



COUCHE LIAISON DE DONNÉES

Chapitre 4

1

OBJECTIFS DU COURS

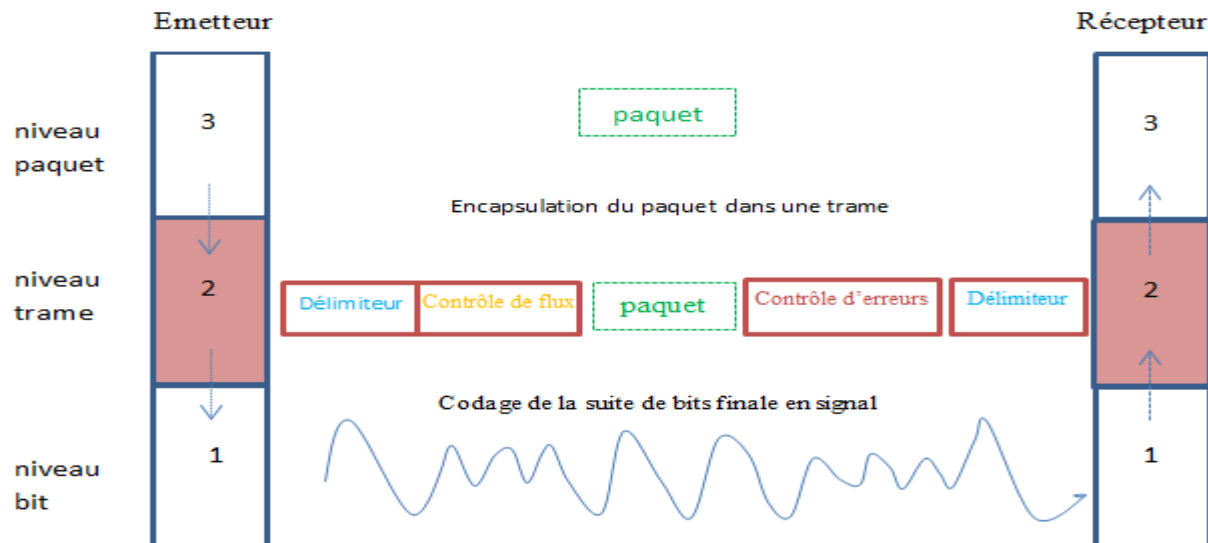
- Connaître le rôle et les fonctions de la couche Liaison de données
- Comprendre la notion de topologie logique
- Comprendre la délimitation des trames et connaître les différentes techniques de délimitation
- Connaître les techniques de détection et de correction d'erreurs
- Connaître les techniques de contrôle de flux
- Connaître les techniques d'accès au canal
- Distinguer les équipements de niveau 2 (Couche Liaison de données)

PLAN

- Introduction
- Rôle
- Fonctions
- Topologie logique
- Délimitations de trames
- Contrôle d'erreurs
- Contrôle de flux
- Accès au canal
- Équipements couche 2
- Conclusion

INTRODUCTION (1)

- La couche **Liaison de données** est la couche 2 des modèles OSI et hybride
- L'unité d'échange des données est la **trame**



INTRODUCTION (2)

- Dans le chapitre 2, nous avons vu que la Couche PHY s'occupe de la transmission des bits sur le support sous forme de signal
- Or ce signal peut s'altérer au point de ne pouvoir être décodé par le récepteur.
- Et des données reçues peuvent être rejetées par défaut de mémoire où les stocker

RÔLE


La couche Liaison de données

- Assure la **liaison logique** entre les équipements :
 - Topologie logique
- Gère les éventuelles **erreurs** subies par les données
- Éventuellement, gère l'accès au canal
 - dans les réseaux à support partagé

Pour assurer ce rôle, elle exécute un ensemble de fonctions implémentées dans des protocoles de liaisons de données...

FONCTIONS ET PROTOCOLES

Fonctions

- Formatage des trames
 - Acheminement des trames
 - Contrôle des erreurs
 - Contrôle des flux
- 
-
- Contrôle d'accès au support

Protocoles

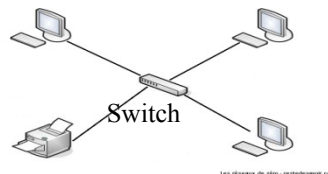
- **HDLC** (générique)
 - **LAP** (réseaux télécoms)
 - **PPP** (Internet)
 - **LLC** (réseaux locaux)
-
- **MAC** (protocole d'accès au support)

TOPOLOGIE LOGIQUE

Comment circulent les données (trames) ?

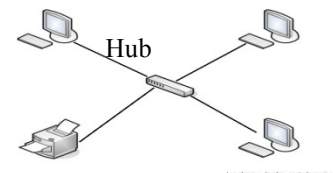
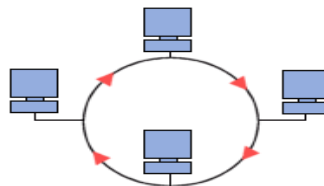
- Uniquement entre l'émetteur et le destinataire (ou l'intermédiaire)

- Topologie logique : **point-à-point**



- Dans tout le réseau

- Topologie logique: **diffusion**



Formatage des trames

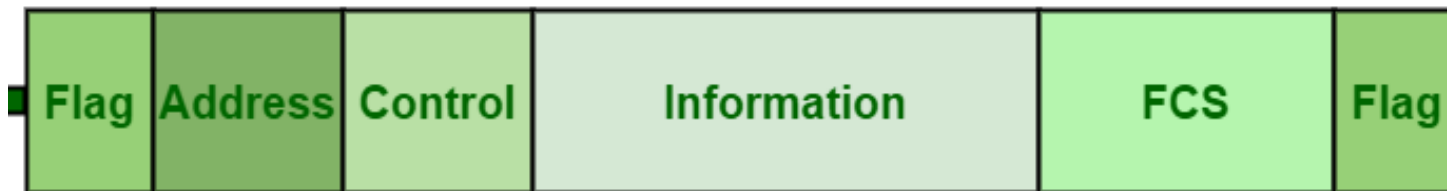
Chaque technologie réseau définit sa propre structure de trame



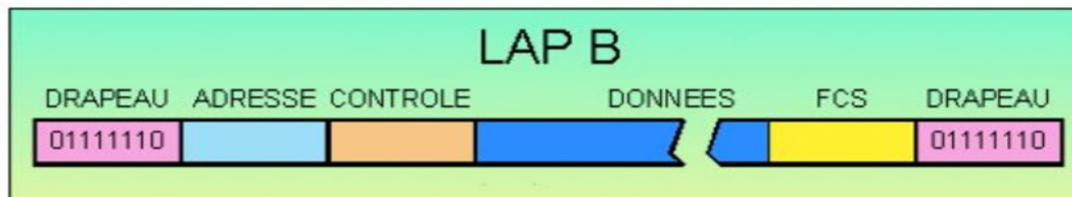
STRUCTURE DE BASE D'UNE TRAME

Une trame contient généralement les champs suivants:

- un indicateur de début et/ou de fin de trame(flag)
- les adresses
- le champ contrôle (ex : type de trame)
- les données
- le code de détection/correction d'erreurs



Exemple



DÉLIMITATION DE TRAMES

- Pour assurer la synchronisation entre l'émetteur et le récepteur

3 méthodes possibles de délimitation

- Comptage de caractères
- Fanions de début et fin
- Violation du codage

COMPTAGE DE CARACTÈRES

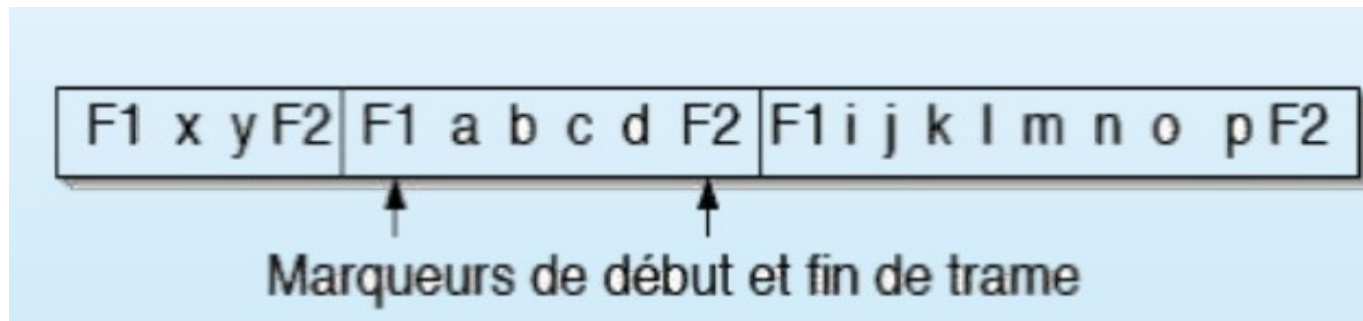
6	s	a	l	u	t	4	a	m	i
---	---	---	---	---	---	---	---	---	---

- **Principe** : mettre le nombre de caractères de la trame dans un champ (généralement le premier)
- **Inconvénients** : erreurs sur le champs contenant le nombre de caractères → **désynchronisation**

MARQUEURS DE DÉBUT ET/OU DE FIN

1/2

- Fanions ou Délimiteurs de trames



- **Problème:** Fanion présent dans le champs de données
- **Solution :** insérer des bits ou des caractères de transparence pour lever l'ambiguïté

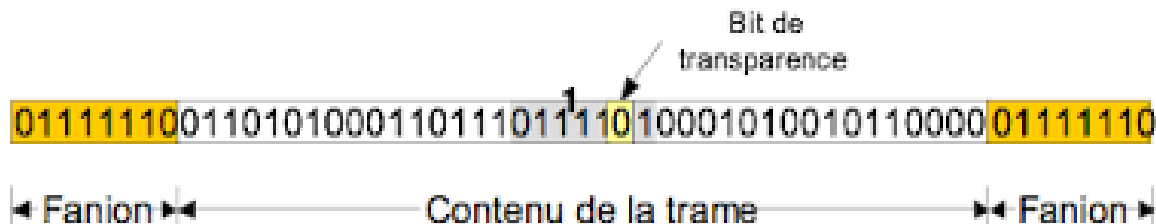
MARQUEURS DE DÉBUT ET/OU DE FIN

2/2

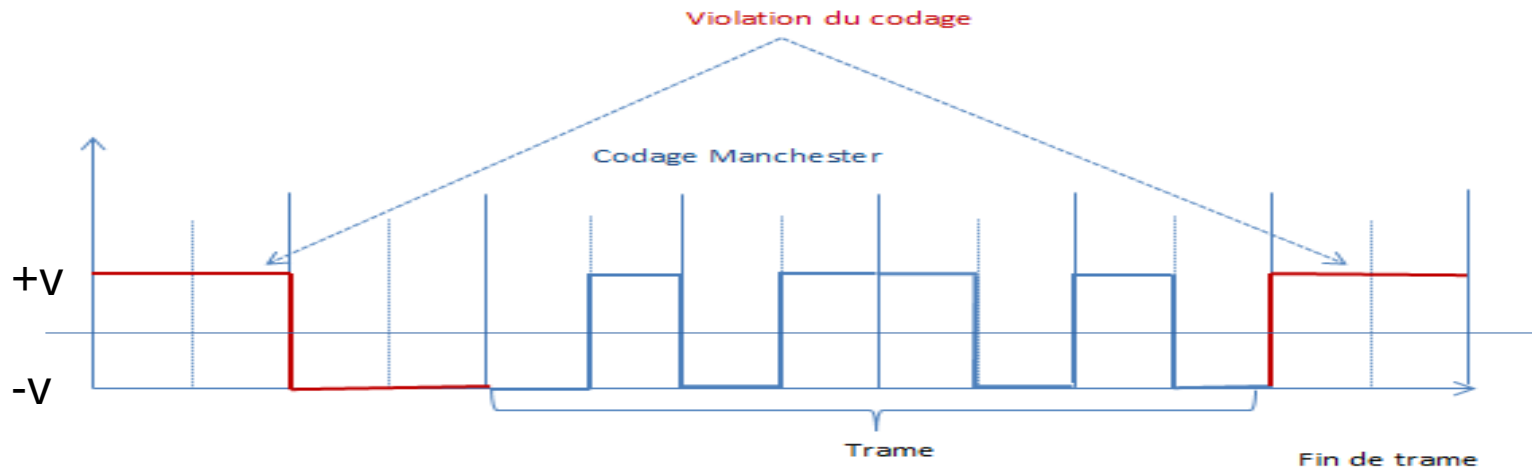
- Exemple : pour le fanion habituel **01111110**, pour lever l'ambiguïté, il faut ajouter un bit **0** après 5 bits à 1 successifs

01101010001101110**1111**100010100101100000

a) Données originales de la couche réseau



VIOLATION DU CODAGE



Utiliser un codage différent pour le délimiteur :
Exemple : **NRZ** pour le délimiteur de trame et
Manchester pour la trame

Acheminement des trames

Les trames sont diffusées ou envoyées au bon port selon le type d'équipement



ACHEMINEMENT DES TRAMES

Selon l'équipement de transfert utilisé, la trame peut être :

- **diffusée** sur tous les ports
 - Par le Hub
 - **Topologie logique en diffusion**
- **envoyée au bon port** selon l'adresse de destination
 - Par le Switch : un seul chemin vers une destination
 - Par le Routeur : plusieurs chemins possibles.
 - **Topologie logique en point-à-point**
 - **N.B** : dans la plupart des réseaux, les trames sont commutées : acheminées par un Switch

COMMUTATION DE CIRCUIT ET COMMUTATION DE PAQUET

○ Commutation de circuit

- Établissement d'un circuit avant l'envoi des données
- Circuit = un chemin vers la destination
- Bande passante réservée

○ Commutation de paquets

- Mode connecté :
 - Établissement d'un circuit **virtuel**
 - **Mais pas** de bande passante réservée
- Mode non connecté
 - Pas de chemin établi avant l'envoi des données
 - Plusieurs chemins possibles.
 - On parle de routage

Contrôle d'erreurs

Le contrôle d'erreurs permet de détecter d'éventuelles erreurs afin de supprimer, corriger ou demander la retransmission des trames erronées

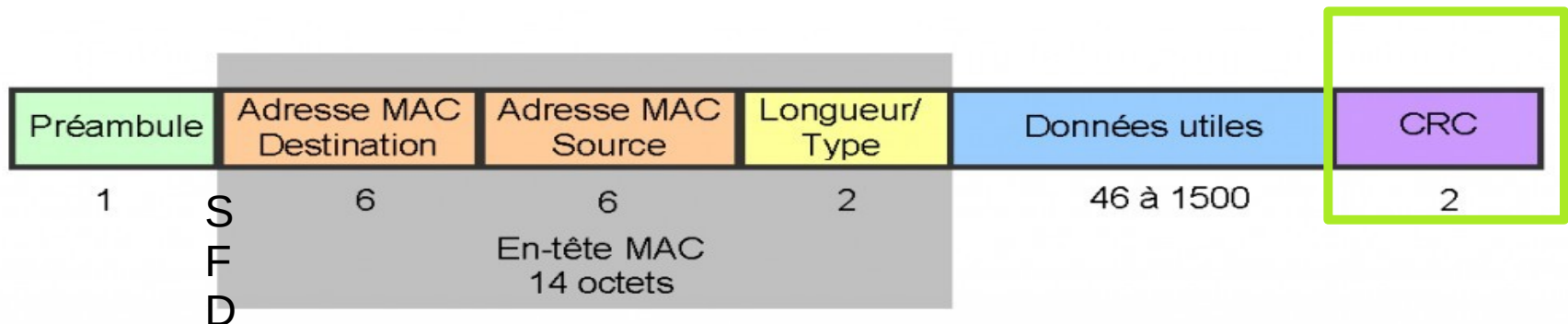


CONTRÔLE D'ERREURS 1/3

- Lors de la transmission sur un support, des bits peuvent subir des erreurs
- Causes possibles des erreurs :
 - Atténuation du signal sur une longue distance
 - Désynchronisation entre le récepteur et l'émetteur
 - bruit
- Certains **supports de transmission sont plus susceptibles d'erreurs** que d'autres:
 - Exemple : supports non guidés
- Il faut alors une fonction de contrôle d'erreurs pour une communication fiable

CONTRÔLE D'ERREURS 2/3

- Elle consiste en l'ajout **de bits de redondance** dans la trame avant son émission
 - Ces bits sont calculés selon un algorithme connu de l'émetteur et du récepteur
 - A la réception ces bits permettent de vérifier si la trame est bien transmise
 - Exemple : **CRC** (Code de redondance cyclique) d'une trame Ethernet



CONTRÔLE D'ERREURS 3/3

- Certains codes peuvent corriger des erreurs détectées : ce sont des **codes correcteurs d'erreurs**
- Les codes ont des degrés d'efficacité différentes et aucun code n'est parfait (des erreurs peuvent passer sans être détectées)
- 3 niveaux de contrôle d'erreurs:
 - **Détection et Correction** : corriger la trame erronée
 - Coûteux en taille mémoire et en calcul
 - **Détection et retransmission** : notifier à l'émetteur que la trame est erronée
 - Coûteux en délai
 - **Détection** (uniquement): ignorer la trame erronée
 - La notification sera faite par le destinataire de la donnée

CODE DÉTECTEUR : CODE DE PARITÉ (VRC)

◉ Vertical Redundancy Check

Principe

- ◉ A l'émission : ajouter un **bit de parité** à chaque mot de n bits.
 - 0 lorsque le nombre de bit 1 est pair et 1 sinon : parité paire
 - 1 lorsque le nombre de bit 1 est pair et 0 sinon : parité impaire
- ◉ A la réception : vérifier pour chaque mot que le nombre de 1 est pair (ou impair)

Exemple :

- ◉ on veut envoyer HI : 1001000 1001001 (code ASCII)
séquence à envoyer (en utilisant une parité paire) : **10010000** **10010011**

Inconvénients :

- ◉ Ne détecte pas les erreurs paires
- ◉ Ne peut pas corriger les erreurs

CODE DÉTECTEUR : CODE DE PARITÉ (LRC)

○ Longitudinal Redundancy Check

Principe

- A l'émission : ajouter un **mot** à chaque bloc de n mots de telle sorte que .
 - le nombre de bits 1 de chaque colonne soit paire : parité paire
 - le nombre de bits 1 de chaque colonne soit impaire : parité impaire
- A la réception : vérifier pour chaque colonne de mots que le nombre de 1 est pair (ou impair)

Exemple :

- on veut envoyer HI : 1001000 1001001 (code ASCII)
séquence à envoyer (en utilisant une parité paire) : **1001000 1001001 0000001**

Inconvénients :

- Ne peut pas corriger les erreurs

CODE DÉTECTEUR:CODE POLYNOMIAL

- ◉ CRC : Cyclic Redondancy Check ou
- ◉ FCS : Frame Check Sequence

Principe :

- ◉ A l'émission, on ajoute n bits de contrôle (CRC ou FCS) au message $M(x)$. Ces n bits sont le reste $R(x)$ de la division euclidienne du message $M(x)$ (auquel on ajoute n zéros) par un polynôme générateur $G(x)$ de degré n . Le message envoyé sera alors $M'(x) = M(x) + R(x)$
- ◉ A la réception le message $M'(x)$ est divisé par $G(x)$. Si le reste vaut 0 alors, il n'y a pas d'erreurs

Inconvénient

- ◉ Ne permet pas de localiser et de corriger les erreurs.

Code polynomial

Exemple

ÉMETTEUR		RÉCEPTEUR	
10111010010000	10011	10111010010110	10011
10011	-----	10011	-----
- 01000	1010010010	- 01000	1010010010
00000		00000	
- 10001		- 10001	
10011		10011	
- 00100		- 00100	
00000		00000	
- 01000		- 01000	
00000		00000	
- 10001		- 10001	
10011		10011	
- 00100		- 00100	
00000		00000	
- 01000		- 01001	
00000		00000	
- 10000		- 10011	
10011		10011	
- 00110		- 00000	
00000		00000	
- 0110		- 0000	
			→ pas d'erreur !
M(x) = 1011101001		2 ⁴ M(x) = 10111010010000	
C(x) = x ² + x + 1		R(x) = 0110	
2 ⁴ M(x) = 10111010010000		T(x) = 10111010010110	
D(x) = 10011			



CODE CORRECTEUR : PARITÉ CROISÉE (VRC+LRC)

Principe:

- Combiner les codes VRC et LRC
 - Ajouter un bit de parité à chaque mot
 - Ajouter un mot de parité à chaque séquence de mots

Exemple

- Pour envoyer HI

VRC

H 1001000 0

I 1001001 1

0000001 1 LRC

Séquence à envoyer 10010000 10010011 00000011

CODE CORRECTEUR : CODE DE HAMMING

Principe :

- On utilise un dictionnaire D de n mots de code de b bits, connu de l'émetteur et du récepteur
- **A l'émission**, on utilise un mot du dictionnaire pour envoyer une séquence de $\log_2(n)$ bits
- **A la réception**
 - on devrait trouver des mots du dictionnaire
 - Mais si on reçoit un mot qui n'est pas du dictionnaire, on cherche le mot du dictionnaire qui lui est le plus proche : mot dont la distance avec le mot reçu est la plus faible (distance entre 2 mots = nombre de bits qui diffèrent)
 - S'il en existe plusieurs, impossible de corriger

CODE DE HAMMING : EXEMPLE

- Soit le dictionnaire suivant $D = \{0000, 1011, 1100, 1111\}$
- Chaque mot du dictionnaire permet de coder une suite de 2 bits ($\log_2(4)$)
 - exemple : 0000 pour 00, 1011 pour 01, 1100 pour 10 et 1111 pour 11

Supposons que

- l'émetteur veuille envoyer la suite 00 10 01, la séquence à envoyer sera 0000 1100 1011

CODE DE HAMMING : EXEMPLE

- Le récepteur reçoit: 0000 0011 1000
 - 0000 : mot du dictionnaire OK
 - 0011 : ne fait pas partie de D
 $d(0011, 0000)=2$; $d(0011, 1011)=1$; $d(0011, 1100)=4$;
 $d(0011, 1111)=2$ – 0011 sera corrigée en 1011
 - 1000 : ne fait pas partie de D
 $d(1000, 0000)=1$; $d(1000, 1011)=2$; $d(1000, 1100)=1$;
 $d(1000, 1111)=3$ – Impossible de corriger

EFFICACITÉ D'UN CODE

Certains supports de transmission sont susceptibles d'erreurs

L'efficacité d'un code ou sa capacité à détecter ou corriger des erreurs est relative au nombre de bits de redondante. Plus ce nombre est grand, plus le code détecte ou corrige mieux les erreurs mais plus le rendement (longueur du mot d'information/longueur du message émis) de la transmission est faible. Il est alors nécessaire de faire un compromis entre l'efficacité de la transmission et son rendement.

Contrôle de flux

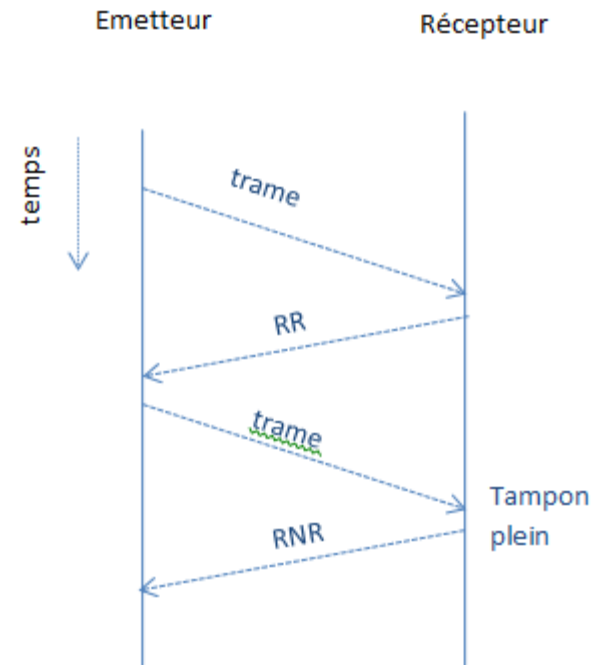


CONTRÔLE DE FLUX

- Il peut arriver qu'un émetteur soit trop rapide par rapport au récepteur
- Il faut alors un contrôle de flux pour éviter des pertes de trames (par faute de mémoire où les stocker)
- Il existe deux types de méthodes :
 - **Send and Wait** :
 - envoyer une trame et attendre sa notification
 - Pour transmission de données dans un sens
 - **Sliding Windows** :
 - Envoyer plusieurs trames en attendant leurs notifications
 - Pour transmission de données dans les deux sens

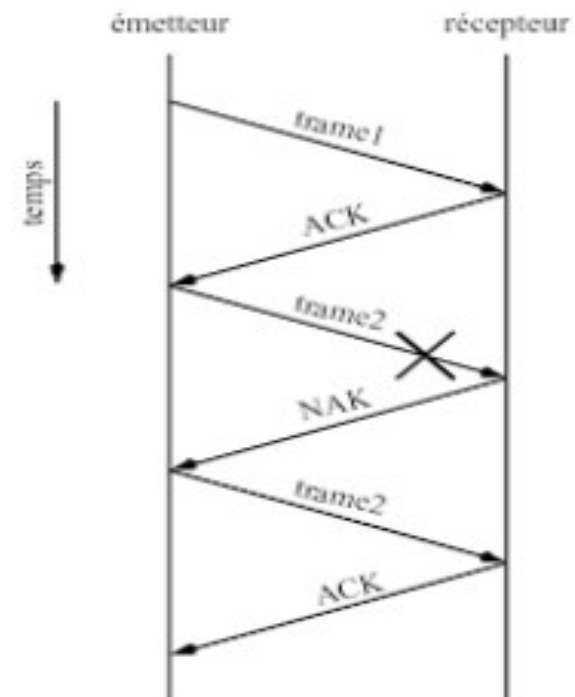
SEND AND WAIT SUR CANAL PARFAIT

- Canal fiable : pas d'erreurs de transmission
- Possibilité de saturation de mémoire
- **Solution** : envoi de trame de supervision
 - **RR** : Receiver Ready
 - **RNR** : Receiver Not Ready



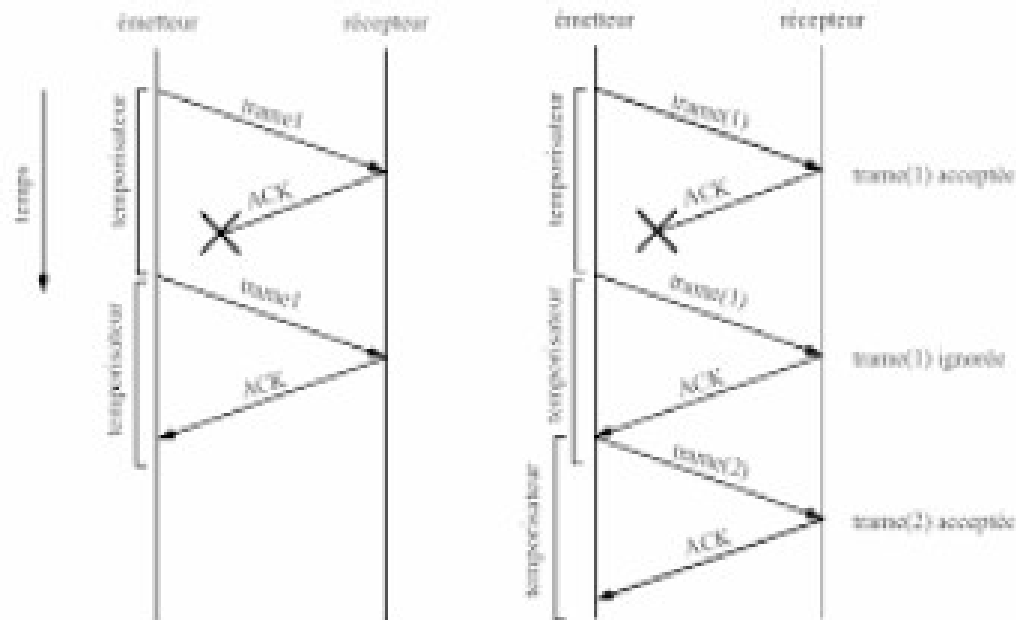
SEND AND WAIT SUR CANAL BRUITÉ

- Possibilité d'erreurs de transmission
 - **Détection d'erreurs**
- **Solution** : envoi d'acquittements
 - **ACK** : positif
 - **NACK** : négatif (en cas de détection d'erreurs)
- A la réception d'un NACK, l'émetteur **retransmet** la trame



SEND AND WAIT SUR CANAL BRUITÉ

- ACK perdu ?
- Solution :
 - l'émetteur maintient un *timeout* pour recevoir un acquittement
 - à son expiration, il retransmet la trame



SLIDING WINDOW (1/3)

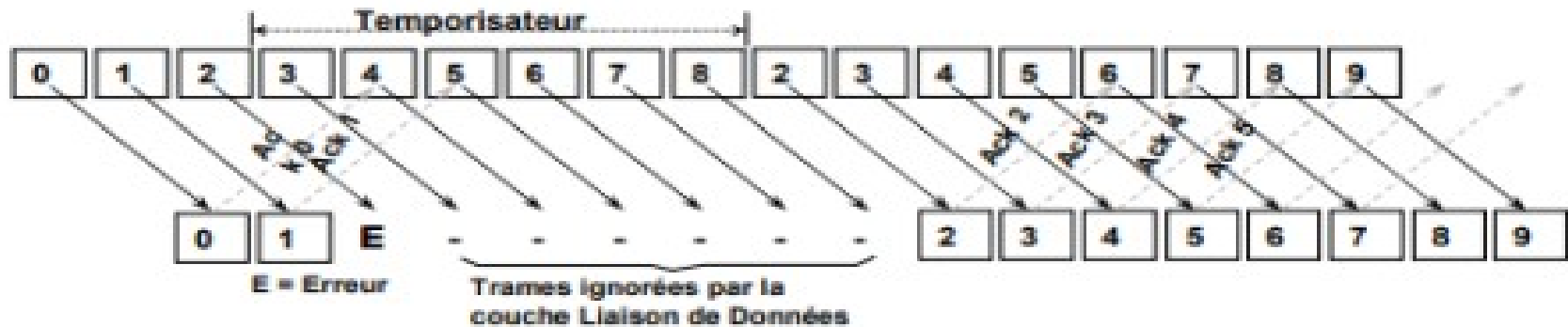
- Éviter les longues attentes du «*Send and Wait* »
- **Principe:**
 - Envoyer **n** trames avant de commencer à recevoir les acquittements
 - **n** est la taille de la fenêtre d'anticipation

SLIDING WINDOW (2/3)

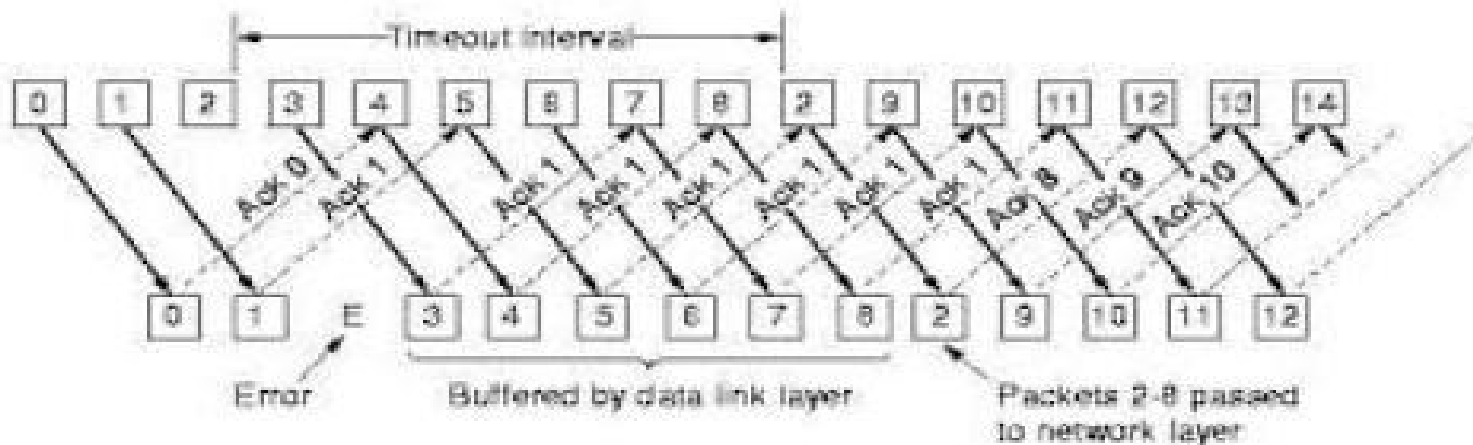
Lorsqu'une trame est erronée, on a 2 possibilités :

- rejeter toutes les trames qui la suivent : **rejet global**
- supprimer la trame erronée et placer en mémoire tampon les autres trames : **rejet sélectif**

SLIDING WINDOW (3/3)



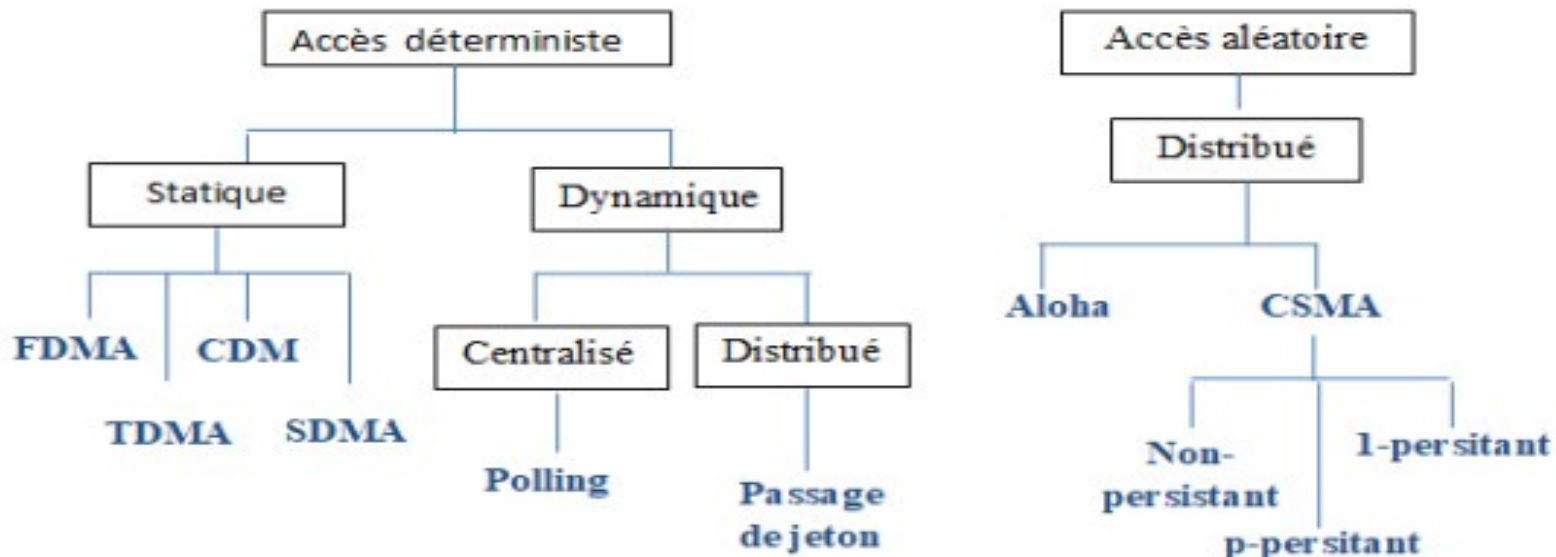
Rejet global



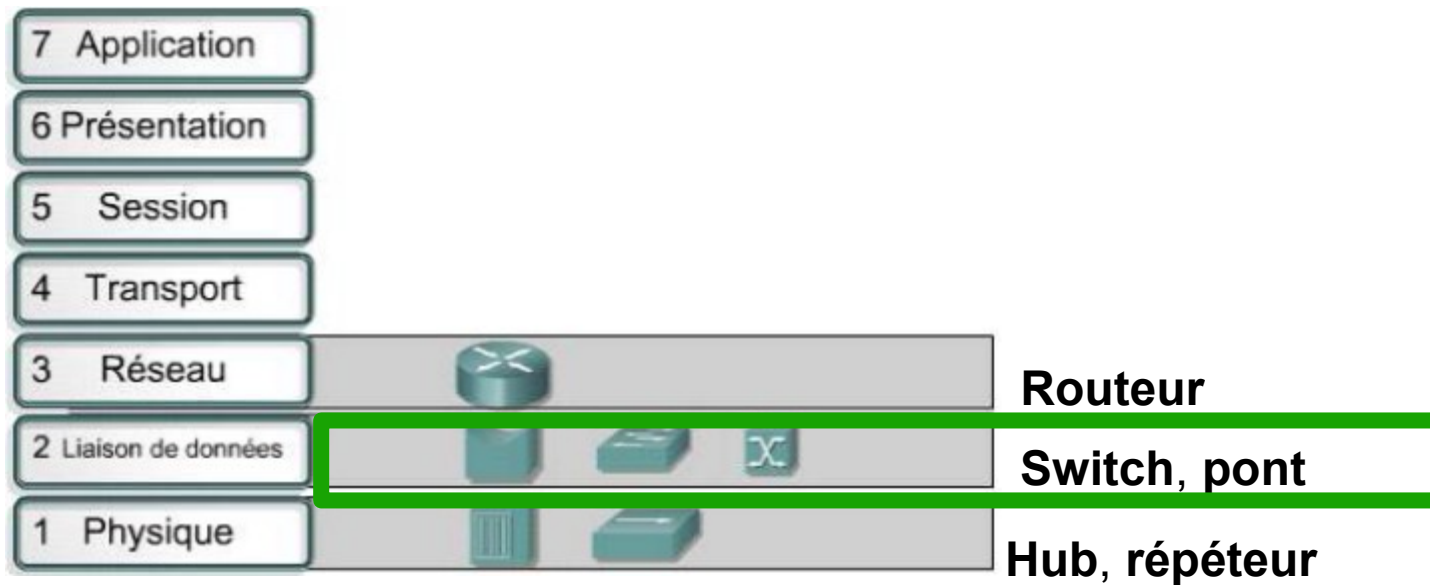
Rejet sélectif

ACCÈS AU CANAL

- Dans les réseaux à support partagé comme les réseaux locaux ou les réseaux sans fil, il est nécessaire de mettre en œuvre un mécanisme d'accès au canal pour éviter des collisions



ÉQUIPEMENTS NIVEAU LIAISON



CONCLUSION

- La couche Liaison de données assure la fiabilisation de la communication entre équipements adjacents
- Le chapitre suivant présente la description des couches PHY et Liaison d'un type (technologie) de réseau (Ethernet)