

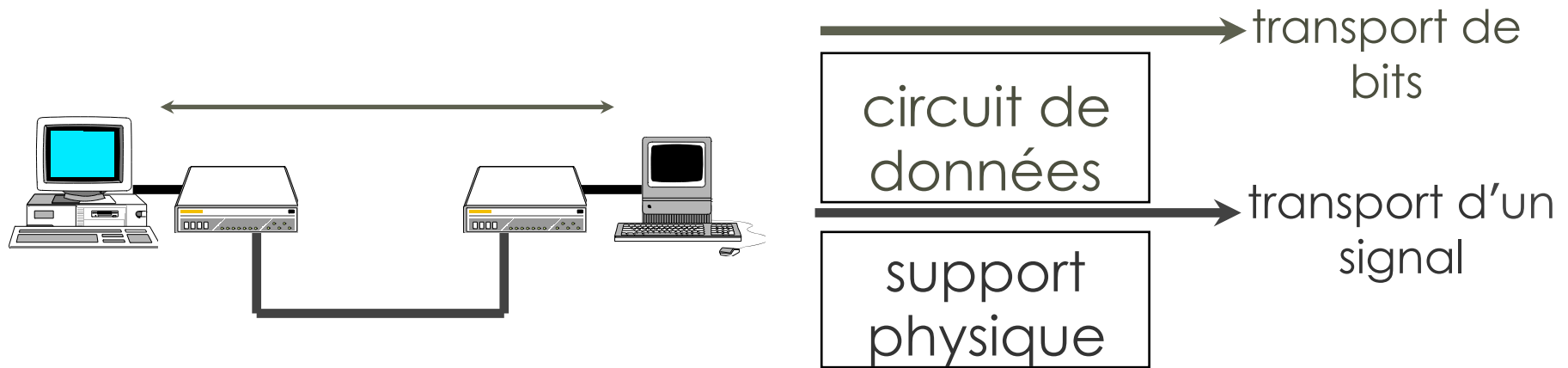
Partie 3

Liaison de données

Liaison de données - Plan

- Rôle de la liaison de données
- Mécanismes de communication
- Un exemple de protocole : HDLC

Etat de notre architecture



Pourquoi une liaison de données ?

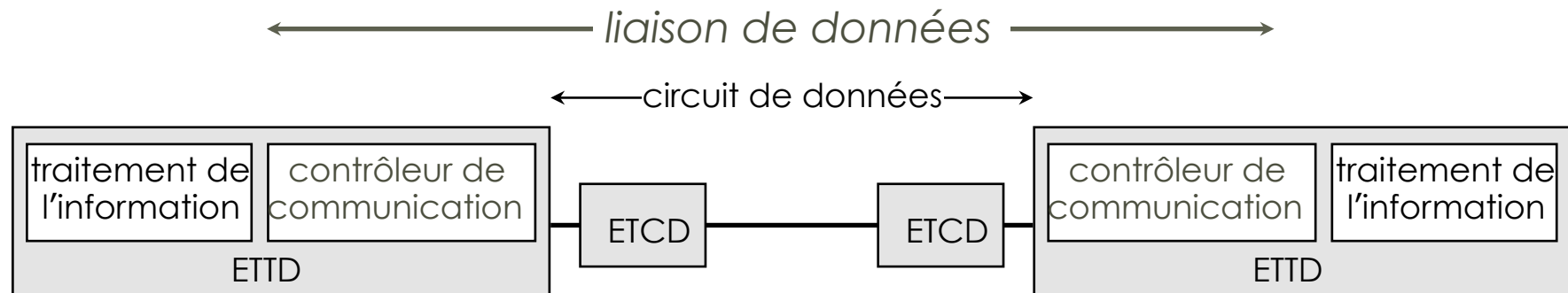
- Le circuit de données permet d'émettre et/ou de recevoir des bits en série

mais...

avec :

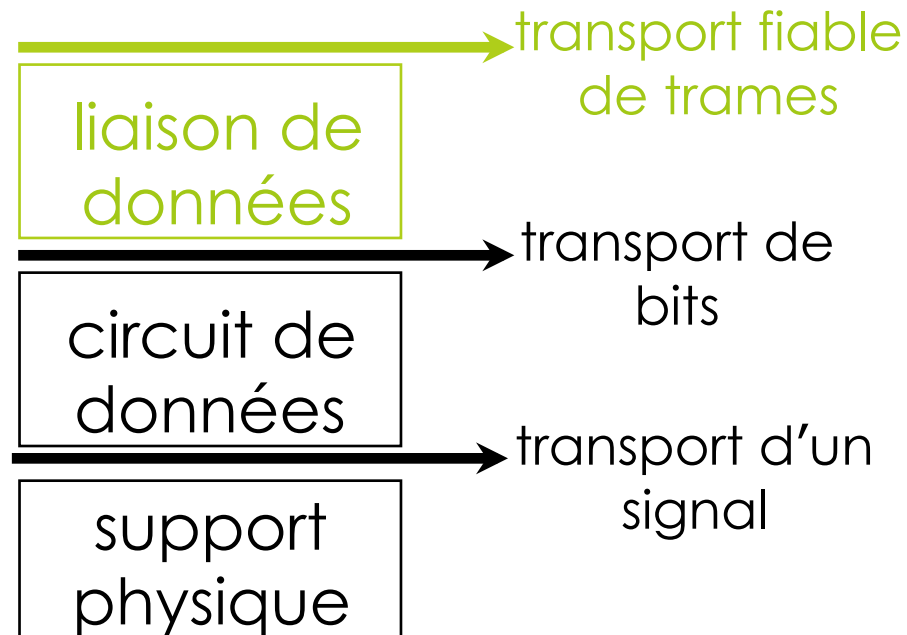
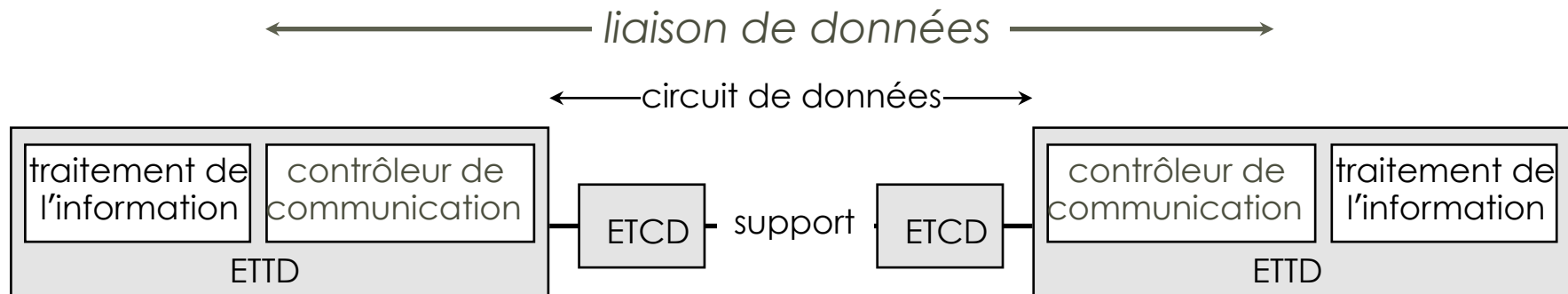
- un certain débit
 - un certain délai
 - un certain taux d'erreurs
- C'est insuffisant !
 - ajout d'une interface logique

Qu'est-ce qu'une liaison de données ?



- Rôle : fiabiliser le transfert d'information entre 2 ETTD
- Fiabilité :
 - pas d'erreur
 - pas de perte
 - pas de déséquencelement
 - pas de duplication

Liaison de données



Services de liaison de données

- Les services à utiliser
 - services offerts par le circuit de données
- Les services à rendre
 - les utilisateurs doivent avoir l'illusion d'un canal de transmission fiable
 - les utilisateurs doivent pouvoir établir et libérer logiquement une liaison au dessus d'un circuit
- 3 éléments de service
 - établissement de liaison
 - transfert de données
 - libération de liaison

Protocole de liaison de données

- une liaison de données : un canal physique capable de transmettre des bits, raccordant 2 (ou N) stations et leur permettant d'échanger de l'information structurée en **frames**
- un **protocole** de liaison de données : un ensemble de **règles** permettant de gérer la liaison
 - règles de **codage**
 - règles de **structuration**
 - règles **d'échange**
- le protocole met en œuvre un certain nombre de **mécanismes de communication**

Liaison de données - Plan

- Rôle de la liaison de données
- Mécanismes de communication
- Un exemple de protocole : HDLC

Les problèmes possibles

- erreurs de transmission
- rupture du circuit de données
- pertes de trames
- débordement du récepteur
- panne d'un des ETTD
- panne d'un des ETCD
- etc.

Délimitation des trames

■ Problème

Le récepteur doit savoir quand commence une trame et quand elle finit

■ Idée

Une trame doit commencer par un marqueur de début de trame et se finir par un marqueur de fin de trame

➤ **fanion de début/fin de trame**

Exemple de fanion : 01111110

- mécanisme de transparence

pour s'assurer qu'à l'intérieur de la trame, la configuration 01111110 ne soit pas interprétée comme un flag de fin

- à l'émission : ajout d'un bit "0" après 5 "1" consécutifs
- à la réception : destruction d'un bit "0" après 5 "1" consécutifs

- émission d'une trame

- construction de la trame
- calcul des bits de contrôle
- transparence (ajout)

- réception d'une trame

- transparence (retrait)
- calcul des bits de contrôle
- examen des champs de la trame

Exemple de fanion : 01111110

émission de

...1 1 0 1 1 0 1 1 1 1 1 0 1 0 1...

insertion de "0"

...1 1 0 1 1 0 1 1 1 1 1 **0** 0 1 0 1...



destruction de "0"

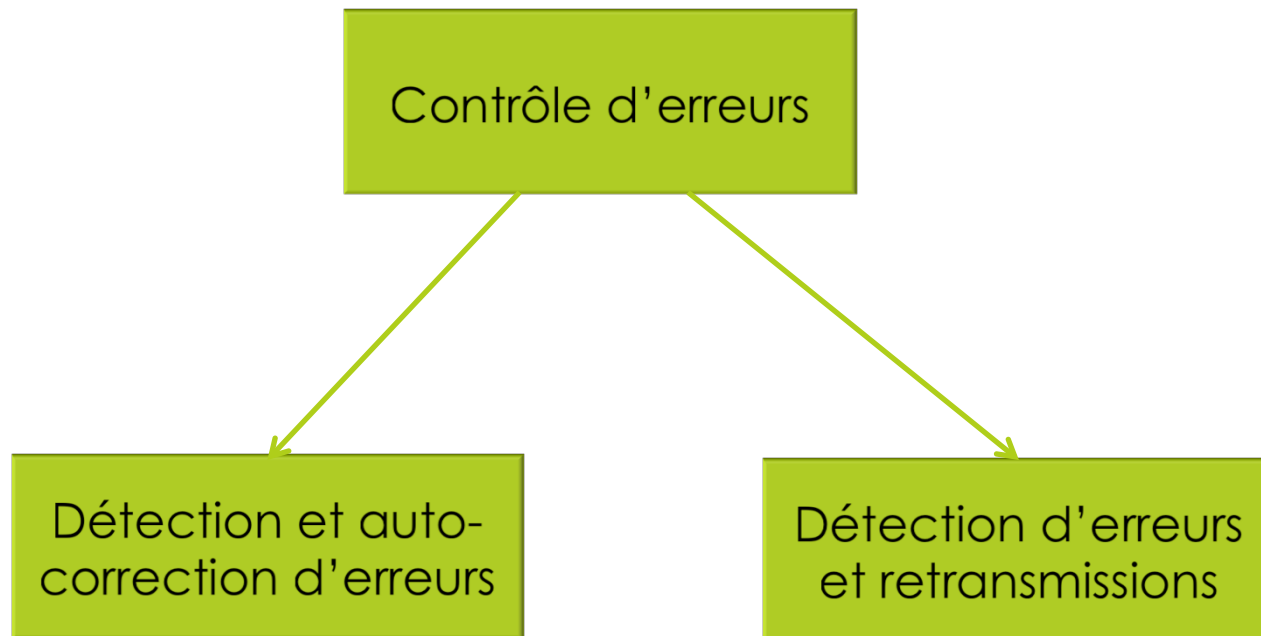
...1 1 0 1 1 0 1 1 1 1 1 ~~0~~ 0 1 0 1...

réception de

...1 1 0 1 1 0 1 1 1 1 1 0 1 0 1...

Contrôle d'erreurs

■ Deux stratégies



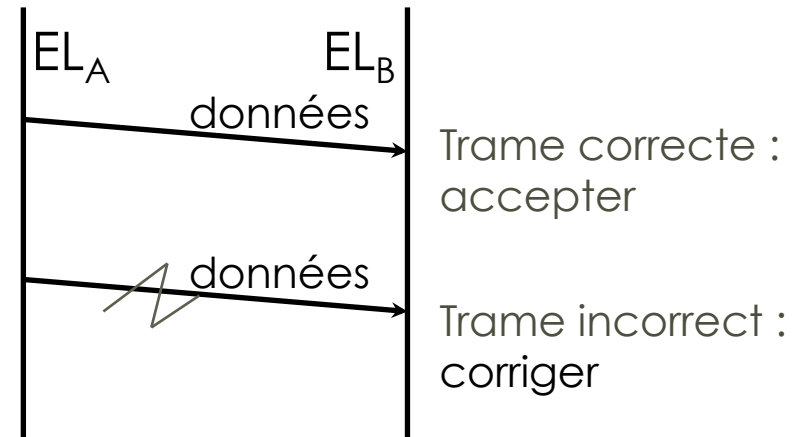
Auto-correction d'erreurs

■ Problème

Des erreurs de transmission peuvent altérer les données

■ Idée

Introduire suffisamment de redondance permettant au récepteur de restituer les données originales à partir de la trame reçue



➤ **Codes correcteurs d'erreurs (e.g. code de Hamming)**

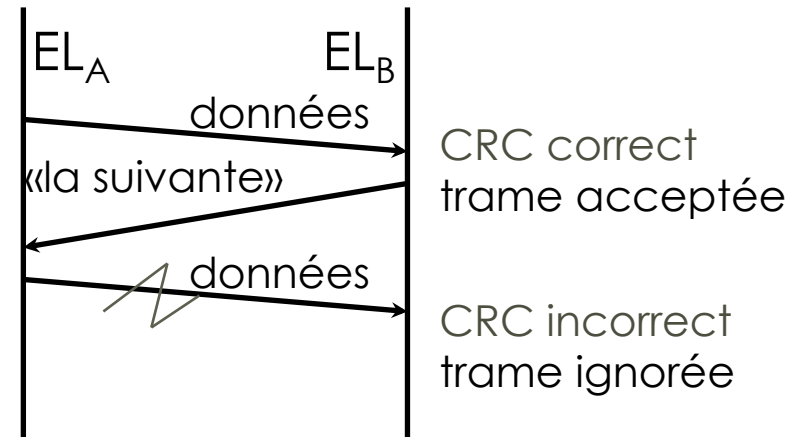
Détection d'erreurs

■ Problème

Des erreurs de transmission peuvent altérer les données

■ Idée

Introduire de l'information de contrôle permettant de détecter la présence d'erreurs de transmission dans une trame et éventuellement demander une retransmission



➤ **champ de contrôle de type CRC**

Capacité d'un code

- Un **mot de code** est l'ensemble de n bits d'une trame de m bits de données et r bits de redondance
- La **distance de Hamming d** entre 2 mots de code est le nombre de bits qui diffèrent entre eux
- La distance de Hamming D d'un code est le minimum de toutes les distances entre tous les mots du code pris deux à deux
- La capacité de détection d'un code est égale à $D - 1$
- La capacité de correction d'un code est égale à $(D - 1)/2$



Ex : Distance de Hamming entre ces deux mots de code est égale à 3

Bit de parité

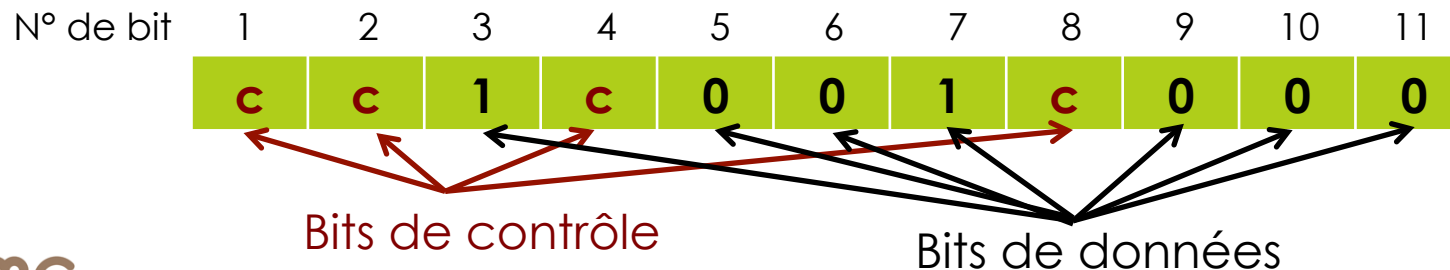
- Code de détection d'erreurs
- Détecter des erreurs simples (distance de Hamming égale à 2)
- Un bit de parité est rajouté aux bits de données
- Deux façons de faire
 - Parité paire
 - Le bit de parité est rajouté pour que le nombre de bits 1 dans le mot de code soit *pair*
 - Ex: 100110 → 100110**1**
 - Parité impaire
 - Le bit de parité est rajouté pour que le nombre de bits 1 dans le mot de code soit *impair*
 - Ex: 100110 → 100110**0**

Code de Hamming (1)

- Code correcteur d'erreurs
- Corriger les erreurs simples (distance de Hamming égale à 3)
- Les position des bits de données et des bits de contrôle dans un mot de code sont les suivantes:
 - Les bits sont numérotés de gauche à droite
 - Les bits dont les numéros sont des puissances de 2 (1, 2, 4, 8, 16, etc.) sont les bits de contrôle
 - Les autres bits (3, 5, 6, 7, 9, etc.) sont des bits de données

Exemple:

Le code ASCII du caractère « H » (7 bits - 1001000) dans le code de Hamming avec 4 bits de contrôle



Code de Hamming (2)

- Déterminer les bits de contrôle pour chaque bit de données
 - Le bit de données numéro k est contrôlé par les bits de contrôle dont les numéros sont les nombres qui apparaissent dans sa composition en puissance de 2

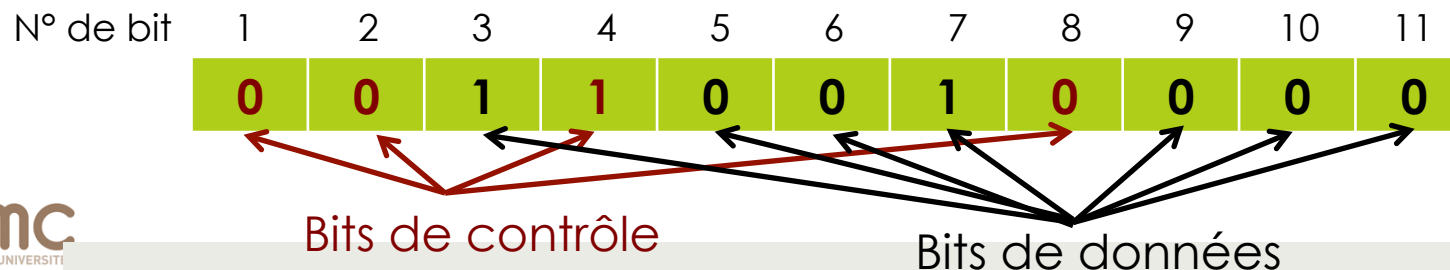
■ Exemple

N° de bit de données	Décomposition en 2^k	N° des bits de contrôle concernés
3	$2 + 1$	1, 2
5	$4 + 1$	1, 4
6	$4 + 2$	2, 4
7	$4 + 2 + 1$	1, 2, 4
9	$8 + 1$	1, 8
10	$8 + 2$	2, 8
11	$8 + 2 + 1$	1, 2, 8

Code de Hamming (3)

- Calculer les bits de contrôle
 - Chaque bit de contrôle est choisi de façon que la série des bits de données qu'il contrôle et lui-même ait une parité paire (ou impaire)
- Exemple (pour une parité paire)

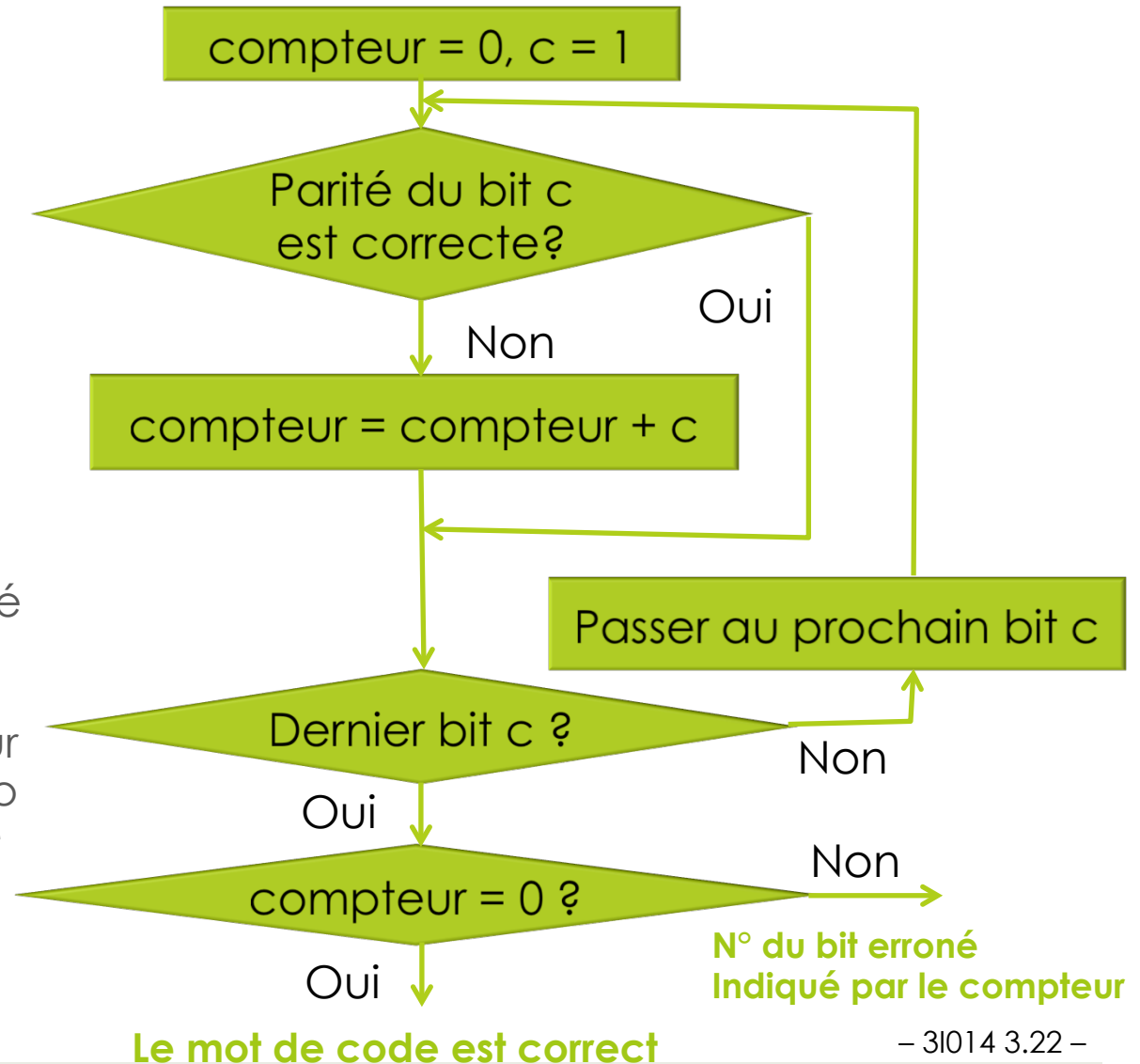
N° de bit de contrôle	N° des bits de données contrôlées	Valeurs des bits de données	Valeur du bit de contrôle
1	3, 5, 7, 9, 11	1, 0, 1, 0, 0	0
2	3, 6, 7, 10, 11	1, 0, 1, 0, 0	0
4	5, 6, 7	0, 0, 1	1
8	9, 10, 11	0, 0, 0	0



Code de Hamming (4)

■ Au récepteur:

- Un compteur est initialisé à 0.
- La parité de chaque bit de contrôle c ($c = 1, 2, 4, 8, \text{etc.}$) est vérifiée.
- Si la parité du bit de contrôle c n'est pas correcte, c est rajouté au compteur.
- A la fin, si le compteur est non nul, le numéro du compteur indique le numéro du bit erroné.



Cyclic Redundancy Control (1)

■ Principe

- hyp : E et R utilisent le même polynôme générateur $G(x)$ de degré r
- en émission
 - $M(x)$ la forme polynomiale de la trame à émettre, de degré $k-1$
 - division mod2 : $x^r.M(x) = G(x).Q(x) + R(x)$
 - $T(x) = x^r.M(x) - R(x) = x^r.M(x) + R(x)$, $T(x)$ divisible par $G(x)$
 - k premiers bits : bits d'information
 - r derniers bits : bits de contrôle
- en réception
 - R reçoit une trame $T'(x)$
 - division mod2 de $T'(x)$ par $G(x)$
 - si reste $\neq 0$ alors erreur(s) de transmission
sinon il y a une très forte probabilité pour que la trame soit exempte d'erreur (possibilité d'erreurs résiduelles)

Cyclic Redundancy Control (2)

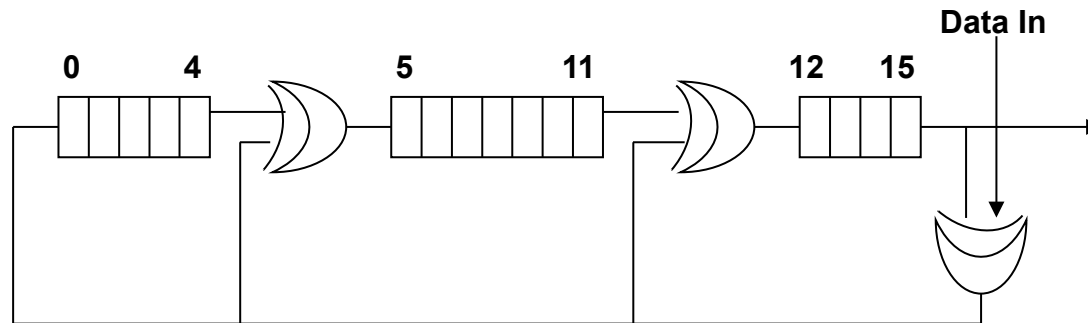
■ Exemples de polynômes générateurs

■ V.41 : $x^{16} + x^{12} + x^5 + 1$

■ CRC-16 : $x^{16} + x^{15} + x^2 + 1$

■ CRC-12 : $x^{12} + x^{11} + x^3 + x^2 + x^1 + 1$

■ Mise en œuvre



➤ le champ de CRC se situe généralement en fin de trame

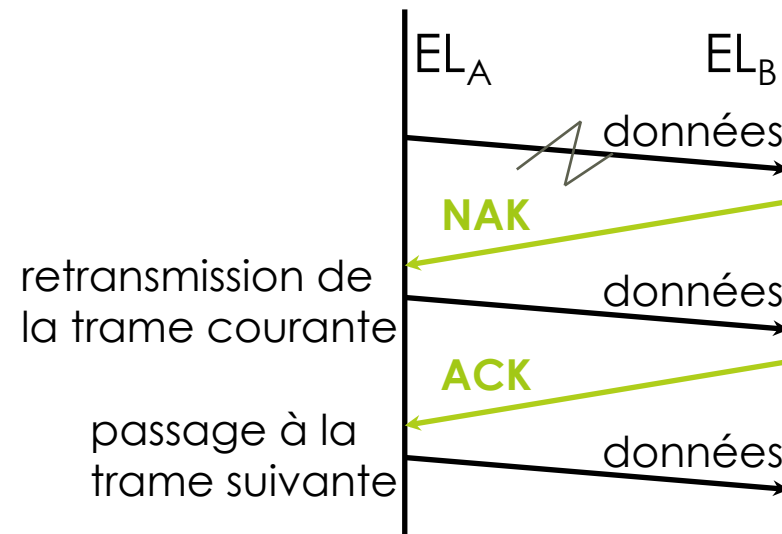
Reprise sur erreur

■ Problème

Récupérer une trame de données précédemment en erreur

■ Idée

Introduire une trame de contrôle demandant la retransmission de la trame de données



➤ **trame de contrôle NAK**

➤ **trame de contrôle ACK**
(« la suivante »)

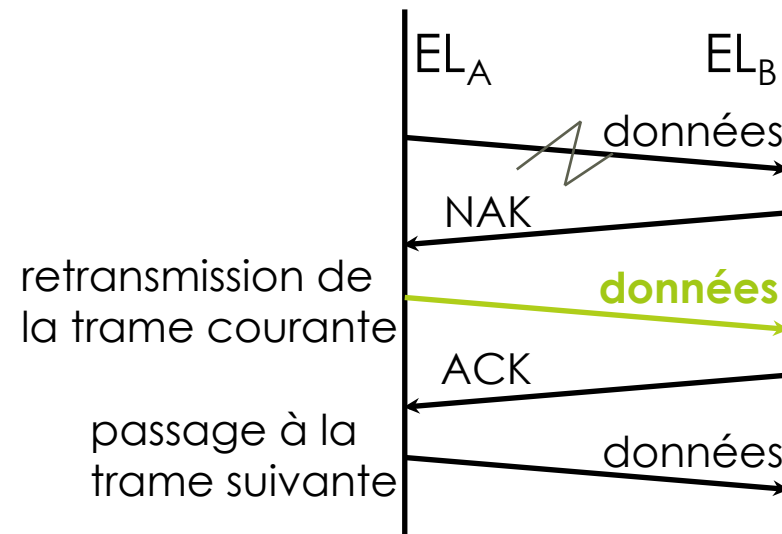
Rétention d'une copie

■ Problème

Pouvoir réémettre une trame de données précédemment envoyée

■ Idée

Garder une copie de toute trame de données émise jusqu'à réception d'un acquittement positif

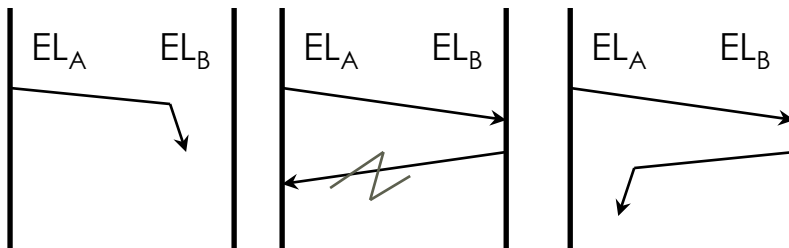


➤ **buffer d'émission**

Temporisateur de retransmission

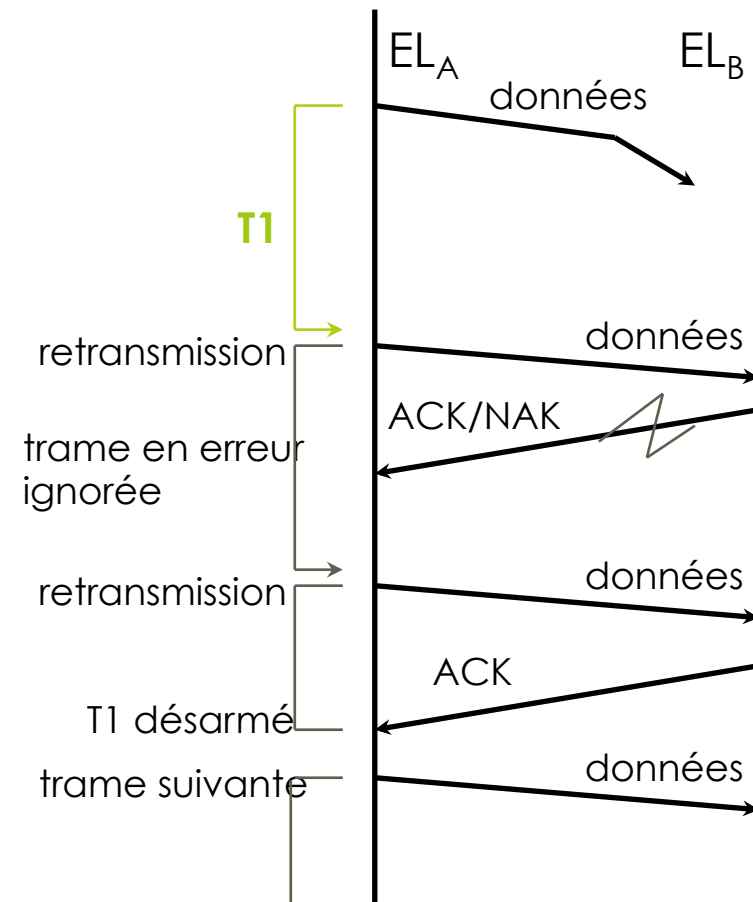
■ Problème

Pertes possibles de trames provoquant des inter-blocages



■ Idée

Utiliser un mécanisme de temporisation limitant la durée d'attente d'une réponse



➤ **temporisateur de retransmission T1**

Temporisateur de retransmission

- Dimensionnement de T1
 - T1 trop petit : retransmissions inutiles
 - T1 trop grand : reprise tardive
 - $T1 = f(?)$

- Les trames NAK ne sont plus obligatoires
 - mais elles permettent d'accélérer la reprise

Nombre maximum de tentatives

■ Problème

En cas d'incident grave du circuit de données,
retransmissions d'une même trame à l'infini

■ Idée

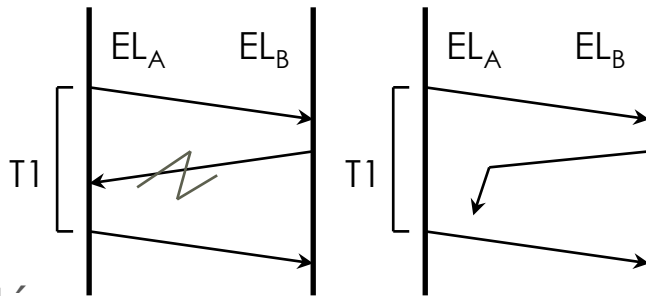
- limiter le nombre maximum de transmissions successives d'une même trame
- libérer éventuellement la liaison (et les ressources de communication associées)

➤ **Nombre maximum $N1$** d'émissions d'une même trame

Numérotation des trames de données

■ Problème

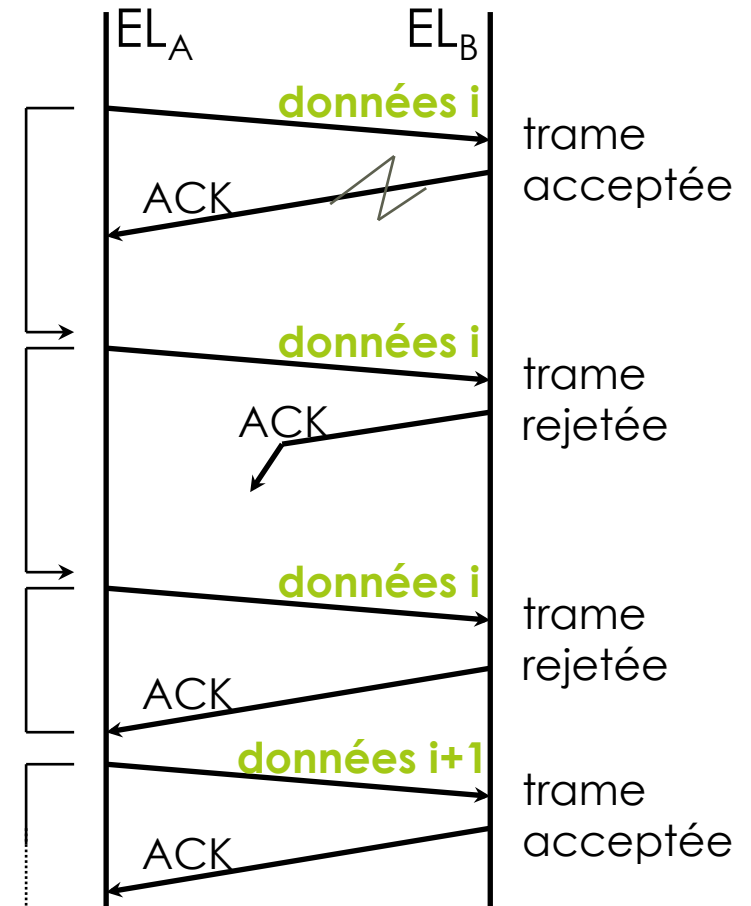
Duplications de données possibles



■ Idée

Utiliser dans la trame de données un champ de numérotation en séquence

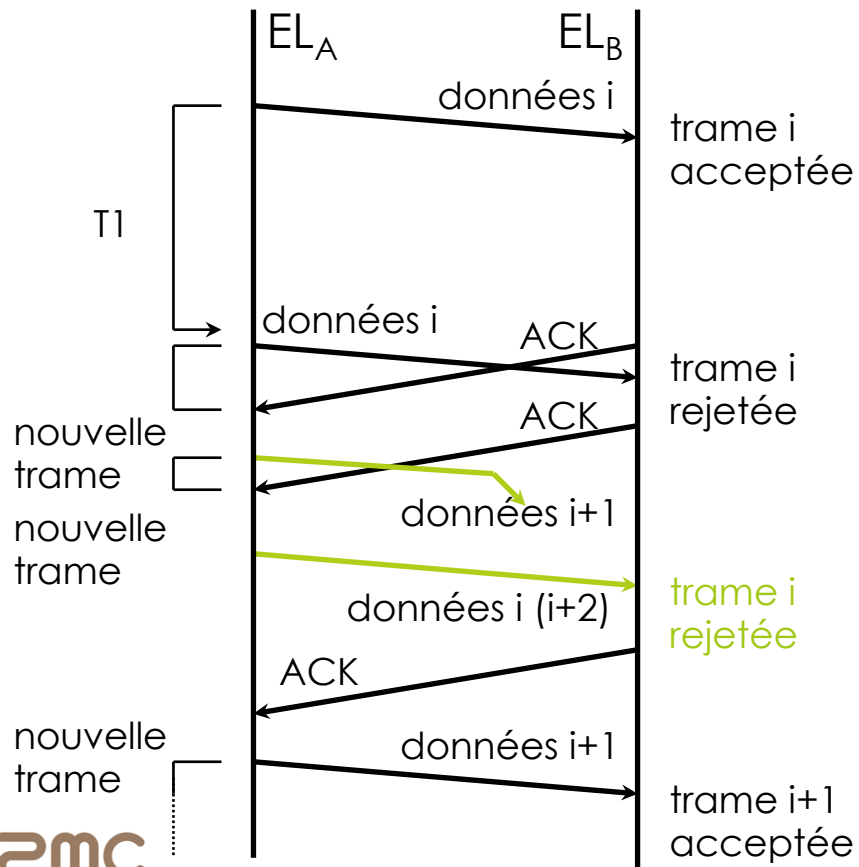
- **champ $N(S)$** pour les trames de données
 - numérotation **modulo m**
 - variable **$V(S)$** en émission



Numérotation des ACK

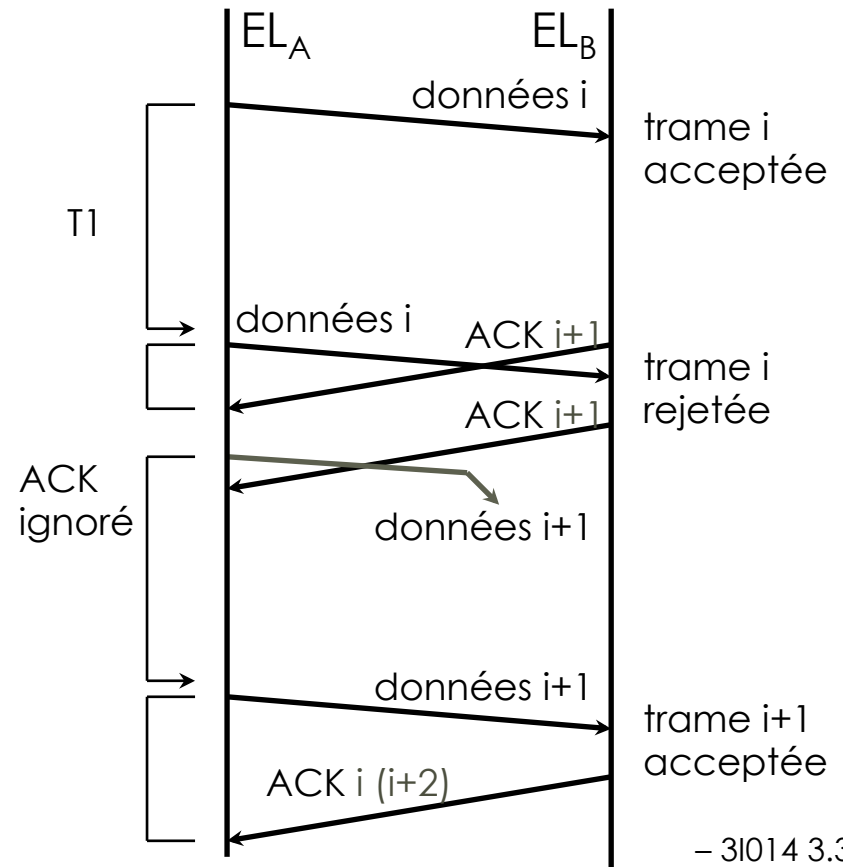
■ Problème

Pertes de données non récupérables possibles



■ Idée

Utiliser dans la trame ACK un champ de numérotation



Numérotation des ACK

- **champ $N(R)$** pour les trames ACK
 - numérotation **modulo m**
 - variable **$V(R)$** en réception

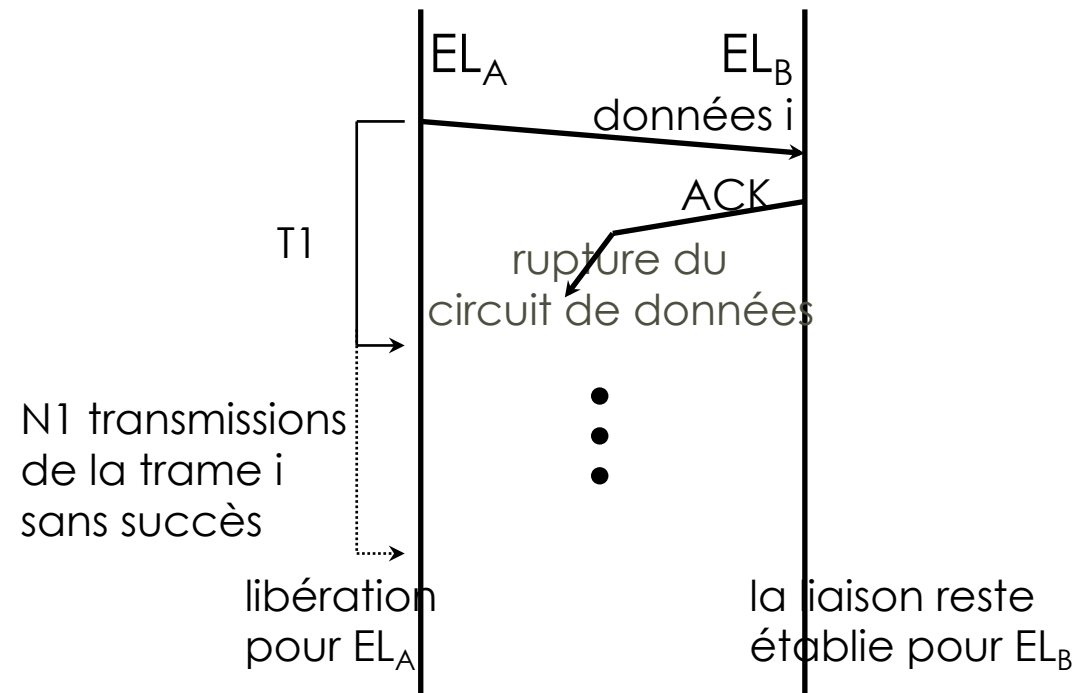
Détection d'inactivité

■ Problème

Ressources de communication bloquées inutilement

■ Idée

Utiliser un mécanisme de temporisation limitant la durée d'inactivité de la liaison



➤ temporisateur de détection d'inactivité I

➤ dimensionnement

➤ envoi de trames pour maintenir une activité

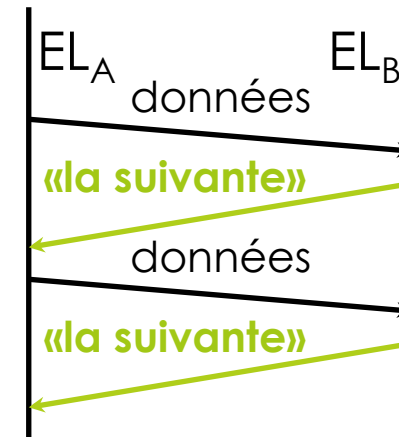
Contrôle de flux

■ Problème

Pertes de données possibles si engorgement du récepteur

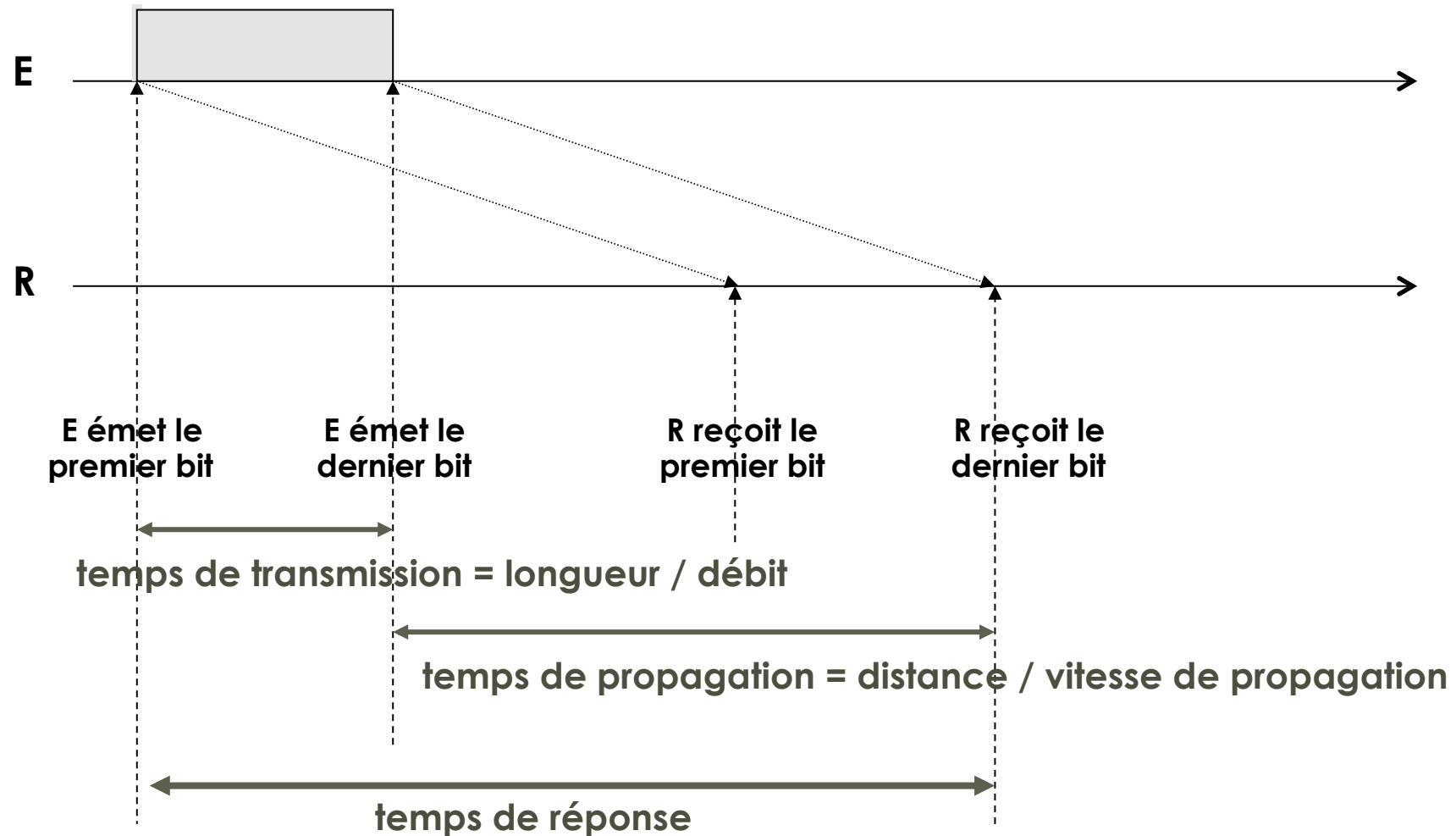
■ Idée

Asservir le taux d'émission de trames au taux d'absorption du récepteur



➤ trame « la suivante »

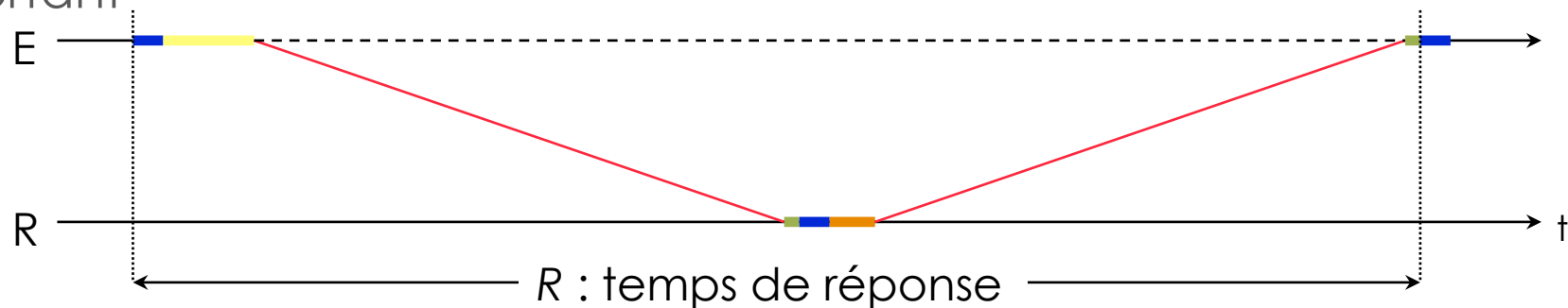
Temps de transmission vs. temps de propagation



Anticipation

■ Problème

Mauvaise utilisation du circuit lorsque le temps de propagation est important



- T_{te} : temps de traitement en émission
- T_{td} : temps de transmission des données
- T_p : temps de propagation
- T_{tr} : temps de traitement en réception
- T_{ta} : temps de transmission de l'acquittement

Anticipation

- Soit ρ le taux d'utilisation du canal de transmission

$$\rho = \frac{T_{td}}{R} \Rightarrow \rho \approx \frac{T_{td}}{T_{td} + 2 T_p} \Rightarrow \rho \approx \frac{l}{l + 2 T_p D}$$

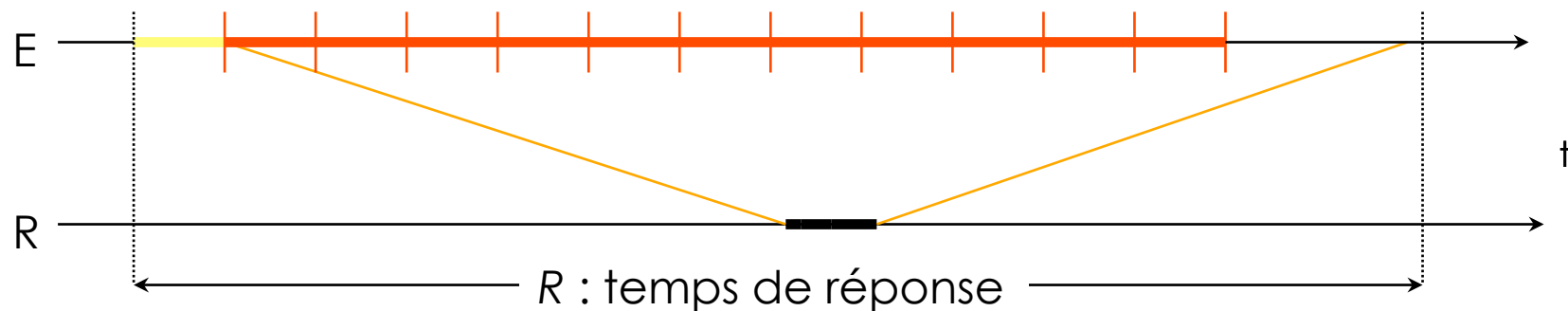
- Exemple d'une liaison satellite

- $D = 2 \text{ Mbit/s}$
- $T_p = 270 \text{ ms}$
- $l = 128 \text{ octets}$
- $\rho = 0,1 \text{ \%} !!!!$

Anticipation

■ Idée

Permettre à l'utilisateur d'envoyer plusieurs frames consécutives avant de se bloquer en attente d'acquittement



➤ **fenêtre d'anticipation de taille W**

Anticipation

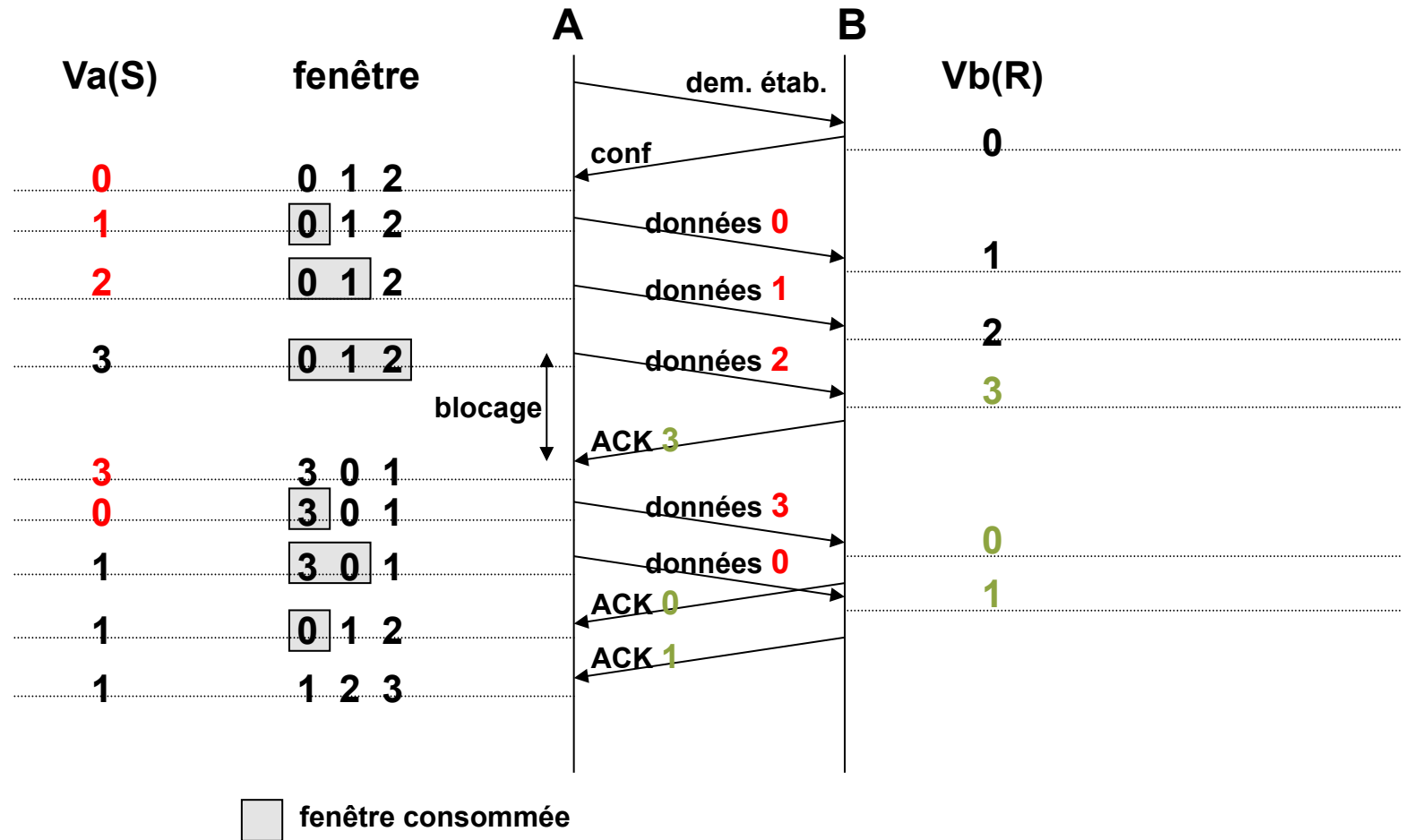
■ Principe

- la fenêtre est une liste de W numéros de séquence
- l'émetteur est autorisé à envoyer les W trames de données dont le $N(S)$ est tel que :

$$\text{dernier } N(R) \text{ reçu} \leq N(S) \leq \text{dernier } N(R) \text{ reçu} + W - 1$$

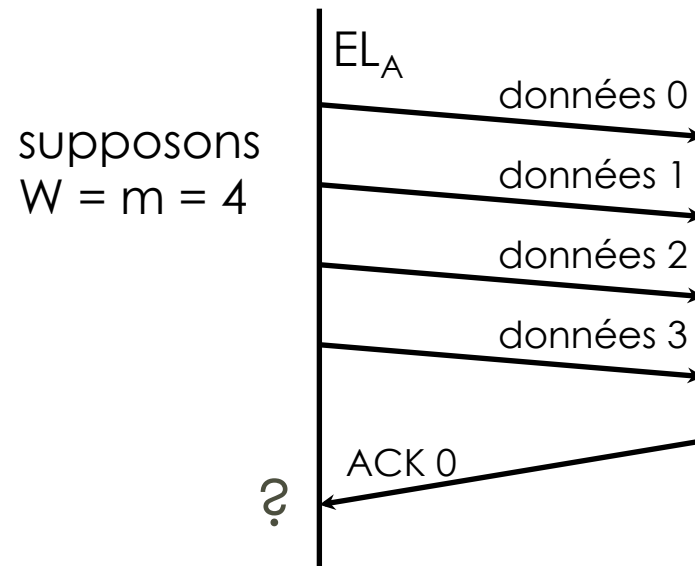
■ Fenêtre coulissante

Exemple : $m = 4$ et $W = 3$



Anticipation

- ▣ Dimensionnement de W
 - ▣ cas du Send-and-Wait : $W = 1$ et $m = 2$
 - ▣ (par absurde) supposons que $W = m$



➤ Règle : $W < m$

Anticipation

- impacts sur les autres mécanismes
 - acquittement positif : plusieurs stratégies
 - un ACK par trame
 - un ACK par fenêtre, etc
 - acquittement négatif : plusieurs stratégies
 - rejet global
 - rejet sélectif
 - régulation de flux
 - pouvoir suspendre temporairement le flux de données
 - pouvoir reprendre le flux de données
- **trame de rejet global**
- **trame de rejet sélectif**
- **trame « Stop! »**

Mécanismes de communication : un bilan

- frames
 - demande d'établissement
 - confirmation
 - demande de libération
 - données
 - acquittement positif
 - rejet global / sélectif
 - stop
- temporisateurs
 - retransmission T1
 - inactivité I
- paramètres
 - modulo de la num. m
 - nb max de transm. N1
 - taille de la fenêtre W

➤ le protocole...

Liaison de données - Plan

- Rôle de la liaison de données
- Mécanismes de communication
- Un exemple de protocole : HDLC

Un exemple de protocole : HDLC

- High-level Data Link Control
- IS 3309-2, IS 4335
- configuration point-à-point ou multipoint
- exploitation en bidirectionnel à l'alternat ou simultané
- fonctionnement en mode connecté
- procédure orientée-bit
- utilisation d'une fenêtre d'anticipation (7 ou 127)

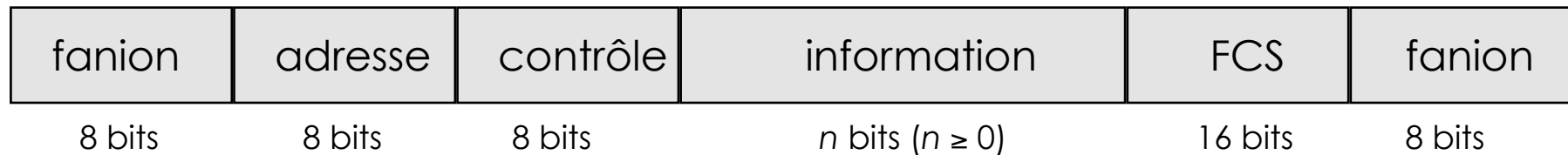
Les modes de fonctionnement HDLC

- NRM (Normal Response Mode)
 - liaison point-à-point ou multipoint
 - gestion hiérarchique (un primaire, des secondaires)
 - exploitation par élection

- ARM (Asynchronous Response Mode)
 - liaison point-à-point ou multipoint
 - gestion hiérarchique (un primaire, des secondaires)
 - exploitation par compétition

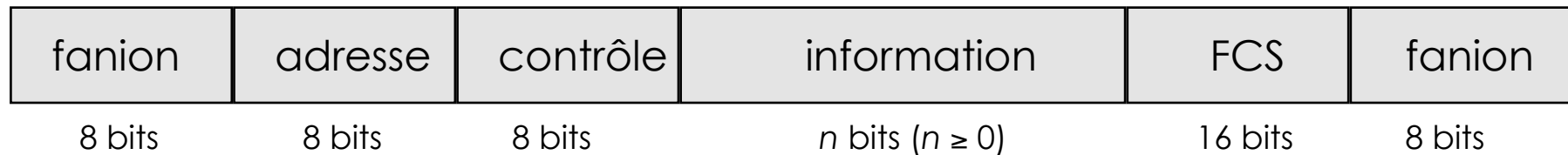
- **ABM** (Asynchronous Balanced Mode)
 - liaison point-à-point
 - gestion symétrique (2 stations identiques)

Structure de la trame HDLC



- fanion : '01111110'
 - délimitation de la trame, début et fin
 - synchronisation
 - mécanisme de transparence : par insertion d'un '0' après cinq '1' consécutifs dans les données
- adresse
 - identification de la station secondaire impliquée dans l'échange
 - trame de commande : la station secondaire destinataire
 - trame de réponse : la station secondaire émettrice

Structure de la frame HDLC



- contrôle
 - type de la frame
 - numéros de séquence
 - extension à 16 bits en modulo 128
- information
 - données de l'utilisateur
 - toute frame de longueur inférieure à 6 octets est non valide
- FCS (Frame Check Sequence)
 - calculé à partir du polynôme V.41 ($x^{16} + x^{12} + x^5 + 1$)

Le champ de contrôle HDLC

frames I
Information



frames S
Supervisory



frames U
Unnumbered



- N(S) : numéro de séquence en émission
- N(R) : numéro de séquence de la prochaine frame I attendue
- P/F :
 - Poll : demande de réponse immédiate pour les commandes
 - Final : réponse au bit P ou frame finale (en NRM)
- jusqu'à 4 frames S différentes
- jusqu'à 32 frames U différentes

Les différentes frames HDLC

- frames I

- frames S

- **RR** - Ready to Receive N(R)
- **REJ** - Reject N(R)
- **SREJ** - Selective Reject N(R)
- **RNR** - Not Ready to Receive N(R)

- frames U

- SNRM, SARM, **SABM** - Set NRM, ARM, ABM
- SNRME, SARME, SABME - Extended
- **UA** - Unnumbered Acknowledge
- **DISC** - Disconnect
- CMDR/FRMR - CoMmanD/FRaMe Reject
- **DM** - Disconnect Mode

Les situations d'anomalie

□ station temporairement occupée

- indication d'un état d'occupation : émission d'un RNR N(R) indiquant le N° de la première trame non acceptée
- indication d'un retour à la normale : émission d'un RR N(R) indiquant le N° de la prochaine trame attendue

□ erreur de transmission

- rejet des trames dont le FCS indique la présence d'erreurs

□ erreur de numéro de séquence

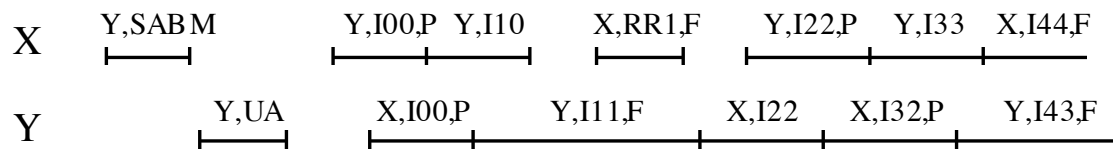
- émission d'un REJ ou d'un SREJ

□ trame non valide

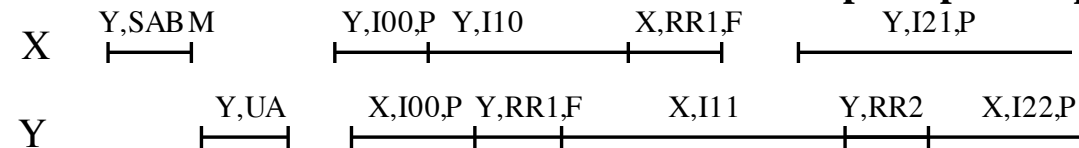
- émission d'un CMDR ou d'un FRMR

Exemples d'échanges en mode ABM

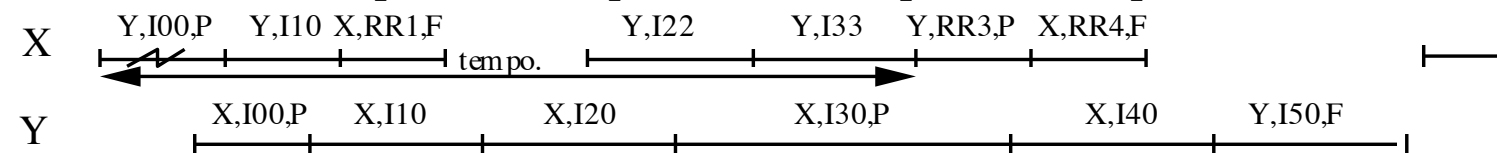
Ouverture et transfert avec accusé de réception normal par frame I



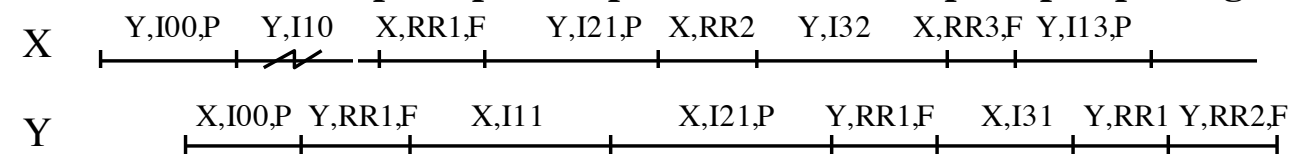
Ouverture et transfert avec accusé de réception préféré par frame RR



Accusé de réception normal par frame I et reprise sur temporisation



Accusé de réception préféré par frame RR et reprise par pointage (P/F)



Une variante : LAP-B

- ▣ Link Access Protocol – Balanced
- ▣ sous-ensemble de HDLC repris par le CCITT
- ▣ mode de fonctionnement équilibré uniquement
- ▣ utilisé dans :
 - ▣ l'interface d'accès **X.25**
 - ▣ les canaux **B** en mode circuit du **RNIS**

Une variante : LAP-D

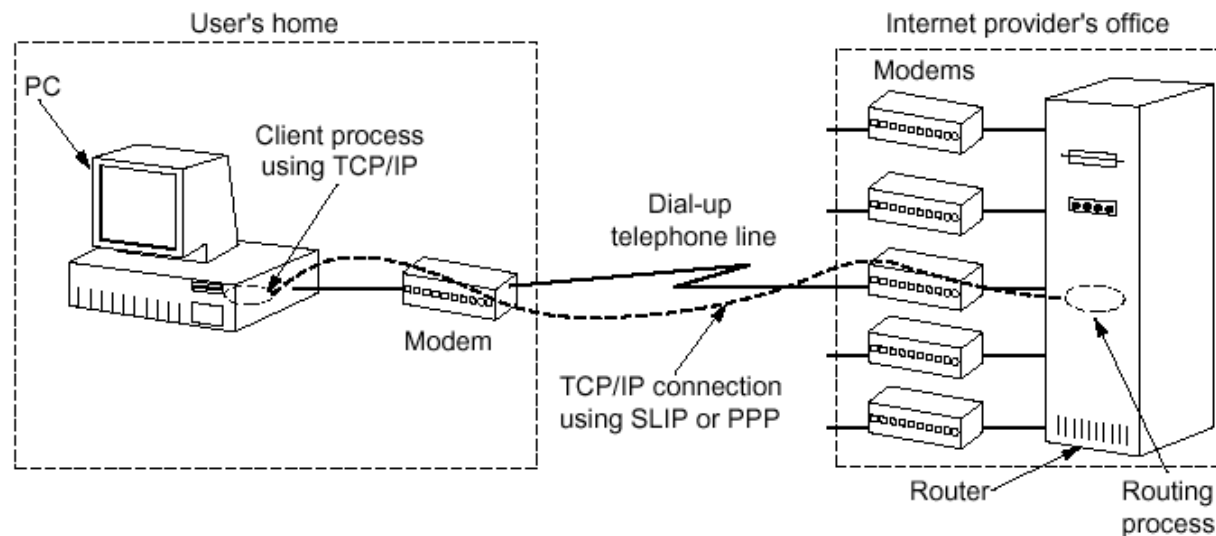
- ▣ développé pour véhiculer les informations sur le canal D en mode paquet du **RNIS**
- ▣ canal D : en multipoint
- champ d'adressage
 - ▣ permet de déterminer un des équipements terminaux reliés au canal D
 - ▣ permet les adresses de diffusion et de multicast
 - ▣ codé sur 2 octets

Une variante : LAP-F

- extension du LAP-D pour le **relais de frames**
- le champ d'adresse ne véhicule pas d'adresse mais un identificateur de connexion (*Data Link Connection Identifier*)

Liaison point-à-point dans l'Internet

- rôle majeur pour les accès domestiques par modem via un FAI
 1. le PC domestique se met en liaison avec un routeur
 2. il se comporte ensuite comme n'importe quel hôte du réseau



- protocole de liaison de données nécessaire
 - SLIP
 - PPP

Le protocole SLIP

- Serial Line IP
- fonctionnalités
 - délimitation de trames
 - les paquets IP bruts sont délimités par le fanion 0xC0
 - transparence au fanion :
 - si 0xC0 apparaît dans le paquet, il est remplacé par (0xDB, 0xDC)
 - si 0xDB apparaît dans le paquet, il est doublé
- inconvénients
 - ☹ pas de contrôle d'erreur (ni détection ni correction)
 - ☹ mono-protocole (IP seulement)
 - ☹ les 2 extrémités doivent connaître a priori leurs adresses réciproques → pas d'affectation dynamique
 - ☹ pas d'authentification
 - ☹ ce n'est pas un standard IETF → versions différentes et incompatibles

Le protocole PPP

- Point to Point Protocol
- variante de HDLC
- fonctionnalités
 - délimitation de trames
 - détection d'erreurs
 - négociation d'adresses
 - authentification
 - multi-protocole
- champ supplémentaire
 - de deux octets
 - déterminant le protocole de niveau supérieur
 - 0x0021 : IPv4
 - 0x002B : IPX
 - 0x800F : IPv6
 - situé après le champ de contrôle du format HDLC

Etat de notre architecture

