CSMA: écouter avant de transmettre:

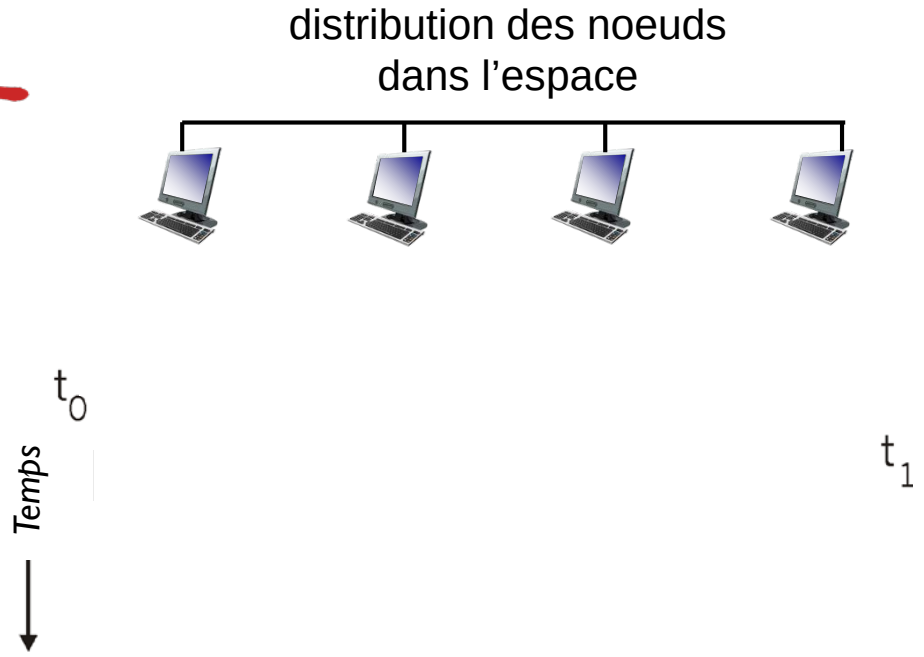
si le canal est libre: transmettre la trame

- si le canal est occupé, la transmission est différé

- analogie: ne pas déranger les autres!

CSMA collisions

- collisions *peuvent toujours avoir lieu*: à cause des délais de propagation deux nœuds peuvent ne pas s'entendre
- collision: tout le temps de transmission d'u paquet est perdu
  - la distance & le délai de propagation jouent un rôle pour déterminer la probabilité de collision



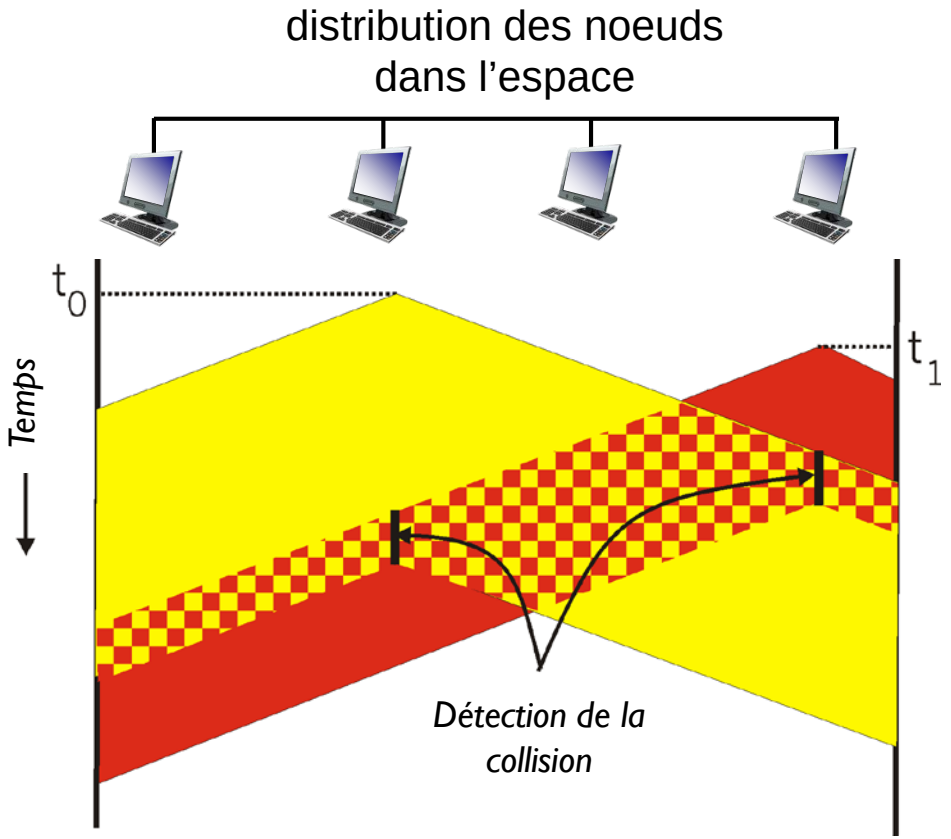
CSMA/CD (détection de collision)

CSMA/CD: détection de porteuse et report de transmission comme pour CSMA

- les collisions sont *détectées* rapidement
- les transmissions en collision sont interrompus, réduit le gaspillage

- détection de collision :
  - simple pour les LAN filaires
  - difficile pour les LAN sans fil
- analogie: une personne polie

CSMA/CD (détection de collision)



Algorithme Ethernet CSMA/CD

1. NIC reçoit un datagramme de la couche réseau, crée une trame
2. si NIC voit que le canal est libre, commence la transmission. si NIC voit qu'il est occupé, attend qu'il se libère, puis transmet.
3. si NIC transmet la trame en entier sans détecter une autre transmission, NIC passe à la suivante !
4. si NIC détecte une autre transmission lorsqu'elle transmettait, elle interrompt sa transmission
5. Après, NIC entre en *backoff binaire:*
   - après *nième* collision, NIC choisit un *K* aléatoire dans $\{0, 1, 2, \ldots, 2^m-1\}$. NIC attend $K \cdot 512$, puis retourna à l'étape 2
   - l'intervalle de backoff devient plus grand après chaque collision

Protocoles MAC “par tour”

Protocoles MAC à partitionnement du canal:

- à grande charge: partage équitable et efficace
- à faible charge: inefficace (chaque nœud dispose de *1/N* de la bande passante même si il y a 1 seul nœud actif!

Protocoles MAC à accès aléatoire

- à faible charge: un nœud utilise tout le canal
- à grande charge: collision

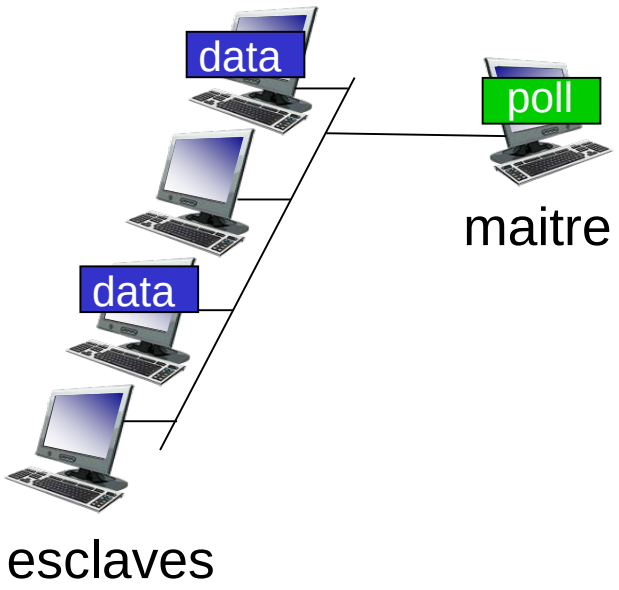
Protocoles "par tour"

combine les avantages des deux!

Protocoles MAC “par tour”

polling:

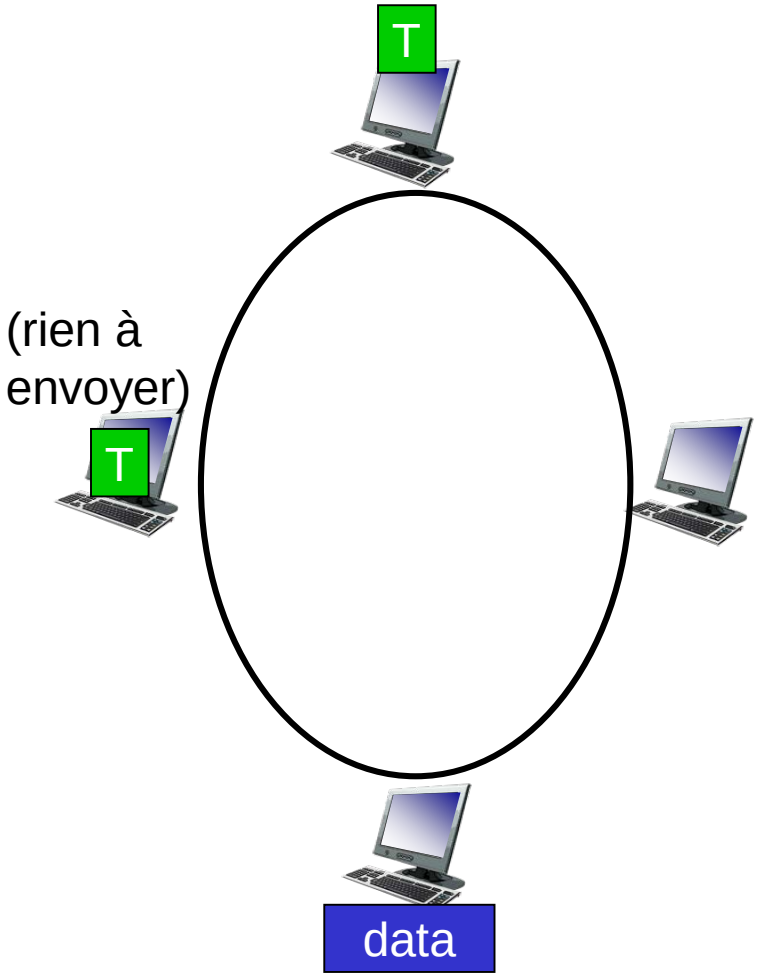
- le nœud maitre “invite” les nœuds esclaves à transmettre par tour
- les nœuds esclaves sont "stupides"
- problèmes:
  - *overhead* pour l’invitation (*polling)*
  - plus de délai
  - le système dépend fortement du maitre



Protocoles MAC "par tour"

passement du jeton:

- un jeton est passé d'un noeud à un autre d'une manière séquentielle.
- *"token message"*
- problème:
  - overhead pour le passage du jeton
  - plus de délai
  - le système dépend fortement du jeton



Résumé des protocoles MAC

- *partitionnement du canal,* par temps, fréquence ou code
  - TDMA, FDMA
- *accès aléatoire* (dynamique),
  - ALOHA, S-ALOHA, CSMA, CSMA/CD
  - détection de porteuse: simple pour certaines technologies (filaire), difficile dans d'autres (sans fil)
  - CSMA/CD utilisé dans Ethernet
  - CSMA/CA utilisé dans 802.11
- *par tour*
  - polling maitre-esclave, passement du jeton
  - bluetooth, FDDI, token ring

1 1 0 1 0 1 0 0 0 1 1 0 1 0 1 1
0 0 1 0 1 0 0 0 1 0 1 1 1 0 0 1
1 1 1 0 0 0 1 0 1 0 0 1 0 0 0 1
1 0 0 1 1 0 1 0 1 1 0 0 0 0 0 0
0 1 0 0 0 1 1 1 1 0 1 0 1 1 0 0

Utilisons la parité impair

1 1 0 1 0 1 0 0 0 1 1 0 1 0 1 1 0
0 0 1 0 1 0 0 0 1 0 1 1 1 0 0 1 0
1 1 1 0 0 0 1 0 1 0 0 1 0 0 0 1 0
1 0 0 1 1 0 1 0 1 1 0 0 0 0 0 1 1
0 1 0 0 0 1 1 1 1 0 1 0 1 1 0 0 1
0 0 1 1 1 1 0 0 1 1 0 1 0 0 0 0 0

Détecter le bit erroné

Exemples CRC

- Données: 10011000
- G: 1001

- Données: 11010111011
- G: 10011