Département Telecom & RSI – 1ère année

Module RE110 & RE111: Introduction aux Réseaux

Introduction aux Réseaux – 2ème partie

Couche Liaison de Données

A. NAHALI

Plan

- **Couche Liaison de Données:**
  - **Pré-requis**
  - Commutation
  - Principes (contrôle de flux, protocoles, contrôle d'erreur, …)

Pré-requis – Nature de l'information

- Nature de l'information:
  - Données discrètes: suite d'éléments appartenant à un ensemble dénombrable
    - Un texte, suite de mots eux-mêmes composés de lettres
  - Données continues: résultant de la variation continue d'un phénomène physique
    - Voix, image, etc.

- Pour être traitée par des équipements informatiques, l'information doit être représentée comme suite de symboles binaires {0,1}

Pré-requis - Bit, Byte

- Bit: Binary Digit
- 1 octet (1 Byte) = 8 bits
- La capacité de stockage d'une machine est exprimée en octet (Byte)

	bit
1Kbit (Kilobit)	**10^3**
1Mbit (Megabit)	**10^6**
1Gbit (Giga bit)	**10^9**
1Tbit (Tera bit)	**10^{12}**

Pré-requis - Débit

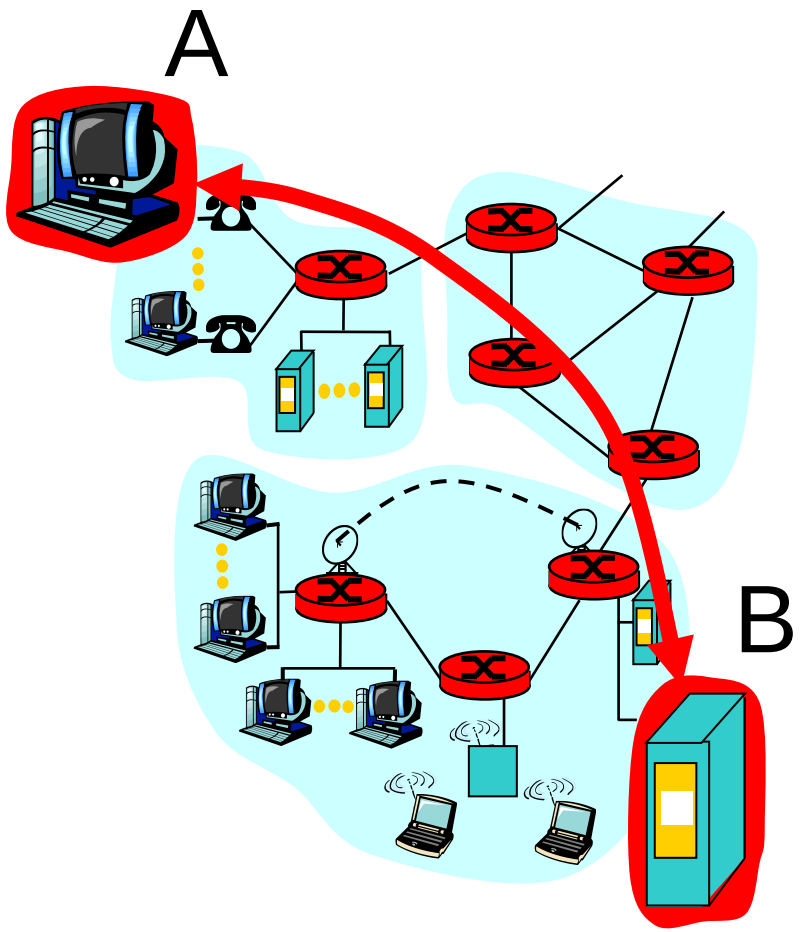
- Les bits représentant l'information sont émis sur le support de transmission sous forme de signal électrique ou électromagnétique
- Le débit binaire est le nombre de bits émis sur le support de transmission durant une unité de temps (1 sec)
- Le débit binaire maximal d'un support de transmission (ou **capacité** de transmission) exprime la quantité d'information maximale transportée par unité de temps (1 sec)

Plan

- **Couche Liaison de Données:**
  - Pré-requis
  - **Commutation**
  - Principes (contrôle de flux, protocoles, contrôle d'erreur, …)

Commutation

- Comment un hôte émetteur A peut envoyer de l'information à un hôte récepteur B?



Commutation

- Différents hôtes doivent pouvoir transmettre de l'information de manière concurrente:

⇨ Partager les ressources du réseau

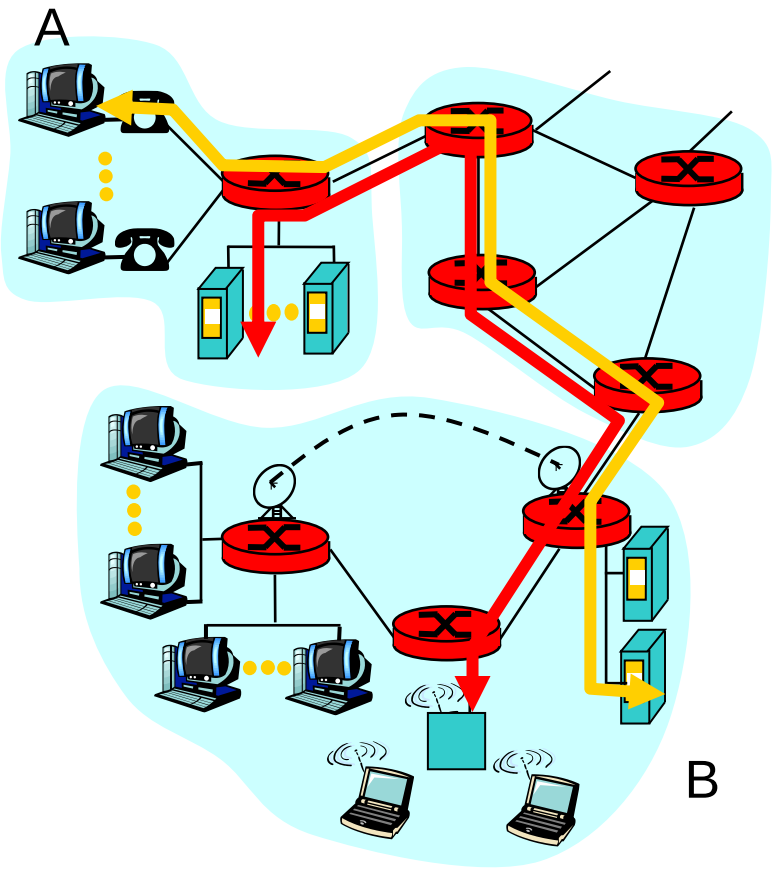
- Deux approches:
  - Réseaux à commutation de circuits
  - Réseaux à commutation de paquets

Commutation de circuits

1. Etablissement d'un circuit physique propre à la communication entre l'émetteur A et le récepteur B avant de transmettre les données

2. Réservation de l'ensemble des ressources pour toute la période de la transmission

3. Libération des ressources une fois le transfert de données terminé



Commutation de circuits

Implémentation du circuit, par exemple:

- Multiplexage fréquentiel (Frequency-division multiplexing)
  - Allocation d'une fraction de la bande passante à chaque utilisateur
- Multiplexage temporel (Time-division multiplexing)
  - Division du temps d'utilisation en intervalles de temps de même largeur et allocation de chaque intervalle de temps à une communication particulière

Commutation de circuits

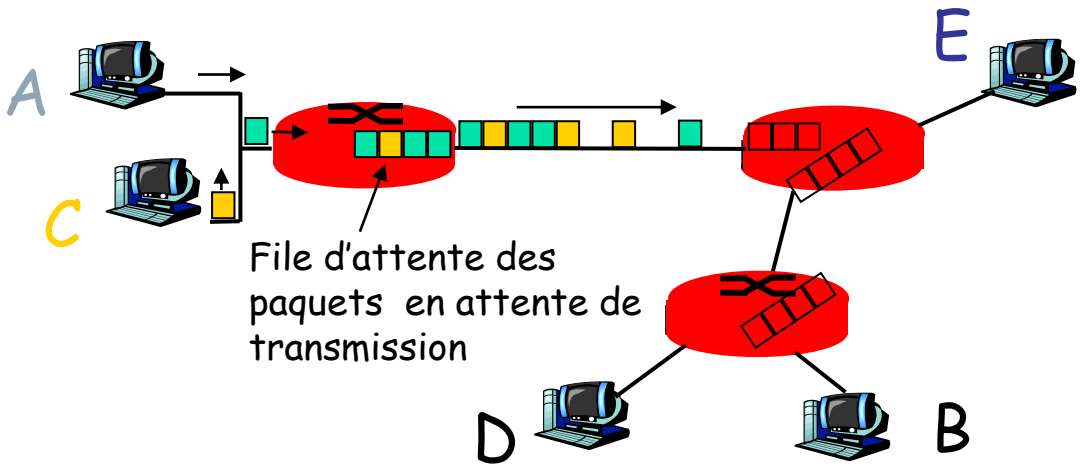
- ☹ Inconvénients
  - Temps d'établissement long
  - Mauvaise utilisation du support
  - Une fois que un émetteur a son circuit, dans les moments où il n'envoie pas des information, les ressources ne sont pas utilisé par quelqu'un d'autre

- ☺ Avantages
  - Temps de transmission constant
  - Pas de congestion

- Exemple : téléphonie analogique

Commutation de paquets

- L'information est découpée en paquets
- Impose une taille limite maximale aux paquets
- Chaque paquet est transmis sur un lien à plein taux de transmission



Commutation de paquets

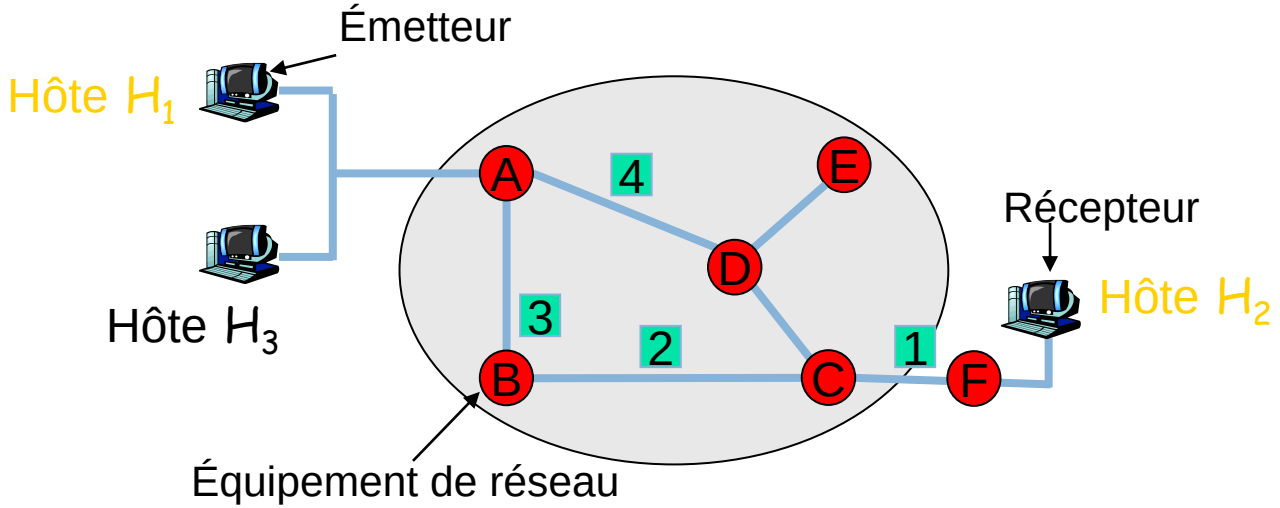
- 2 types de réseaux à **commutation de paquets**:

1. Réseaux à datagrammes:
   - Ni établissement d'un chemin logique ni réservation de ressources (chaque paquet est géré indépendamment des autres)
   - Ex : Internet (IP)

2. Réseaux à circuits virtuels:
   - Etablissement d'un circuit logique entre les deux extrémités communicantes mais pas de réservation de ressources
   - Ex: X25, ATM

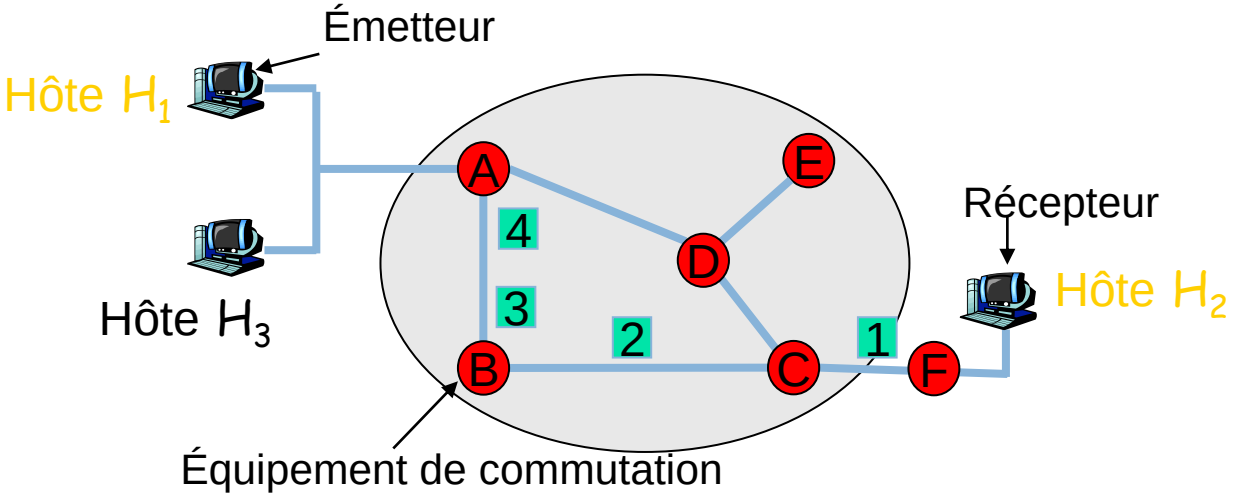
Réseau à datagramme

- Chaque paquet contient l'adresse de l'hôte destination
- Chaque équipement de réseau décide de l'acheminement des paquets, paquet après paquet ⇨ les paquets peuvent être acheminés au travers de chemins différents



Réseau à circuit virtuel

- Motivation : éviter de devoir choisir une route pour chaque paquet
- Avant d'envoyer le premier paquet, on établit une route entre l'émetteur et le récepteur:
  1. L'émetteur établit une connexion avec le récepteur (création d'une entrée dans la table de circuit virtuel au sein de tous les équipements de réseau sur la route)
  2. L'émetteur transfert les données
  3. La connexion est fermée (effacement de l'information correspondante à la connexion dans toutes les tables de circuit virtuel des équipements sur la route)



Réseaux à commutation de paquets

- ☹ Inconvénients :
  - Introduction de retards variables et imprévisibles
  - L'hôte doit assembler les paquets pour reconstruire l'information

- ☺ Avantages : plus efficace, support mieux utilisé que la commutation de circuits

- Exemple : Internet

Pourquoi fragmenter l'information en paquets?

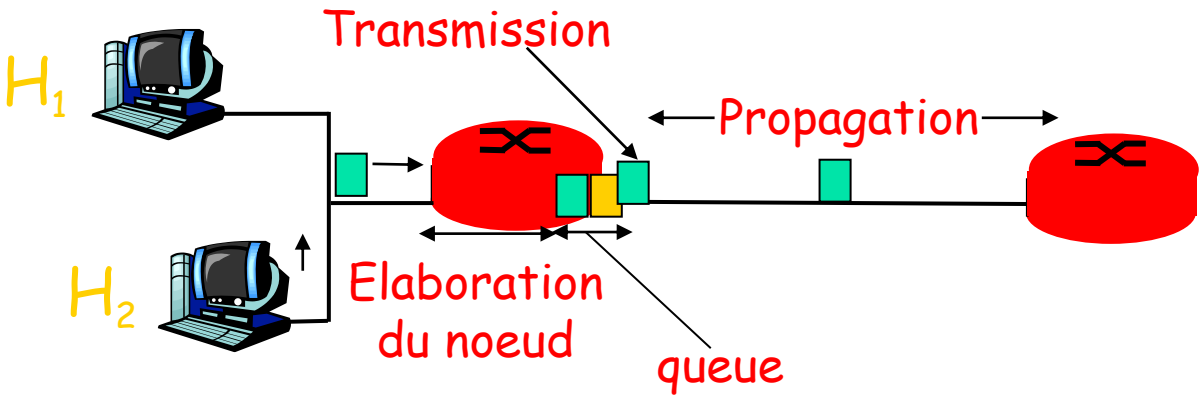
- Fragmenter les données en paquets permet de transmettre en parallèle différents paquets

⇨ Réduit le temps de transfert des données

- La commutation de paquets est l'évolution de la commutation de messages

Temps de transfert dans les réseaux à commutation de paquets

- Temps Total = Tps transmission+Tps propagation+Retards
- Temps de transmission
  - R = débit de la transmission (bps)
  - L = taille du paquet (bit)
  - Tps transmission (= Temps pour transmettre le paquet sur le lien) = L/R
- Temps de propagation
  - s = vitesse de propagation sur le support physique (~2,7x10^8 m/sec pour cuivre)
  - d = longueur lien physique
  - Tps de propagation = d/s



Commutation de messages

- Long messages transmis sans être divisés en paquets de taille + petite
- Les premiers systèmes électromécaniques de télécommunications faisaient appel à la commutation de messages pour transmettre des télégrammes
- Non utilisé dans les réseaux modernes

Plan

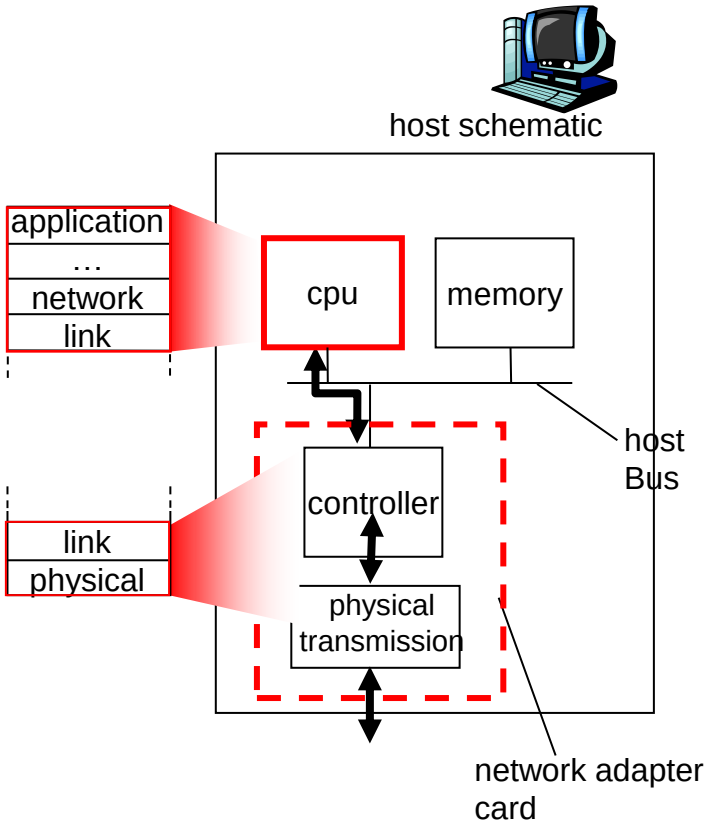
- **Couche Liaison de Données:**
  - Pré-requis
  - Commutation
  - **Principes (contrôle de flux, protocoles, contrôle d’erreur, …)**

Principes de la Couche Liaison de Données

1. Offrir une interface simple à la couche réseau
2. Réguler le flux de données pour éviter que des destinataires lents ne soient submergés par des expéditeurs rapides
3. Détecter et corriger les erreurs de transmission
4. Garantir la fiabilité de livraison
5. Gérer l'accès au support partagé dans les réseaux à diffusion

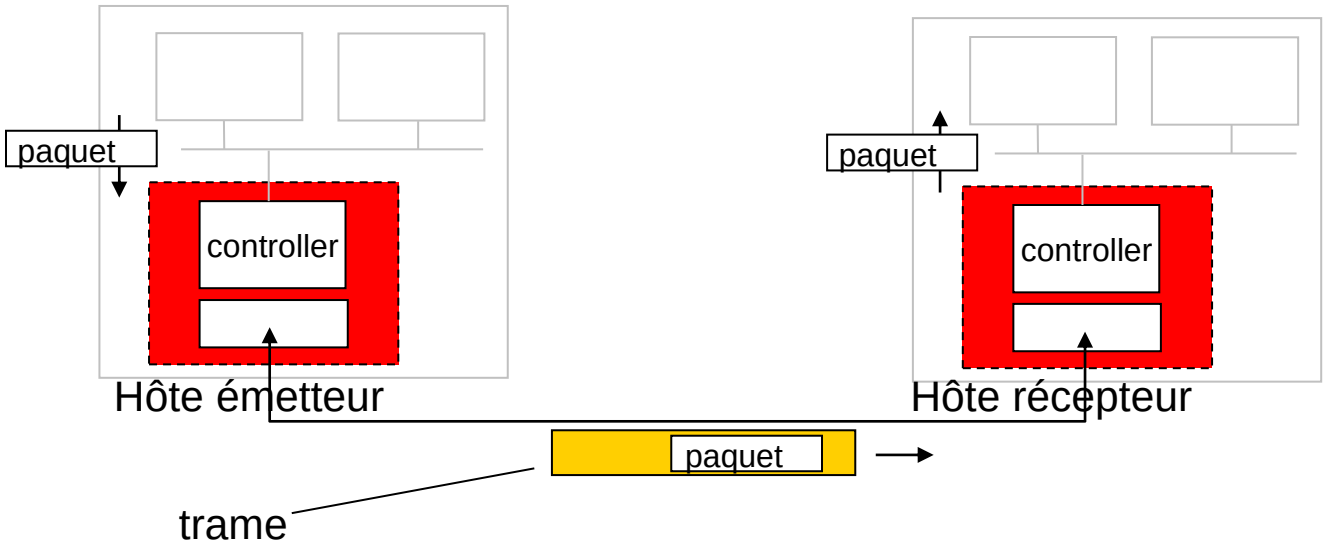
Où est mise en œuvre la Couche Liaison?

- La majeur partie des fonctionnalités de la couche liaison est mise en œuvre dans la carte réseau (NIC)
  - Carte Ethernet, carte WiFi, carte PCMCIA, …



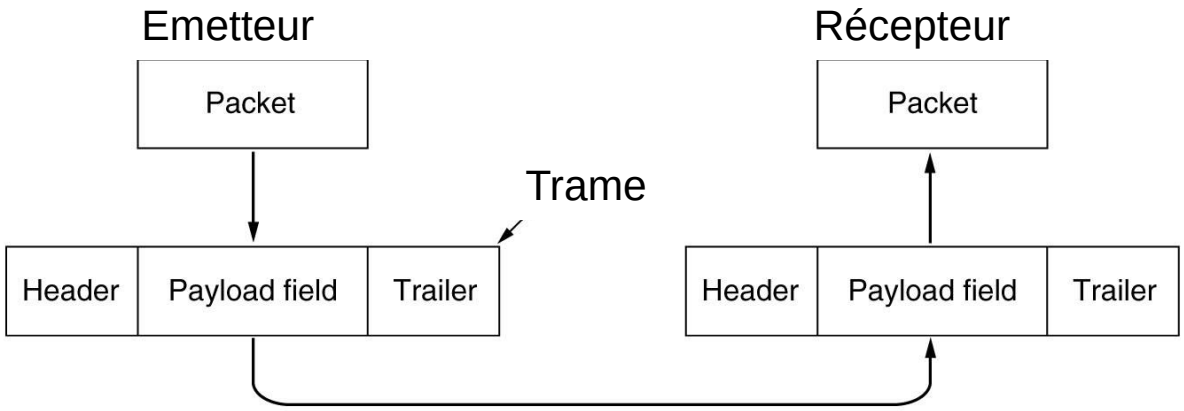
Service vers la couche réseau

- Transfert des paquets de la couche réseau de la machine émettrice à la couche réseau de la machine réceptrice
  - La couche liaison prend les paquets venant de la couche réseau et les encapsule dans des *trames* de transmission



Trame

- Chaque trame contient:
  - Un en-tête de trame (header)
  - Un champ de donnée pour héberger le paquet (payload)
  - Une terminaison de trame (trailer)



Délimitation de trames

- Pour fournir un service à la couche réseau, la couche liaison utilise le service de la couche physique :
  - Transport du flux de bits sur le support et remise à la machine destination

- Les bits peuvent être perdus ou changés ⇨ Pour gérer les erreurs correspondantes, la couche liaison divise le train de bits en trames contenants des *informations de contrôle* pour détecter/corriger les erreurs

Contrôle de flux

- Gérer un émetteur qui transmet trop vite des trames que le récepteur ne réussit pas à traiter
- La couche liaison du récepteur doit souvent gérer plusieurs lignes en entrée ⇨ retard entre la réception de la trame et son traitement
- **Contrôle de flux avec retour d'information** : le récepteur envoie des informations à l'émetteur pour lui donner la permission de transmettre ou pour lui donner des informations sur l'état de la réception

Contrôle d'erreur et fiabilité

- Le récepteur envoie un *acquittement* (ack) pour dire que la trame a été reçue correctement
- Possibilité de *non acquittement* (nack) pour demander la retransmission de la trame
- Numéro de séquence pour distinguer les retransmissions des originaux
- Temporisateur pour retransmettre dans le cas où l'acquittement/non acquittement ou la trame sont perdus

Protocoles
pour le contrôle de flux / contrôle d'erreur

- Protocoles « sans contrôle de flux/d'erreur »
- Protocoles « avec acquittement et retransmission »
  - PAR: Positive Acknowledgment and Retry
- Protocoles « à fenêtre d'anticipation »
  - Avec retour arrière: « Go-Back n »
  - Avec rejet sélectif: « Selective Repeat »

Protocoles « sans contrôle de flux/d’erreur »

- Solution de base d'un protocole sans connexion qui se contente d'acheminer des trames et laisse aux niveaux supérieurs toutes les tâches.

- Pas d’acquittement.

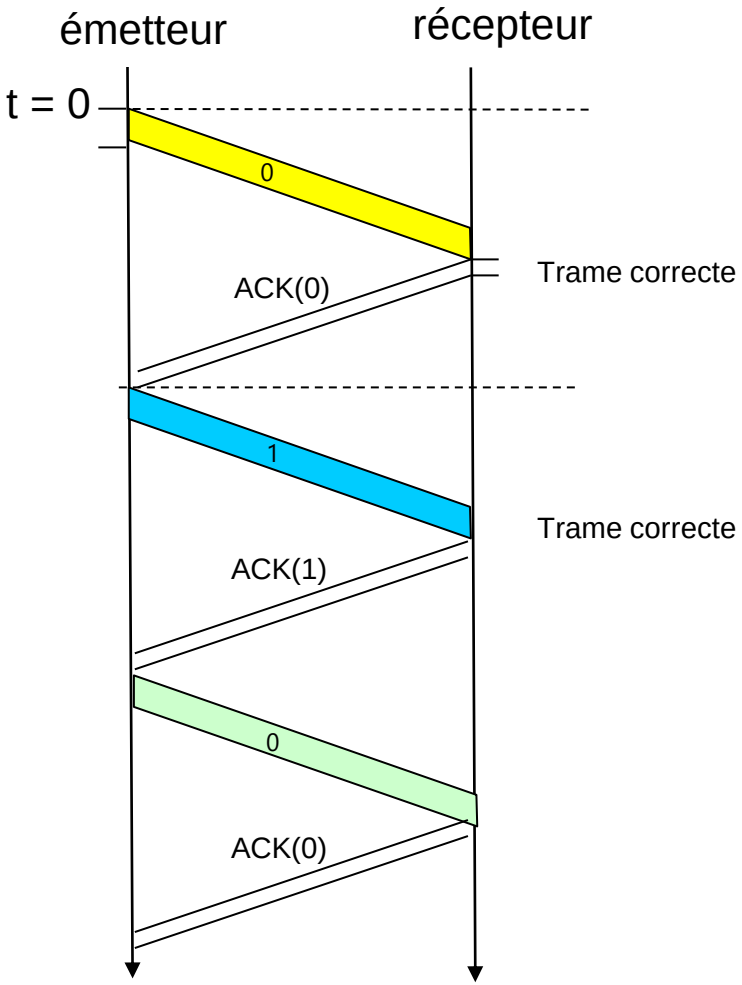
Protocoles « avec acquittement et retransmission »

- PAR: Positive Ack & Retransmission
- L'émetteur envoie une trame τ et attend l'ack avant de transmettre la prochaine trame
- Si le canal est bruité et la trame reçue est erronée, le récepteur demande la retransmission (pas de ACK ou un ACK contenant le numéro de séquence de la dernière trame correctement reçue)
- À la réception d'un ACK, l'émetteur retransmet la trame suivant le numéro de séquence de l'ack

Protocoles « avec acquittement et retransmission »

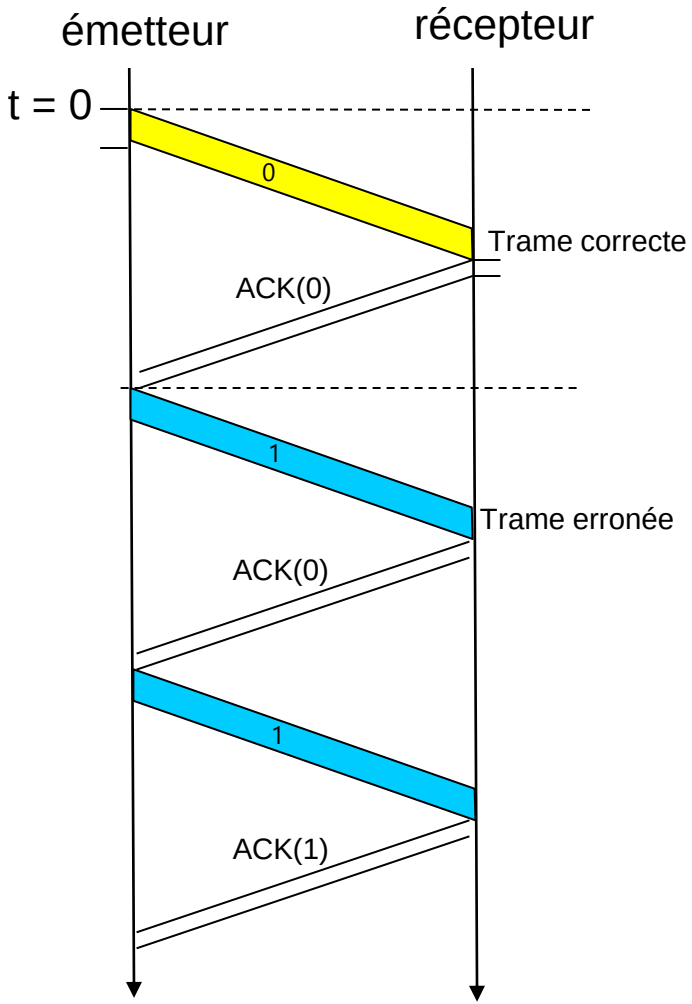
- Pour gérer la perte d'information du canal, l'émetteur utilise un temporisateur pour savoir quand renvoyer la trame:
  - Après la transmission de la trame, l'émetteur arme un temporisateur (s'il est déjà armé, cela provoque son réarmement)
  - La temporisation doit durer assez longtemps pour permettre à la trame d'arriver à la destination, d'être traitée et pour permettre à l'émetteur de recevoir l'acquittement

PAR: scenario sans erreurs ni pertes



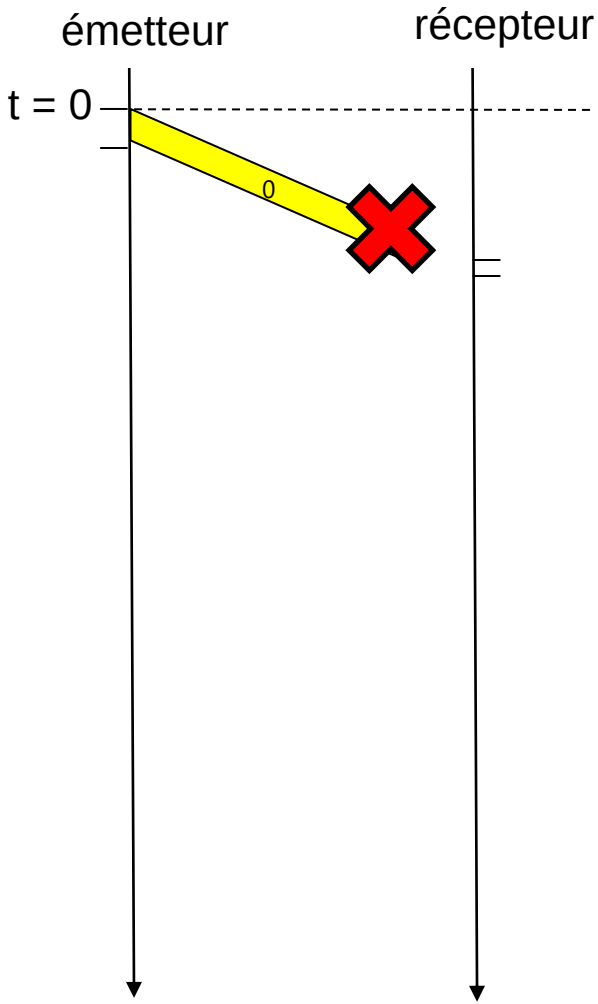
- Toutes les trames (d'information et d'acquittement) sont reçues correctement
- L'émetteur envoie 3 trames
- Aucune trame n'est retransmise
- Le récepteur comprend que la 3ème trame est une nouvelle trame et non une retransmission de la première, même si les deux trames ont le même numéro de séquence:
  - Le récepteur stocke localement le numéro de séquence de la dernière trame correctement reçue (1 dans ce scenario avant la réception de la 3ème trame)

PAR: scenario avec erreurs sans pertes



- Toutes les trames d'acquittement sont reçues correctement
- La deuxième trame d'information envoyée est reçue avec des erreurs:
  - Le récepteur envoie un acquittement avec numéro de séquence 0 pour dire à l'émetteur que la dernière trame correctement reçue est celle avec numéro de séquence 0
  - L'émetteur retransmet donc la deuxième trame
- La deuxième trame est reçue correctement la deuxième fois

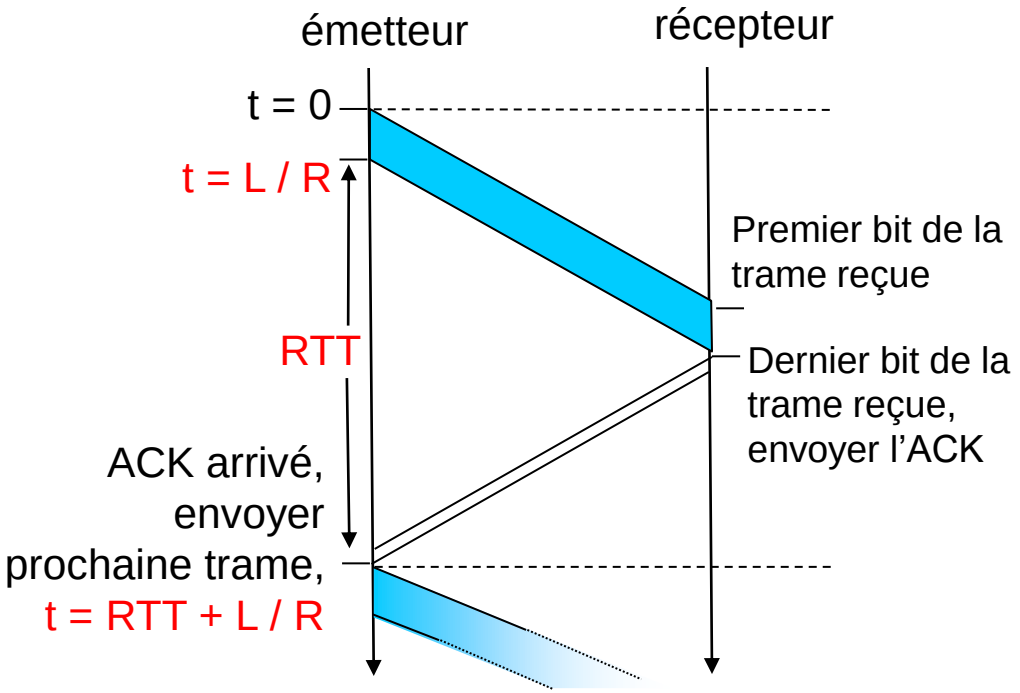
PAR: scenario sans erreurs avec pertes



- La trame d'information est perdue

PAR: Temporisateur

- Si le débit du canal est R bits/sec, la taille de la trame est L bits et le délai de propagation aller-retour de RTT:
  - Il faut L/R secondes pour la transmission de la trame et RTT seconde au moins pour que l'acquittement parvienne à l'émetteur
- Taux d'utilisation du canal= L/(L+R·RTT)



Protocoles à fenêtre d'anticipation

- L'émetteur peut émettre au plus N trames en pipeline (**fenêtre d'émission**)
  - Les trames émises possèdent un numéro de séquence variant de 0 a 2^n-1 (codé donc sur *n* bits)
- Le récepteur à chaque instant détient la liste des numéros de séquence des trames que l'on attend en réception (**fenêtre de réception)**
- Utilisation de mémoires tampon

Protocoles à fenêtre d'anticipation – Go-Back-N

Avec retour arrière: Go-Back-N

- L'émetteur peut émettre au plus N trames en pipeline
- Le récepteur envoie des acquittements cumulatifs :
  - Un ack pour une trame avec numéro de séquence *n* signifie que toutes les trames avec numéro de séquence ≤ n ont été correctement reçues par le récepteur
- L'émetteur maintient un temporisateur pour la plus vielle trame non acquittée
  - Au timeout, l'émetteur retransmet toutes les trames déjà transmises et non acquittées
  - Le temporisateur est ré-initialisé si un acquittement est reçu

Go-Back-N : scenario

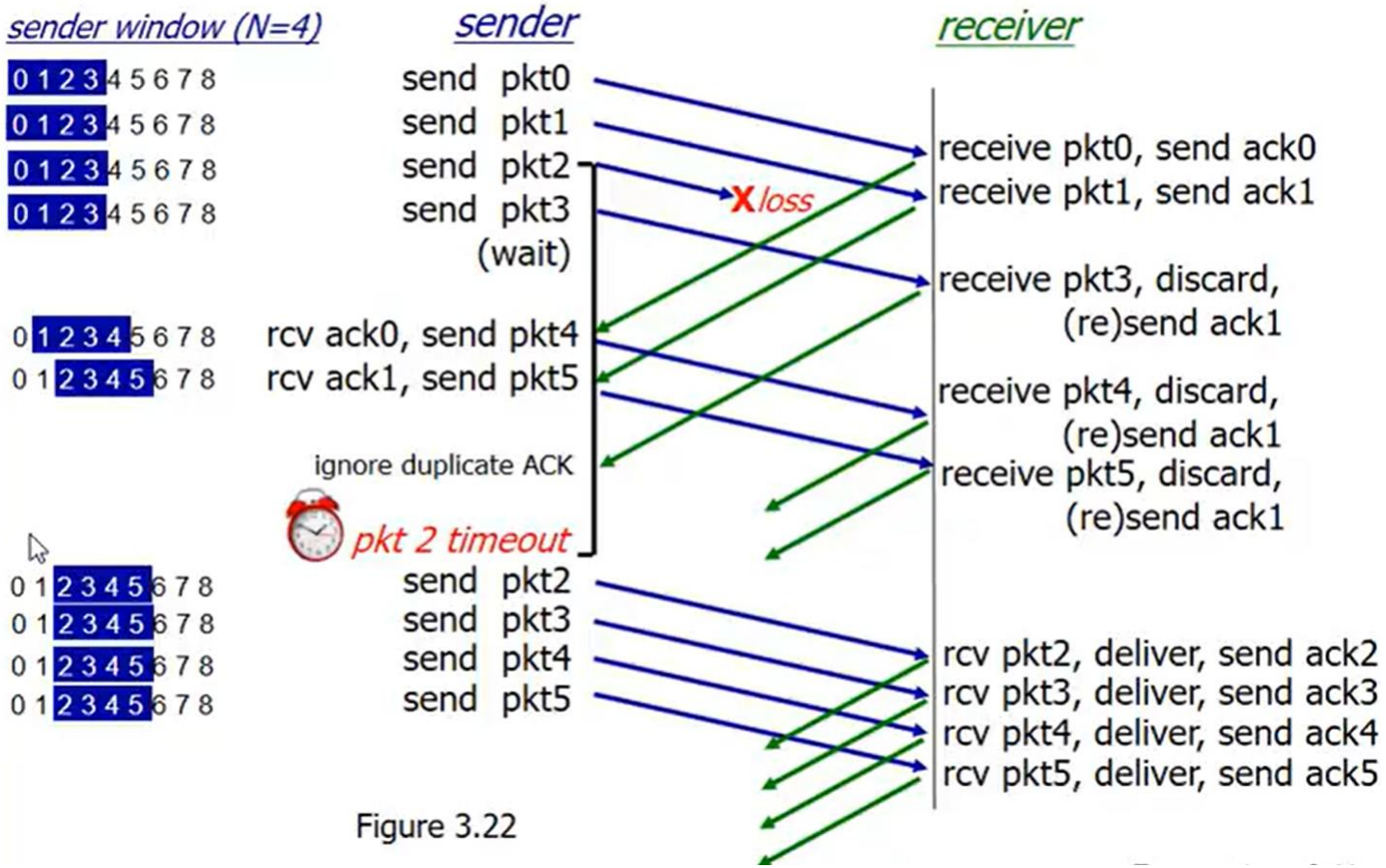


Figure 3.22

Protocoles à fenêtre d'anticipation – Go-Back-N

Avec retour arrière: Go-Back-N

Problème de Go-back-n :

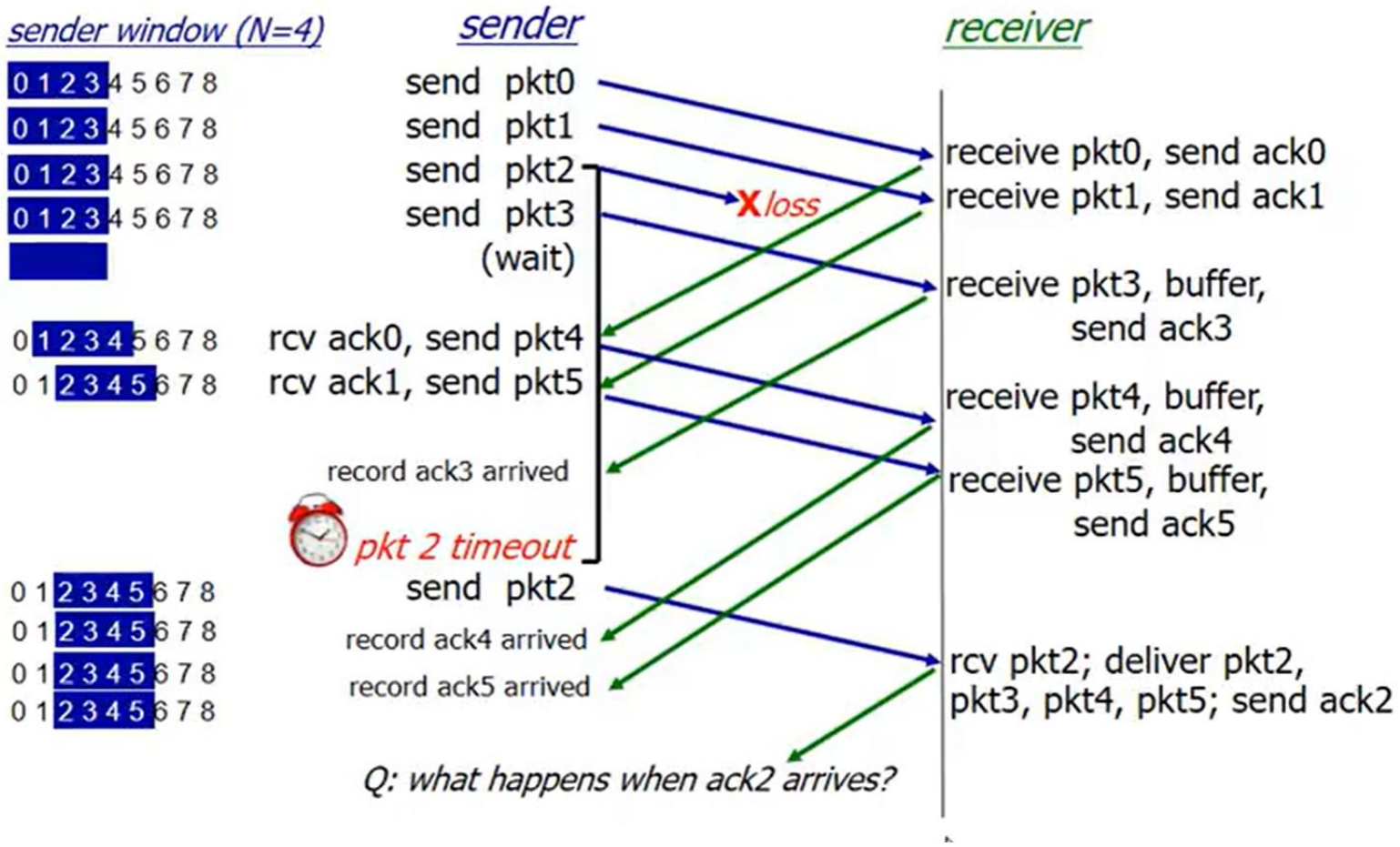
- Une erreur dans une trame provoque la retransmission de toutes les trames à partir de celle erroné ;
- Le récepteur n'accepte pas de trames hors séquence ;

Protocoles à fenêtre d'anticipation – Selective Repeat

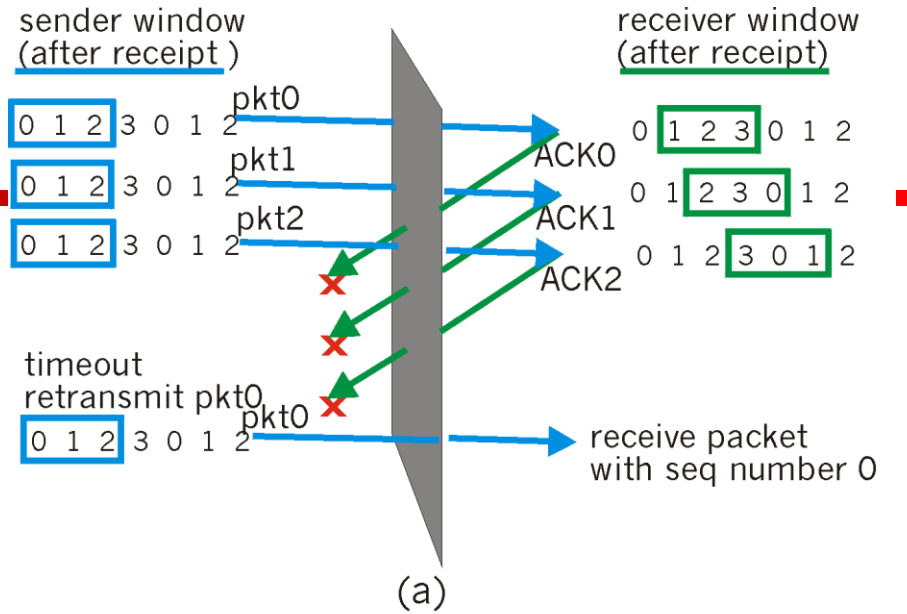
Avec rejet sélectif: Selective Repeat

- L'émetteur peut émettre au plus N trames en pipeline
- Le récepteur envoie un acquittement individuel pour chaque trame correcte
  - Les trames correctes qui ne sont pas ordonnées sont mises dans un tampon mémoire
- L'émetteur maintient un temporisateur pour chaque trame non acquittée
  - Timeout: renvoie seulement de la trame correspondante

Selective Repeat : scenario



Selective Repeat : dimension de la fenêtre

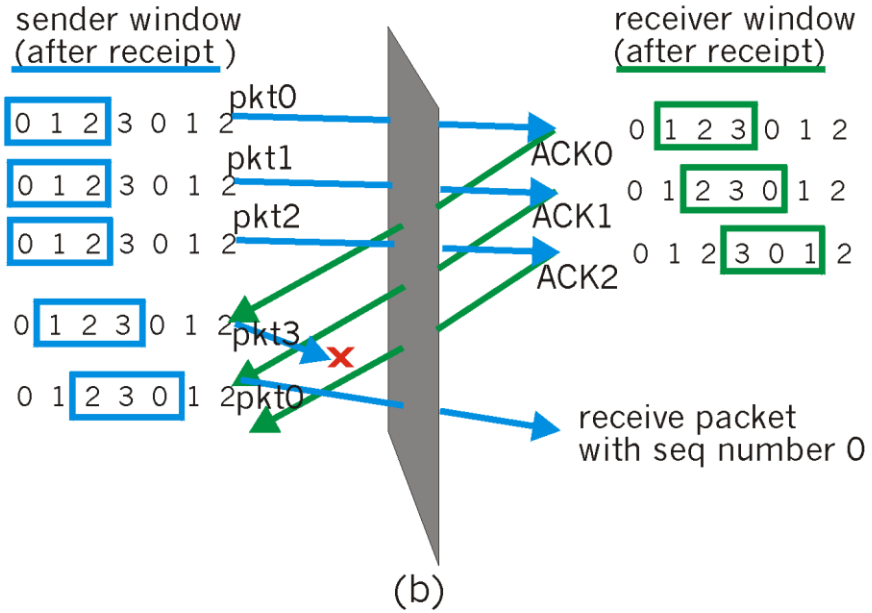


Exemple :

- numéros de séquence N: 0, 1, 2, 3
- Taille de la fenêtre W=3

- Le récepteur ne distingue pas les deux scénarios

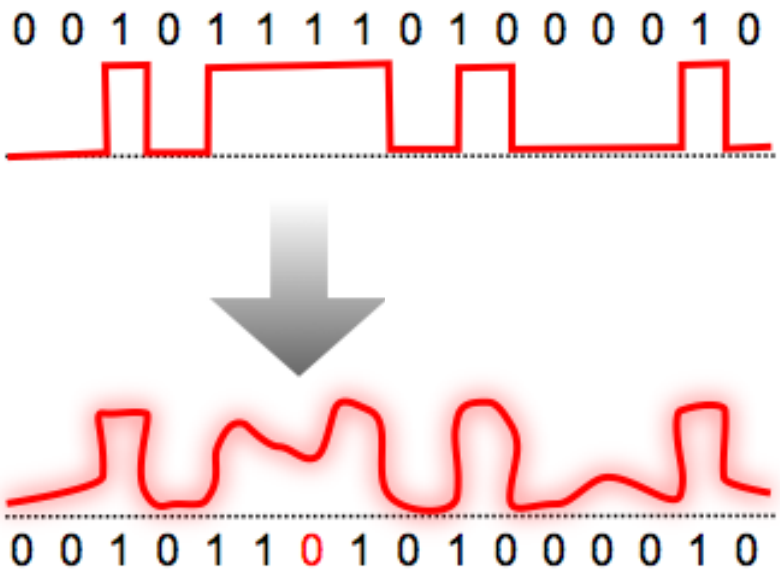
- Dans le scénario (a), le récepteur passe des copies de données à la couche réseau en les considérant nouvelles =>**N>=2 W**



Contrôle d’erreur

- Les lignes de transmission ne sont pas parfaites ;
  - Erreurs de transmission d’informations binaires ;
- A la réception, une erreur d'interprétation peut se produire;
  - Un signal ‘1’ est interprété comme si c'était le ‘0’ et vice versa ;
- Deux types :
  - Erreurs isolées (un bit erroné dans une séquence correcte) ;
  - Erreurs groupés (un nombre important de bits erronés regroupé dans une courte séquence) ;

Contrôle d'erreur



- Le bruit et les interférences peuvent générer des erreurs dans les bits transmis
  - Parfois dans les câbles, souvent dans les transmissions sans fils
- On ajoute aux paquets des codes pour la détection et la correction des erreurs

Code détecteur : Code Polynomial ou Contrôle de la Redondance Cyclique

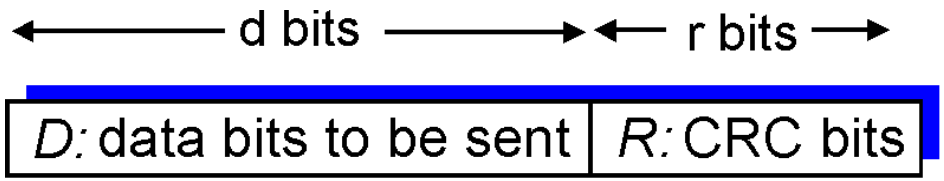
Appelé aussi CRC (Cyclic redundancy code)

- un moyen de contrôle d'intégrité des données puissant
- Facile à mettre en œuvre.
- Il représente la principale méthode de détection d'erreurs utilisée dans les télécommunications.
- On considère que les bits d’une chaîne de caractères sont des coefficients d’un polynôme

Code détecteur : Code Polynomial ou Contrôle de la Redondance Cyclique

- Les bits D d'une chaîne de caractères sont considérés des coefficients d'un polynôme:
  - Un bloc de k bits est vu comme un polynôme de degré k-1, le bit le plus à gauche est le coefficient de x^{k-1}
    - Exemple: 110001⇨ x^5+x^4+1
- L'émetteur et le récepteur se mettent d'accord sur un polynôme générateur G(x)
  - Suite de r+1 bits ayant le bit de poids fort et le bit de poids faible égaux à 1

  ⇨ G(x) est de degré r
- L' émetteur choisit une suite R de r bits à ajouter aux d bits de la suite D tel que la suite de d+r bits représente un polynôme divisible pour G(x)
- Le récepteur vérifie si la suite de bits reçus est divisible par G(x)



Code détecteur : Code Polynomial ou Contrôle de la Redondance Cyclique

- Supposons que les données soient transmises par bloc de « k » bits
- A chaque bloc correspond un polynôme M(x) de degré « k »
- Exemple :

1.1.0.1.0.1.1.0.1.1 ➔ $x^9 + x^8 + x^6 + x^4 + x^3 + x + 1$

Codage à l'émission pour un bloc de « k » bits de données

- Multiplier M(x) par x^r où r est le degré de G(x)
- Calculer le reste R(x) de la division modulo 2 de $x^r.M(x)$ par G(x)
- Ajouter R(x) à M(x)
- Transmettre la séquence de « k + r » bits correspondant au polynôme
- $T(x) = x^r * (x) + R(x)$
- ***N .B :*** *par construction T(x) est divisible par G(x)*

Code détecteur : Code Polynomial ou Contrôle de la Redondance Cyclique

- Exemple:

- Soit à transmettre la trame : « P(x) = 1101011011 » ;
- Soit le polynôme diviseur G(x) = «x4+x+1» = « 10011 » ;
- Il faut donc diviser P(x) par G(x) comme suit :
- On ajoute au dividende, et à sa droite, autant de 0 que le diviseur comporte de bit moins un (ici 4=5-1) ce qui donne «11010110110000»
- Puis on exécute la division par soustractions successives grâce au OU-exclusif ;
- La dernière ligne de la division donnera le reste «1110» et donc le CRC qui sera ajouté au message que l'on va transmettre ;

Code détecteur : Code Polynomial ou Contrôle de la Redondance Cyclique

- Exemple:

G(x) = x^3+x+1 **d = 3**

SDU
1001 000 | 1011
1011
0010
10 00
10 11
00 **110**
reste

PDU ou message émis
1001 110

SDU
1011 0001 000 | 1011
1011
0000
1 000
1 011
0 **011**
reste

PDU ou message émis
1011 0001 011

Code détecteur : Code Polynomial ou Contrôle de la Redondance Cyclique

A la réception

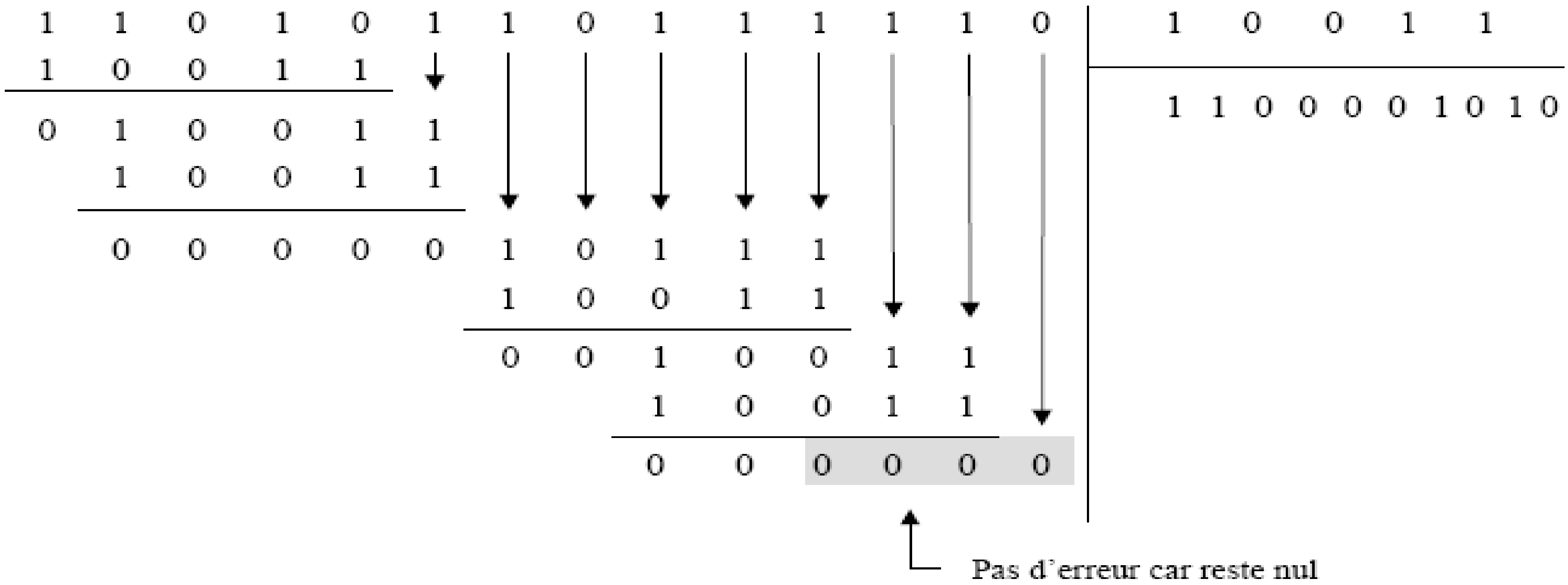
A la réception, on divise le polynôme T'(x) correspondant à la suite totale de bits reçus (information+CRC) par le polynôme générateur ;

- Si le reste calculé est non nul, c'est qu'une erreur s'est produite dans la transmission ;
- Si le reste est nul, on est à peu près sûr que la transmission s'est faite sans erreur ;

Code détecteur : Code Polynomial ou Contrôle de la Redondance Cyclique

- Exemple:

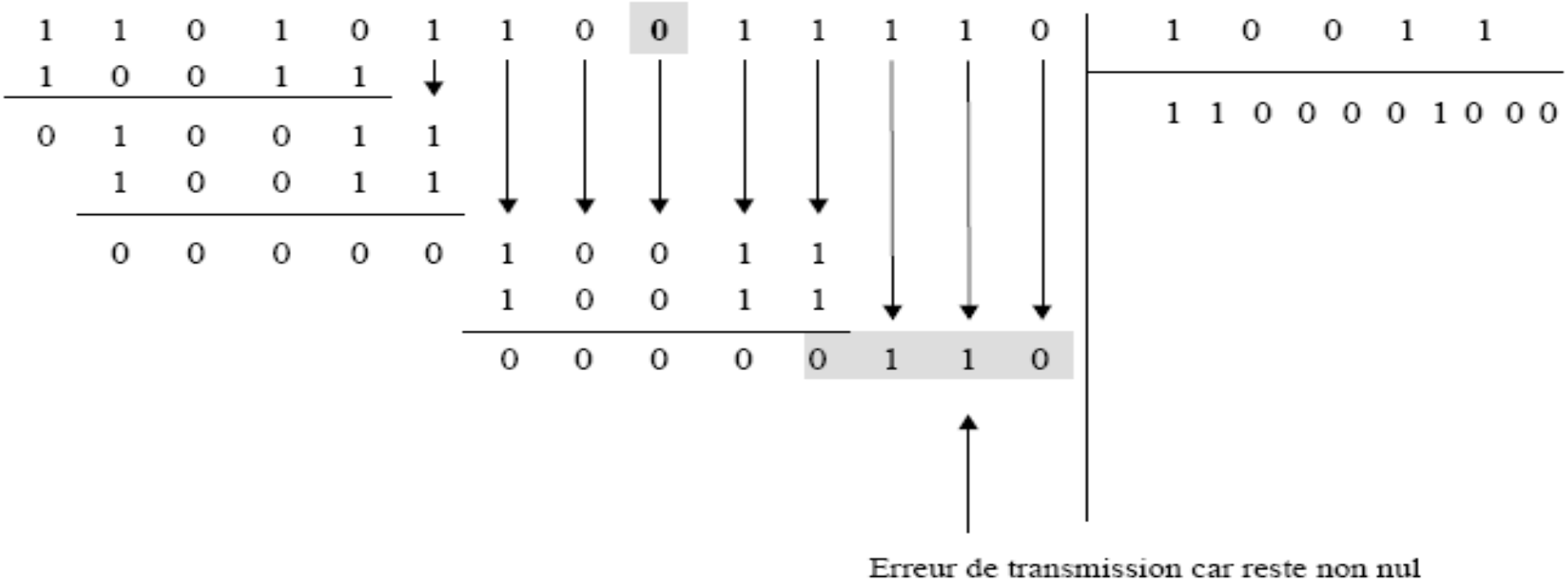
➢ **Sans erreur de transmission**



Code détecteur : Code Polynomial ou Contrôle de la Redondance Cyclique

- Exemple:

➢ **Avec erreur de transmission**



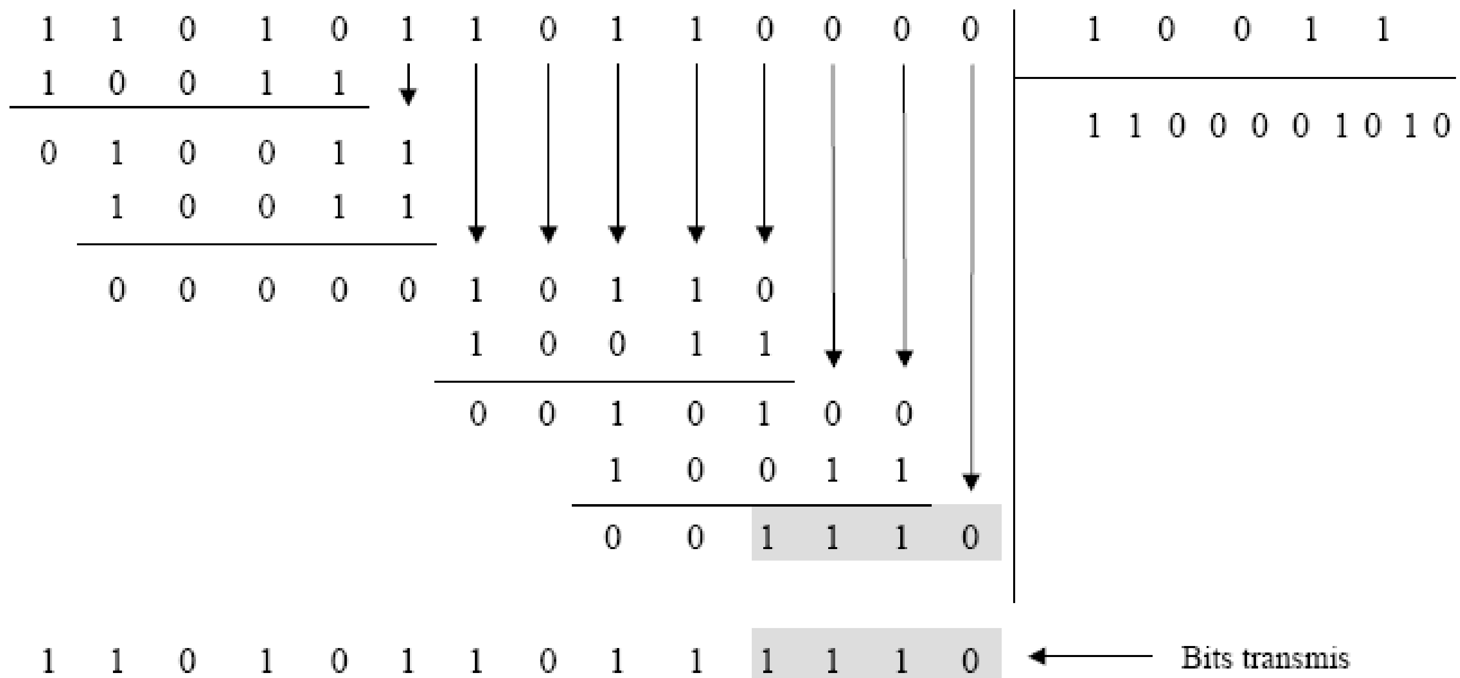
Code CRC : exemple

- Bits de données à envoyer : 1101011011
- Polynome générateur: $G(x)=x^4+x+1$

- Quelle trame est effectivement transmise?

Code CRC : exemple

- Bits de données à envoyer : 1101011011
- Polynome générateur: $G(x)=x^4+x+1$



Efficacité du Code CRC

- Les erreurs qui se traduisent par des polynômes facteur de G(x) ne sont pas détectées, toutes les autres le sont
- Si on utilise r bits de contrôle, le code CRC permet de détecter toutes les rafales d'erreurs de longueur inférieure ou égale à r

- Très utilisé en pratique : Ethernet, 802.11 WiFi, ATM
- Polynômes générateurs les + courants:
  - CRC-12: $x^{12}+x^{11}+x^3+x^2+x+1$
  - CRC-16: $x^{16}+x^{15}+x^2+1$
  - CRC-32 (Ethernet): $x^{32}+x^{26}+x^{23}+x^{22}+x+1$

Codes correcteurs d’erreur

- Introduire suffisamment de redondance pour que le récepteur soit capable de reconstruire l’information envoyée

- Utilisés sur les canaux sans fils où les erreurs sont courantes: => retransmettre le bloc n’assure pas que ce soit sans erreur