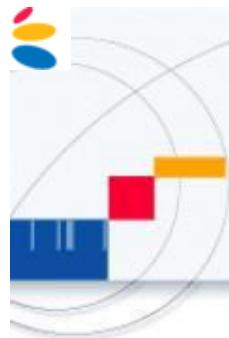


Réseaux Informatiques

Filière: SMI - S5

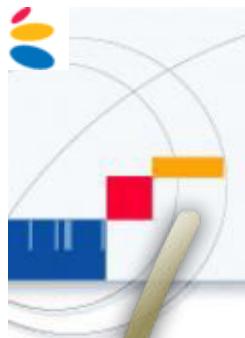


Pr. K. HOUSNI
Faculté des sciences
Université Ibn Tofail



Chapitre III

La couche liaison de données Les concepts fondamentaux



Chapitre III

Définition

Rôle de la couche liaison de données

La délimitation de données: notion de fanion, notion de transparence

- ✓ Le taux d'erreur binaire
- ✓ La détection d'erreur
- ✓ La détection par clé calculée

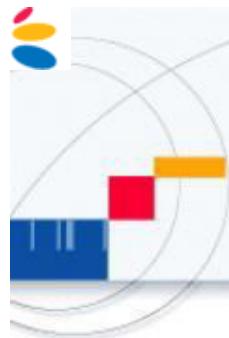
Contrôle de l'échange

- ✓ Les mécanismes de base
- ✓ Contrôle de flux
- ✓ Les modes de connexion

Le protocole de la couche liaison de données

- ✓ Le protocole HDLC
- ✓ Structure de la trame HDLC
- ✓ Le champs commande
- ✓ Structure d'un dialogue HDLC





La couche liaison de données

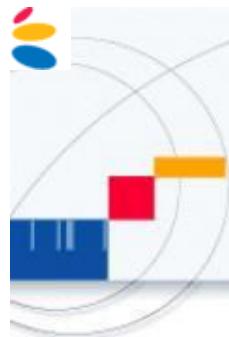
Définition:

La couche liaison de données fournit les moyens fonctionnels et procéduraux nécessaires à l'établissement, au maintient et à la libération des connexions de liaison de données entre entités du réseau.

Elle **déetecte** et **corrige**, si possible, les erreurs dûes au support physique et signale à la couche réseau les erreurs irrécupérables.

Elle supervise le fonctionnement de la transmission et définit la structure syntaxique des messages appelés **trames**.

Une **trame** est une suite binaire de taille bornée contenant des informations de types " **données** " et/ou des **informations de contrôle** nécessaires pour réaliser les fonctions de ce niveau.



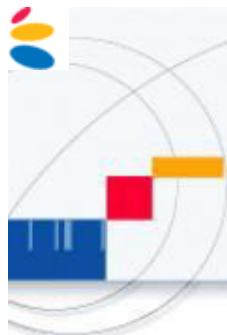
Rôle de la couche liaison de données

Alors que le circuit de données (couche physique) est capable de transmettre des éléments binaires, les protocoles de liaison de données travaillent sur des blocs d'éléments binaires (les trames). La trame est l'unité de base que gère le protocole de liaison de données.

La trame transporte les données de l'utilisateur mais elle contient aussi des informations de contrôle (l'entête) qui sont nécessaires au protocole pour le bon déroulement du dialogue.

Définir un protocole de liaison de données consiste à préciser :

- le format des trames,
- le critère de début et de fin de trames,
- la technique de détection d'erreur utilisée,
- la place et la signification des différents champs dans une trame,
- les règles de dialogue : les procédures après détection d'erreur ou de panne et la supervision de la liaison.

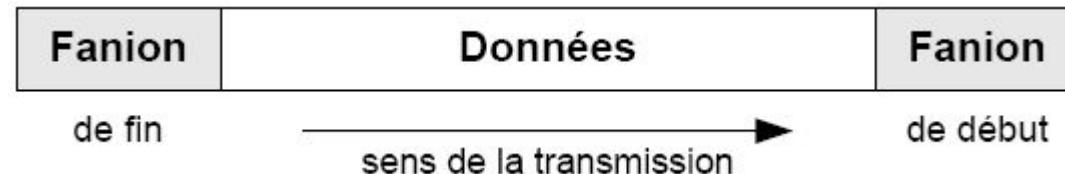


La délimitation de données: notion de fanion

Lors d'une transmission de données, il faut pouvoir repérer le début et la fin de la séquence des données transmises

bit de "start" et bit de "stop" en transmission asynchrone

en transmission synchrone on utilise un fanion (flag) = une séquence de bits particulière



Les trames sont des blocs composés d'un nombre quelconque de bits et on **parle de protocole orienté bit**. Le fanion sert à délimiter les trames.

La suite d'éléments binaires **01111110** est utilisée comme fanion dans le cas de protocole HDLC.

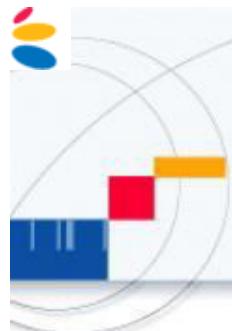
Un mécanisme de transparence est nécessaire pour éviter l'apparition de la séquence du fanion à l'intérieur de la trame → **technique du bit de bourrage**.

Il consiste, **en émission**, à insérer dans le corps de la trame un élément binaire de valeur 0 après avoir rencontré 5 éléments binaires consécutifs de valeur 1.

En réception, si on rencontre 5 éléments binaires consécutifs de valeur 1, l'automate regarde le bit suivant

s'il est à "1", il s'agit du fanion

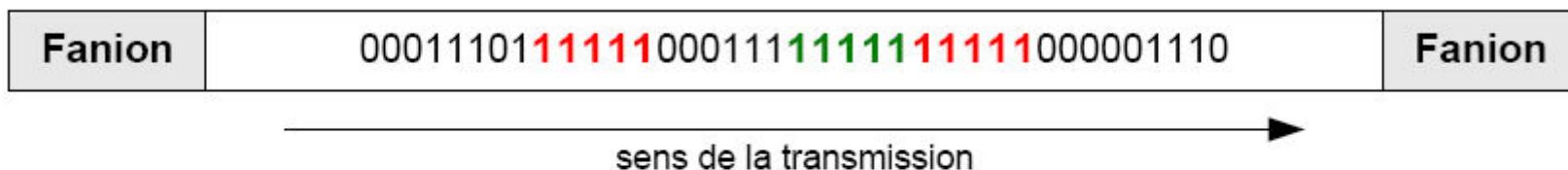
s'il est à "0", le "0" est enlevé de la séquence (il a été introduit à l'émission)



La délimitation de données: notion de transparence

La technique du bit de bourrage (**sens de transmission à refaire**)

Séquence originale

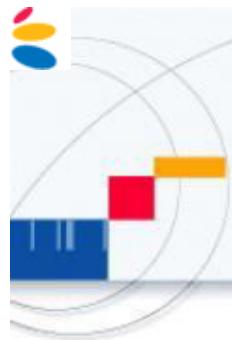


Séquence transmise



Avec un tel mécanisme, on interdit donc l'émission de plus de 5 éléments binaires de valeur 1 sauf pour la délimitation de trames (Les fanions).

Cette méthode a l'avantage de permettre la transmission de trames de longueur variable, sans limitation, mais elle introduit des variations sur la durée de transmission des données utilisateur.



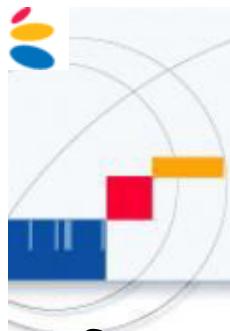
Contrôle de la validité de l'information

Notion d'erreur

Les circuits de données peuvent introduire des perturbations (erreurs de transmission), il faut pouvoir détecter ces erreurs.

Ceci est réalisé en introduisant une redondance dans la transmission et en vérifiant à la réception que cette redondance est conservée.

Si des erreurs arrivent, il est nécessaire de spécifier des procédures de correction des erreurs détectées.



Le taux d'erreur binaire

Sur une ligne de transmission finie, le bruit peut perturber le signal et être source d'erreurs de transmission.

BER = Bit Error Rate

TEB = Nb bits erronés / Nb bits transmis

Exemple

L'émetteur transmet la suite 0001110101101010

Le récepteur reçoit la suite 00011**0**01011**1**101**1**

TEB = 3/16 = 0,1875

En pratique

Réseaux locaux : TEB=10⁻⁹

TEB représente généralement la probabilité de recevoir un bit erroné

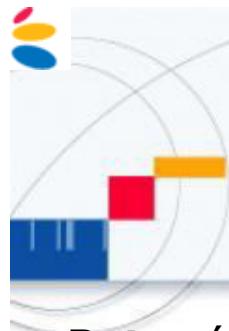
La probabilité de recevoir correctement un bloc de N bits est alors :

$$p = (1-TEB) \dots (1-TEB) = (1-TEB)^N$$

La probabilité de recevoir un bloc erroné est alors :

$$p = 1 - (1-TEB)^N$$

Plus la longueur d'un bloc est grande, plus la probabilité de réception correcte est faible !



La détection d'erreur

But : vérifier la validité des données reçues chez le destinataire

Idée : ajouter une certaine redondance dans l'information transmise

4 techniques :

- la détection par écho
- la détection par répétition
- la détection d'erreur par clé calculée
- la détection et correction d'erreur par code

la détection par écho

-Le récepteur renvoie chaque message reçu (écho)

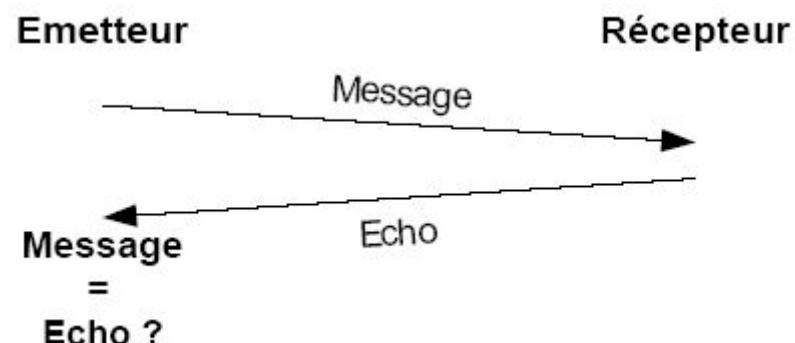
-L'émetteur compare l'écho au message initial et renvoie ce dernier si les deux messages sont différents

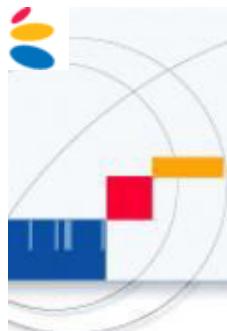
-Utilisée par terminaux asynchrones
(telnet, minitel, ...)

Problèmes

redondance totale

l'écho peut lui-même être erroné

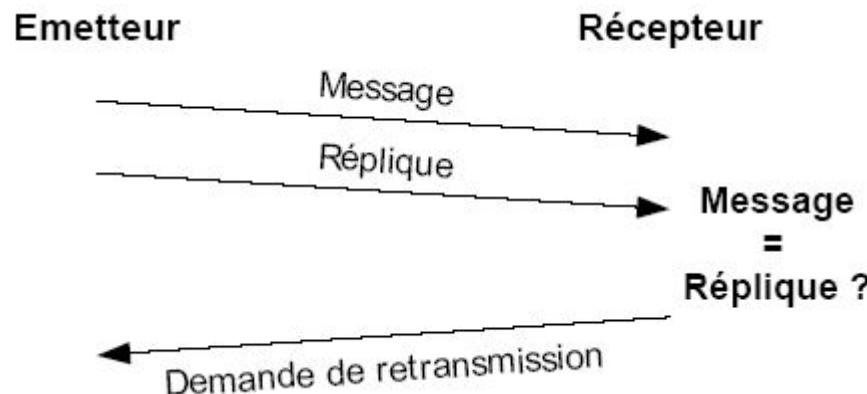




La détection d'erreur

la détection par répétition

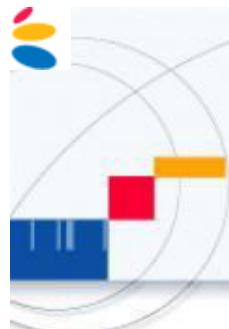
- Chaque message émis est suivi de sa propre réplique
- Si les deux messages sont différents, le récepteur demande une retransmission



Utilisée dans les milieux sécurisés très perturbés
(applications temps réel)

Problèmes

- redondance totale
- la réplique peut être erronée



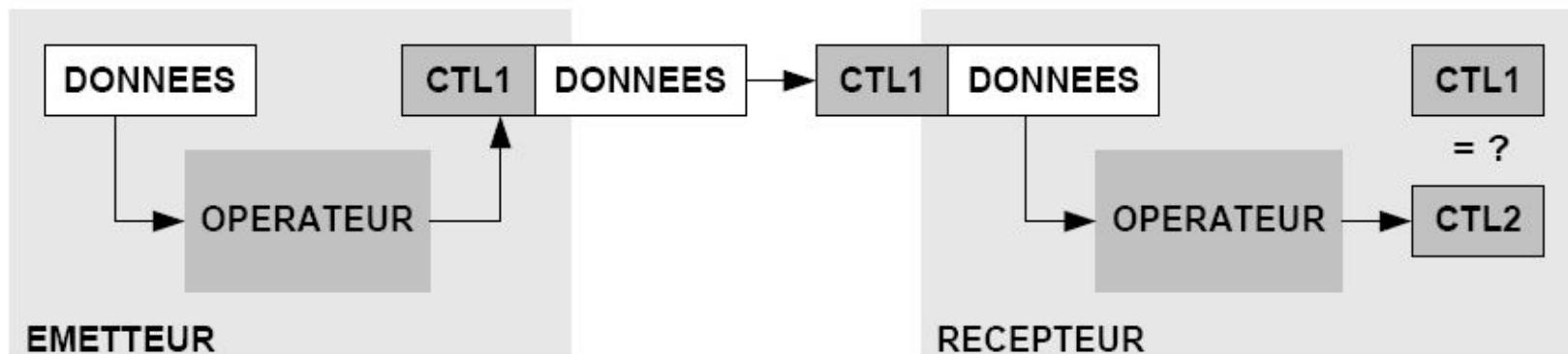
La détection par clé calculée

la détection d'erreur par clé calculée

L'émetteur ajoute au message une information supplémentaire (clé noté CTL - **Control** -) calculée à partir du message d'origine

Le récepteur recalcule la clé selon la même méthode à partir des informations reçues et compare à la clé reçue

Le récepteur ignore les données si les clés sont différentes et peut demander la retransmission (reprise sur erreur)



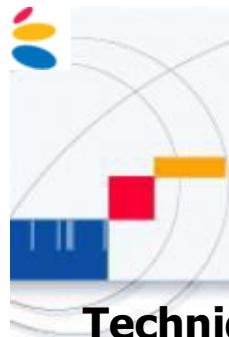
La clé est parfois appelée

CRC : Cyclic Redundancy Check

FCS : Frame Check Sequence

Problème

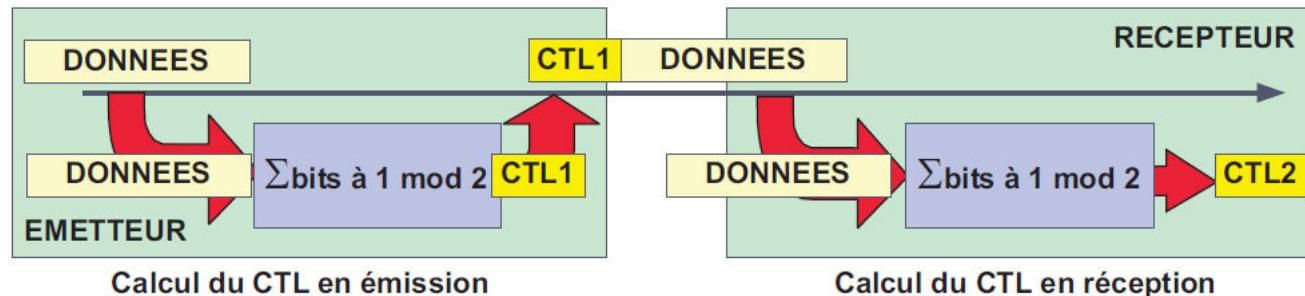
La clé peut elle-même être corrompue



La détection par clé calculée

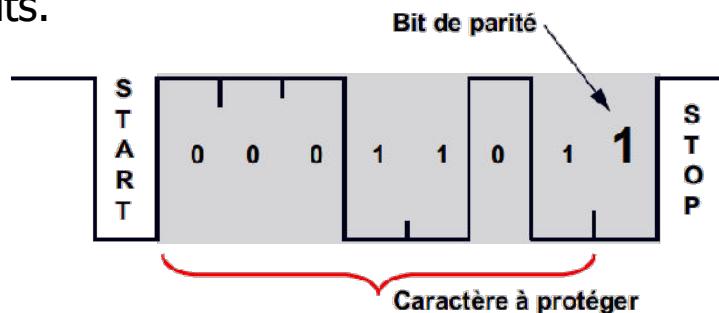
Technique dite du bit de parité

La technique du bit de parité consiste à ajouter, à la séquence binaire à protéger, un bit, telle que la somme des bits à 1 transmis soit paire (bit de parité) ou impaire (bit d'imparité).

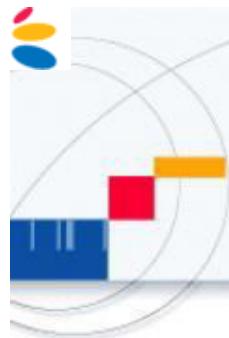


Les caractères ASCII (7 bits) sont protégés par l'introduction d'un 8^{eme} bit : le bit de parité.

Cette technique, connue sous le nom de **VRC (Vertical Redundancy Check)**, vérification par redondance verticale ne permet de détecter que les erreurs portant sur un nombre impair de bits.



Caractère	O	S	I
Bit 6	1	1	1
Bit 5	0	0	0
Bit 4	0	1	0
Bit 3	1	0	1
Bit 2	1	0	0
Bit 1	1	1	0
Bit 0	1	1	1
Bit de parité	1	0	1
Bit d'imparité	0	1	0



La détection par clé calculée

Technique dite du bit de parité (Parité croisée)

Dans les transmissions synchrones, les caractères sont envoyés en blocs. La technique du bit de parité est insuffisante, elle est complétée d'une autre information : le **LRC** (**Longitudinal Redundancy Check**).

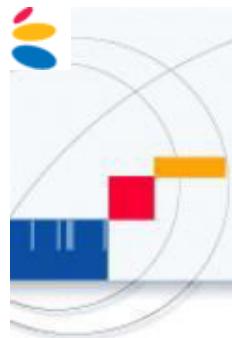
Caractère à transmettre	bit de parité	Caractère à transmettre	bit de parité	...	Caractère LRC	bit de parité
-------------------------	---------------	-------------------------	---------------	-----	---------------	---------------

Un caractère le LRC est ajouté au bloc transmis. Chaque bit du caractère LRC correspond à la parité des bits de chaque caractère de même rang : le premier bit du LRC est la parité de tous les 1er bits de chaque caractère, le second de tous les 2e bits...

Le caractère ainsi constitué est ajouté au message. Le LRC est lui-même protégé par un bit de parité (VRC).

	H	E	L	L	O	LRC →
bit 0	0	1	0	0	1	0
bit 1	0	0	0	0	1	1
bit 2	0	1	1	1	1	0
bit 3	1	0	1	1	1	0
bit 4	0	0	0	0	0	0
bit 5	0	0	0	0	0	0
bit 6	1	1	1	1	1	1
VRC ↓	0	1	1	1	1	0

1001000	0	1000101	1	1001100	1	1001100	1	1001111	1	1000010	0
H		E		L		L		O		LRC	



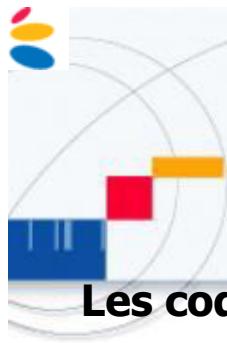
Parité croisée: efficacité

- Permet de **corriger** toutes les erreurs simples
- Permet de **déetecter** toutes les erreurs sur 2 ou 3 bits
- Permet de **déTECTER** toutes les rafales plus courtes que la longueur d'une ligne
- ! 4 erreurs bits peuvent passer sans être détectées

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

Parités
horizontales

Parités verticales



La détection par clé calculée

Les codes à redondance cyclique

(CRC : Cyclique Redundancy check)

Sont nommés aussi codes polynomiaux

basés sur le traitement des suites de bits comme étant une représentation polynomiale avec les coefficients 0 et 1.

Principe:

on considère une suite de bits par exemple : 110111

A cette suite on fait correspondre le polynôme suivant :

$$(1 \cdot x^5) + (1 \cdot x^4) + (0 \cdot x^3) + (1 \cdot x^2) + (1 \cdot x^1) + 1 \rightarrow x^5 + x^4 + x^2 + x + 1$$

Remarque : Les calculs sont faits en binaire (modulo 2 : $1+1=0$; $X+X=0$; $X=-X$)
[par exemple, $(x^7 + x^3) + (x^3 + x) = x^7 + x$]

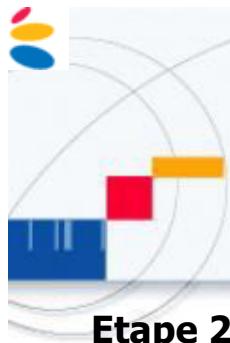
Etape 1 :

On définit un polynôme connu de l'émetteur et du récepteur appelé polynôme générateur, soit r son degré :

$G(x) = x^r + \dots + 1$; il faut que les termes des deux extrêmes soient non nuls (x^r, x^0)

Codes normalisés

$x^{16} + x^{12} + x^5 + 1$ Réseaux X25



La détection par clé calculée

Etape 2 :

soit M le message à transmettre (de m bits) dont le polynôme correspondant est $(M(x))$ de degré $m-1$.

On ajoute au message M , r bits (appelés bits de contrôle, CRC (ou FCS : Frame Check Sequence) pour envoyer une trame P de n bits

$(n=m+r)$ dont le polynôme correspondant $P(x)$ de degré $n-1$ est divisible par $G(x)$

$P(x)$ est calculé suivant la méthode suivante :

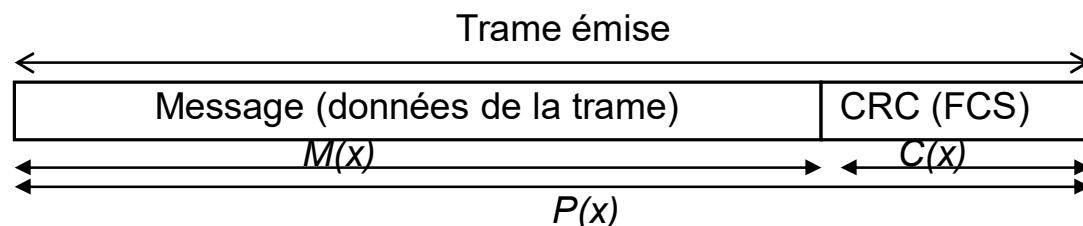
on multiplie $M(x)$ par x^r : on a le polynôme $x^r \cdot M(x)$ Ceci équivaut à un décalage de $M(X)$, de r positions vers la gauche.

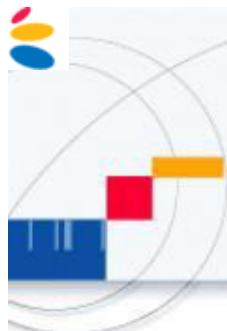
on divise $x^r \cdot M(x)$ par $G(x)$, soit :

$$x^r M(x) = Q(x)g(x) + C(x)$$

$C(x)$ est le reste de cette division. Par définition son degré est $r-1$

on envoie le mot P (de $n=m+r$ bits) composé de m bits du message M suivi des r bits de FCS dont le polynôme correspondant est $C(x)$.





La détection par clé calculée

En réception :

Le récepteur effectue la division du mot reçu par $G(x)$. Si le résultat est nul, il conclut qu'il n'y a pas d'erreur.

Problème

Les erreurs qui produisent un polynôme multiple de $G(x)$ ne sont pas détectées.

Exemple CRC :

Polynôme générateur : x^2+x+1 Message : 110111

$$(x^5 + x^4 + 0x^3 + x^2 + x^1 + 1).x^2 = x^7 + x^6 + 0x^5 + x^4 + x^3 + x^2 + 0 + 0$$

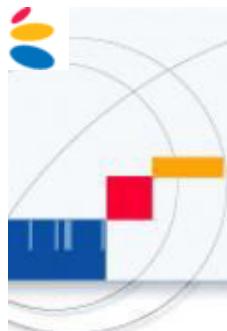
Message transmis : 110111 11

Décodage : division de 110111 11 par 111.

On contrôle si le reste est nul.

$$\begin{array}{r} x^7 + x^6 + 0 + x^4 + x^3 + x^2 + 0 + 0 \\ \hline x^7 - x^6 - x^5 \quad \downarrow \quad \downarrow \quad \downarrow \\ \quad x^5 - x^4 - x^3 \quad \downarrow \\ \quad x^5 - x^4 - x^3 \quad \downarrow \\ \qquad \qquad x^2 \quad 0 \quad 0 \\ \qquad \qquad x^2 \quad x \quad 1 \\ \text{RESTE} \Rightarrow \qquad \qquad x \quad 1 \end{array}$$

Le reste de la division polynomiale est de degré inférieur à celui du diviseur, la division est terminée.



Détection et correction d'erreur par code

Code de Hamming

Code de Hamming

Le code de Hamming est calculé à partir d'une mesure de dissimilarité entre deux séquences de bits de même longueur, appelée distance de Hamming.

Définition (distance de Hamming)

La distance de Hamming entre deux séquences binaires m_1 et m_2 de même taille est le nombre de bits de même rang par lesquels ces deux séquences diffèrent. Elle est notée $d(m_1; m_2)$.

Exemple : $d(1\mathbf{1}00110; \mathbf{1}0\mathbf{1}0110) = 2$.

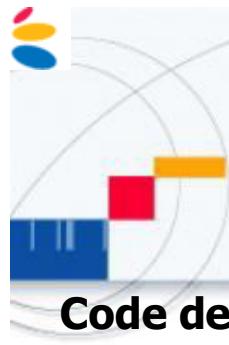
1 $\mathbf{1}$ 00110

10 $\mathbf{1}$ 0110

Propriété

Un code de distance $d(C)$ détecte $d(C) - 1$ erreurs et corrige $k = \left\lfloor \frac{d(c)-1}{2} \right\rfloor$ erreurs.

→ La distance de Hamming minimale doit être la plus grande possible



Détection et correction d'erreur par code

Code de Hamming

- Structure d'un mode de code de Hamming

les **m bits du message** à transmettre et les **n bits de contrôle de parité**.

longueur totale : $2^n - 1$

longueur du messages : $m = (2^n - 1) - n$

→ on parle de code $x - y$ où $x = n + m$ et $y = m$.

- Efficacité du code : rapport y/x

- Exemple de code de Hamming :

un mot de code **7 - 4** a un coefficient d'efficacité de **$4/7 = 57\%$** ,

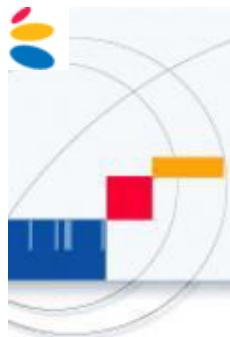
un mot de code **15 - 11** a un coefficient d'efficacité de **$11/15 = 73\%$** ,

un mot de code **31 - 26** a un coefficient d'efficacité de **$26/31 = 83\%$** ,

- Les bits de contrôle de parité C_i sont en position 2^i pour $i=0,1,2,\dots$

- Les bits du message D_j occupe le reste du message

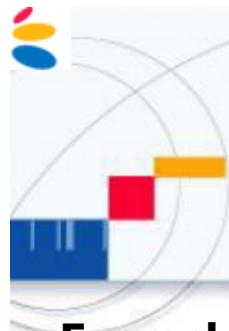
1	2	3	4	5	6	7
C0	C1	D0	C2	D1	D2	D3



Détection et correction d'erreur par code

Méthode de Hamming :

- ✓ bits numérotés de 1 à n
- ✓ **bits puissance de 2 sont les r bits de contrôle (1,2,4,8...)**
- ✓ **les autres sont les m bits de données (3,5,6,7,9,10...)**
- ✓ bits de contrôle = calcul de parité sur certains bits de données : **ceux dont la décomposition en puissance de 2 fait intervenir le bit de contrôle concerné**
- ✓ Chaque bit de donnée est contrôlé par les bits de contrôle qui entrent en compte dans sa décomposition en somme de puissances de 2.
- ✓ **Exemple : le bit $11=8+2+1$ est vérifié par les bits 8, 2 et 1.**



Détection et correction d'erreur par code

Exemple

- ✓ Le message à transmettre

a	b	c	d
---	---	---	---
- ✓ **bits puissance de 2 sont les r bits de contrôle (1,2,4,8...)**
- ✓ **les autres sont les m bits de données (3,5,6,7,9,10...)**
- ✓ bits de contrôle = calcul de parité sur certains bits de données : **ceux dont la décomposition en puissance de 2 fait intervenir le bit de contrôle concerné**

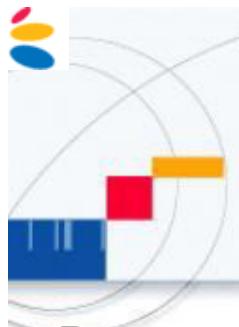
1	2	3	4	5	6	7
x	y	A	z	B	C	D

001	010	011	100	101	110	111
x	y	a	z	b	c	d

$$x = a + b + d$$

$$y = a + c + d$$

$$z = b + c + d$$



Détection et correction d'erreur par code

Retrouver l'erreur dans un mot de Hamming

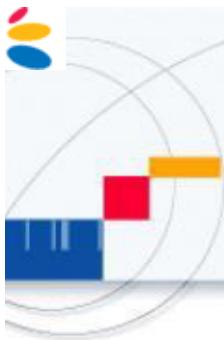
Correction

C2' vaut $1 + 0 + 1 + 0 = 0$ (bits d'indice 7, 6, 5 et 4).

C1' vaut $1 + 0 + 1 + 1 = 1$ (bits d'indice 7, 6, 3 et 2).

C0' vaut $1 + 1 + 1 + 0 = 1$ (bits d'indice 7, 5, 3 et 1).

→ $C_2'C_1'C_0''$ vaut 011, c'est à dire 3 en base 10. Il y a donc une erreur à l'indice 3 du mot

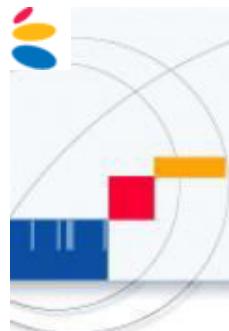


Exercice

- Soit un mot de Hamming de longueur 15

1	0	1	1	0	1	1	1	1	0	1	1	0	1	1
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1

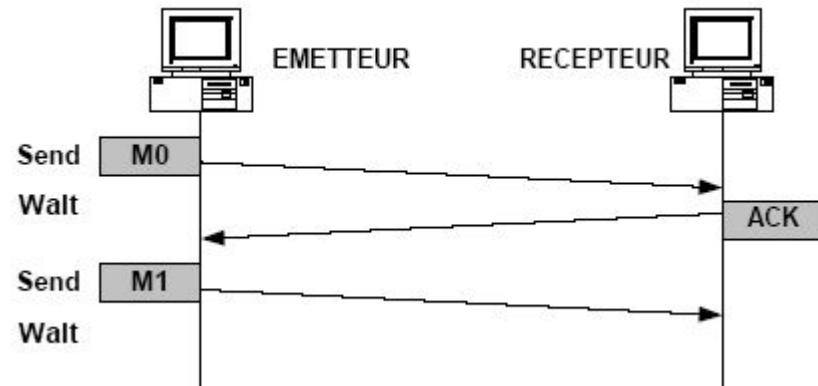
- Quels sont les bits de contrôle de parité ?
- Quel est le message reçu ?
- Est-ce que le message reçu correspond au message transmis ?
- Quel a été le message transmis ?



Contrôle de l'échange: mécanismes de base

Le mode Send & Wait

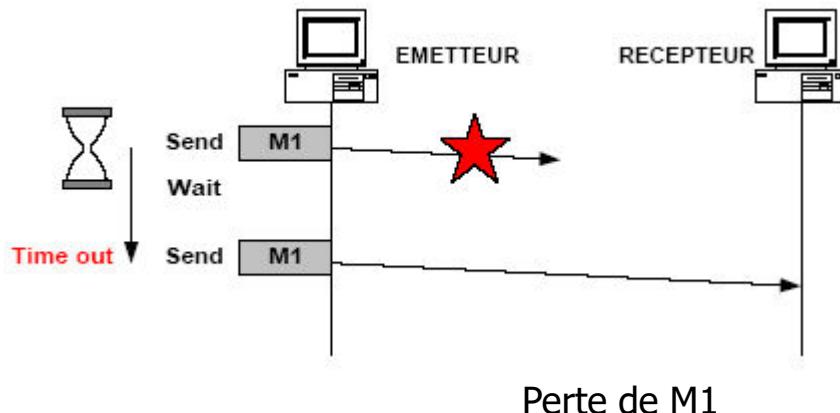
lorsqu'une trame est bien reçue, la station réceptrice envoie une trame d'acquittement ACK, après réception de l'ACK l'émetteur peut transmettre une nouvelle trame.

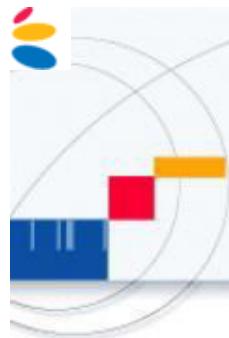


Problème : l'émetteur peut rester bloqué indéfiniment si M0 ou ACK est perdu

Solution : La reprise sur temporisation

Time out = compteur

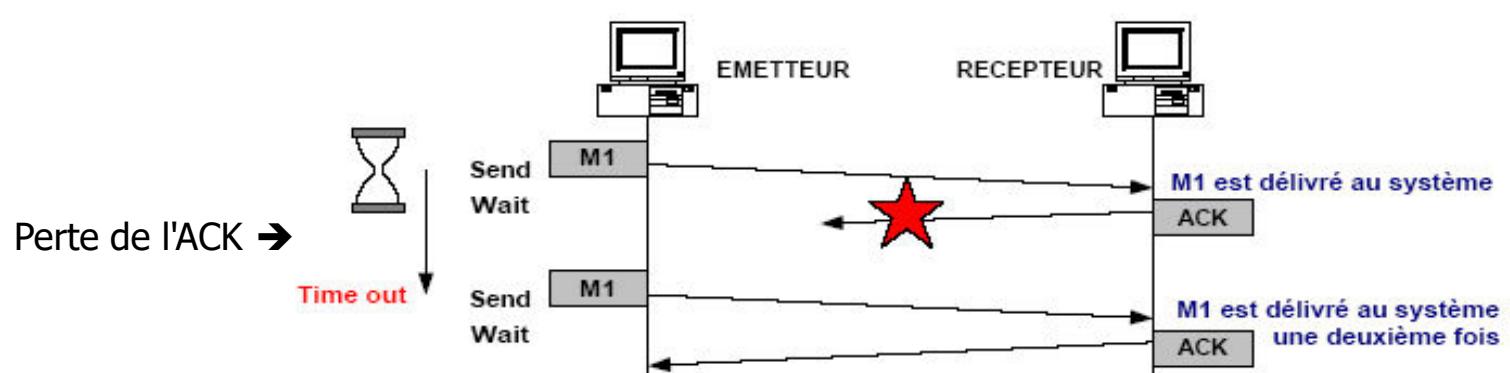


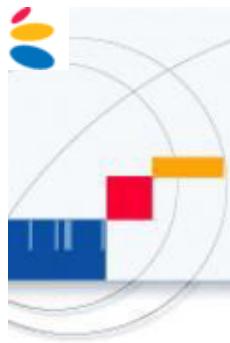


Contrôle de l'échange: mécanismes de base

Le mode Send & Wait

Problème : si l'ACK est perdu, les données sont retransmises alors qu'elles ont déjà été reçues ; le message est alors dupliqué chez le récepteur





Contrôle de l'échange: mécanismes de base

Solution : Numérotation des messages émis

On utilise 2 compteurs (Ns en émission, Nr en réception)

Ns et Nr sont initialisés à zéro

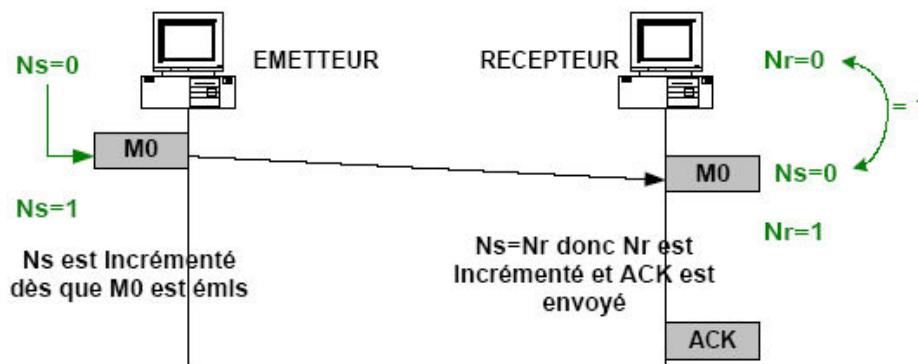
Ns contient le numéro du prochain message à émettre

Nr contient le numéro du prochain bloc à recevoir

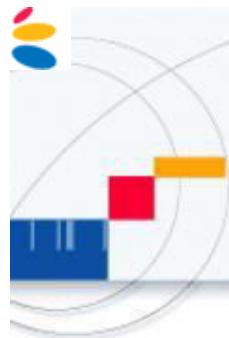
Ns est transmis de l'émetteur vers le récepteur

Un message n'est délivré côté récepteur que si le Ns reçu est égal au Nr local

Si $Ns < Nr$, le message a déjà été reçu, le récepteur le "jette" et l'acquitte de nouveau



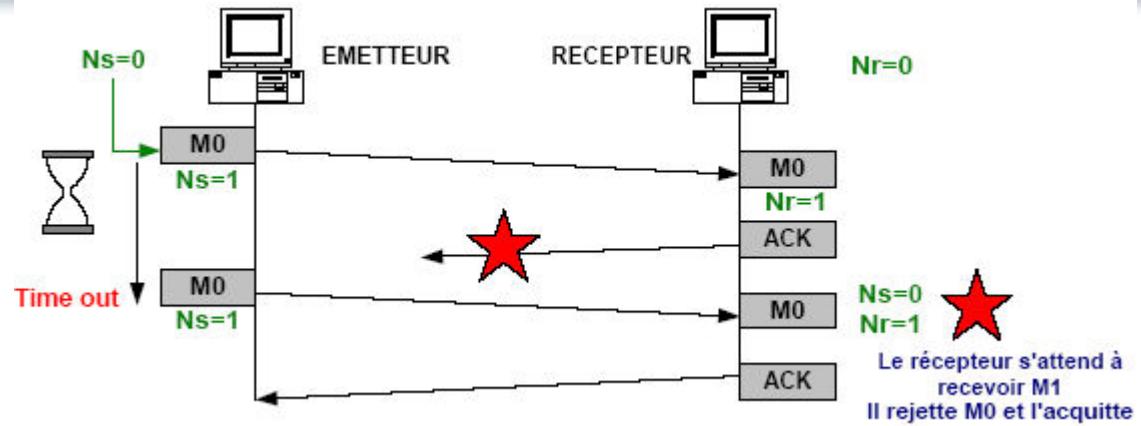
→ La numérotation des messages, évite la duplication et permet le contrôle de séquencement des données reçues



Contrôle de l'échange: mécanismes de base

Problème :

Perte de l'acquittement



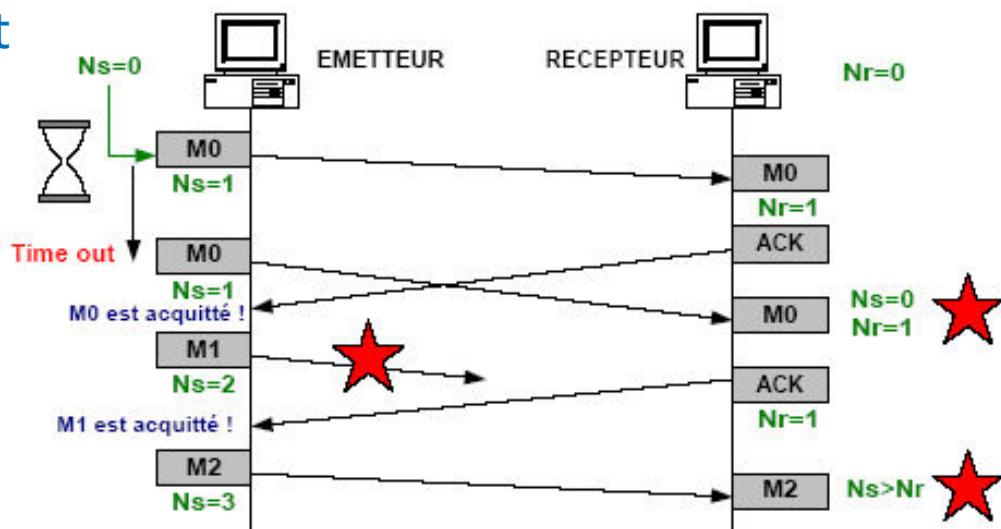
Le deuxième MU reçu est rejeté

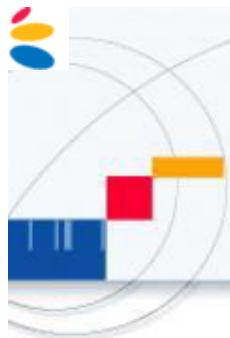
Remarque : le timer doit être bien réglé

Si le délai d'acquittement trop important

Problème : M1 n'a jamais été reçu et pourtant il est acquitté

Solution : il faudrait numéroté aussi les acquittements !





Contrôle de l'échange: mécanismes de base

Attend t-on pour envoyer M_{i+1} que M_i soit acquitté ?

Soit on utilise le mode Send&Wait, → on attend

Si on n'attend pas, il faut pouvoir :

l'émetteur stocke les messages non acquittés

→ numéroter les acquittements

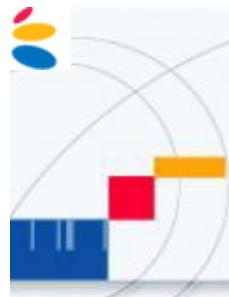
$N_s > N_r$ est-il possible ?

Possible si on envoie M_{i+1} alors que M_i n'est pas acquitté

Les messages n'arrivent alors pas dans le bon ordre sur le récepteur

→ soit on refuse les messages tels que $N_s > N_r$

→ soit on stocke les messages désordonnés sur le récepteur



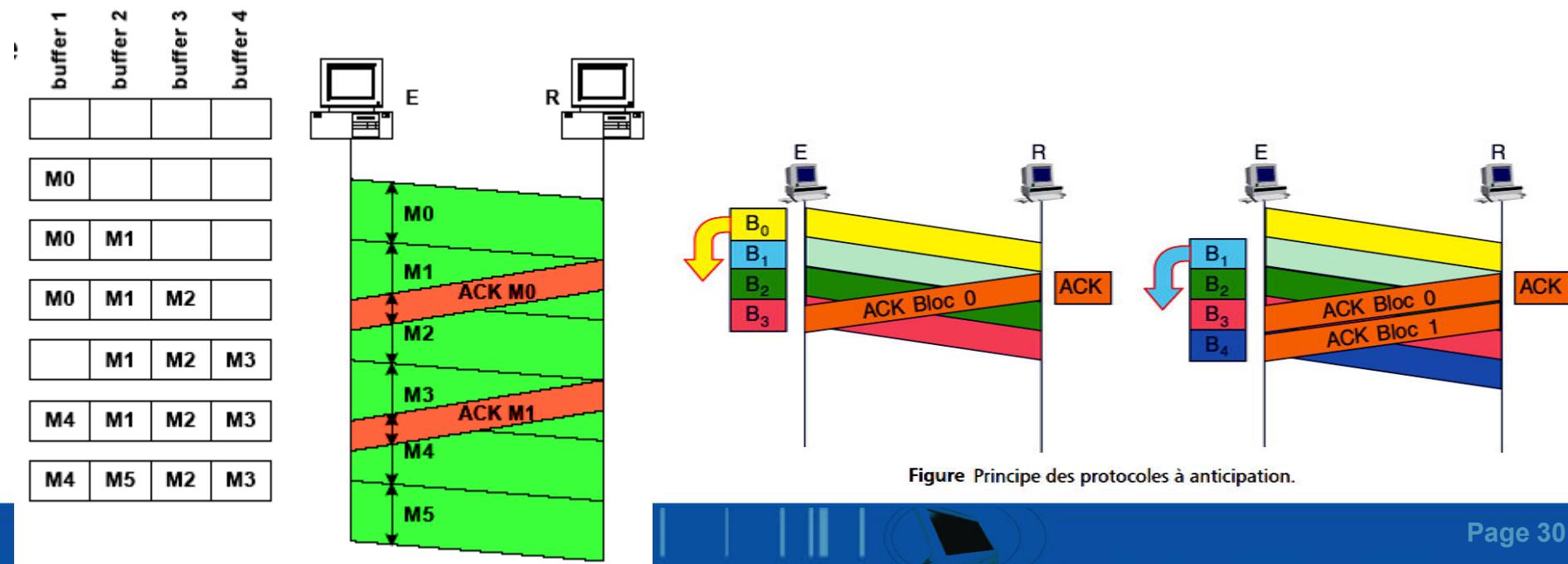
Les protocoles à anticipation

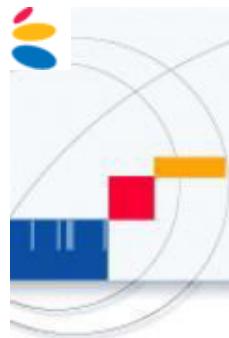
Dans le mode Send & Wait les performances sont dégradées du fait de l'attente de l'ACK avant d'envoyer un nouveau message.

Solution : Protocole à anticipation

- l'émetteur peut faire plusieurs émissions successives sans attendre l'ACK des messages précédents
- il faut mémoriser TOUS les messages non acquittés sur l'émetteur dans des "buffers"
 - quand un ACK arrive, l'émetteur peut libérer le buffer correspondant au(x) message(s) acquitté(s)
 - s'il n'y a plus de buffer libre, l'émetteur doit attendre l'arrivée d'un ACK pour continuer d'émettre

Principe :





Les protocoles à anticipation

Fenêtre d'anticipation (notée W)

crédits d'émission dont dispose l'émetteur

Taille optimale de la fenêtre quand l'émission se fait en continu
(l'émetteur n'attend jamais un ACK)

$$W \text{ optimale} = T_a / T_b$$

RTT :Round Trip Time, temps aller et retour

T_b représente le temps d'émission d'un bloc

T_a :temps d'attente, temps entre l'émission du premier bit
de la trame N et le premier bit de la trame $N + 1$ en mode Send and Wait,

Gestion glissante de la fenêtre

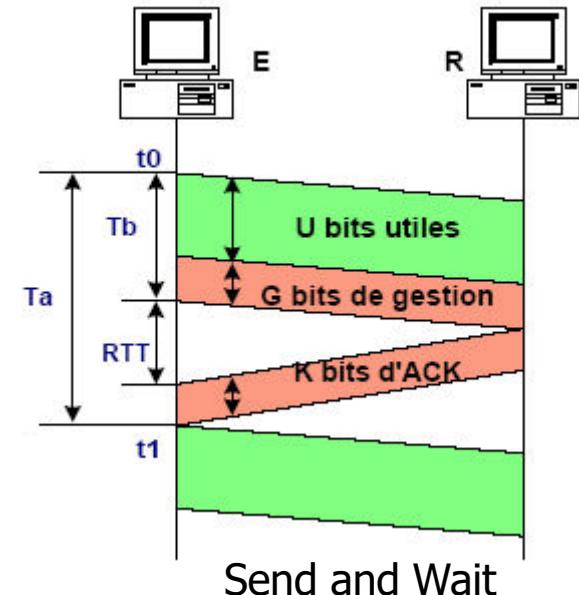
Chaque bloc est acquitté. Lors de la réception d'un ACK, l'émetteur libère un buffer et émet le bloc suivant.

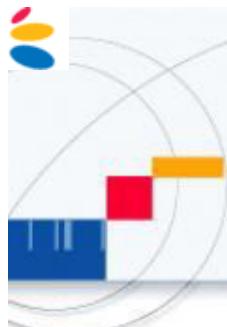
Gestion sautante de la fenêtre

l'acquittement est différé et concerne plusieurs messages

si $W=3$, M0, M1 et M2 sont acquittés en une seule fois

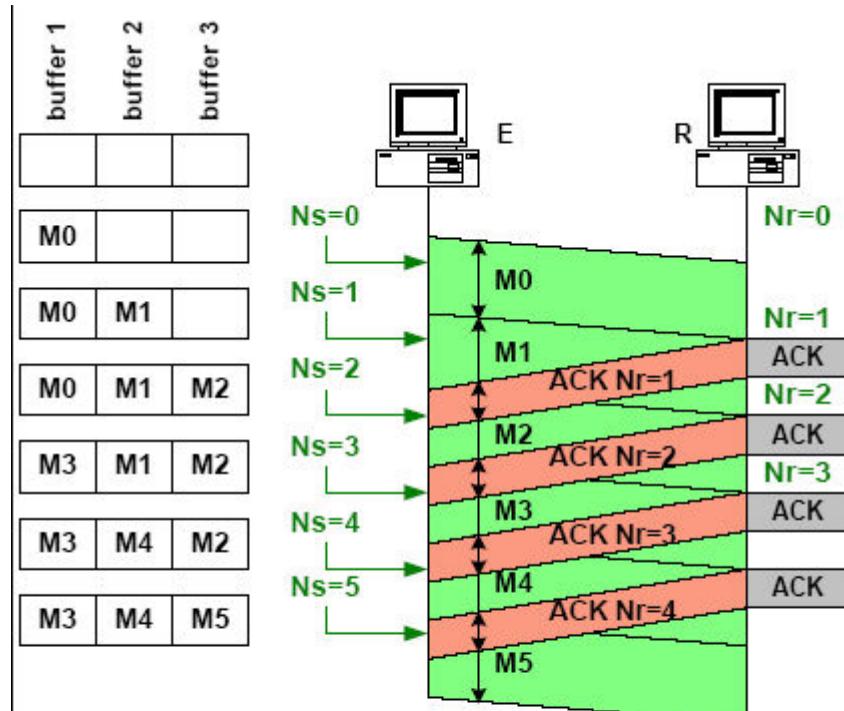
les émissions s'arrêtent quand les crédits d'émission sont épuisés



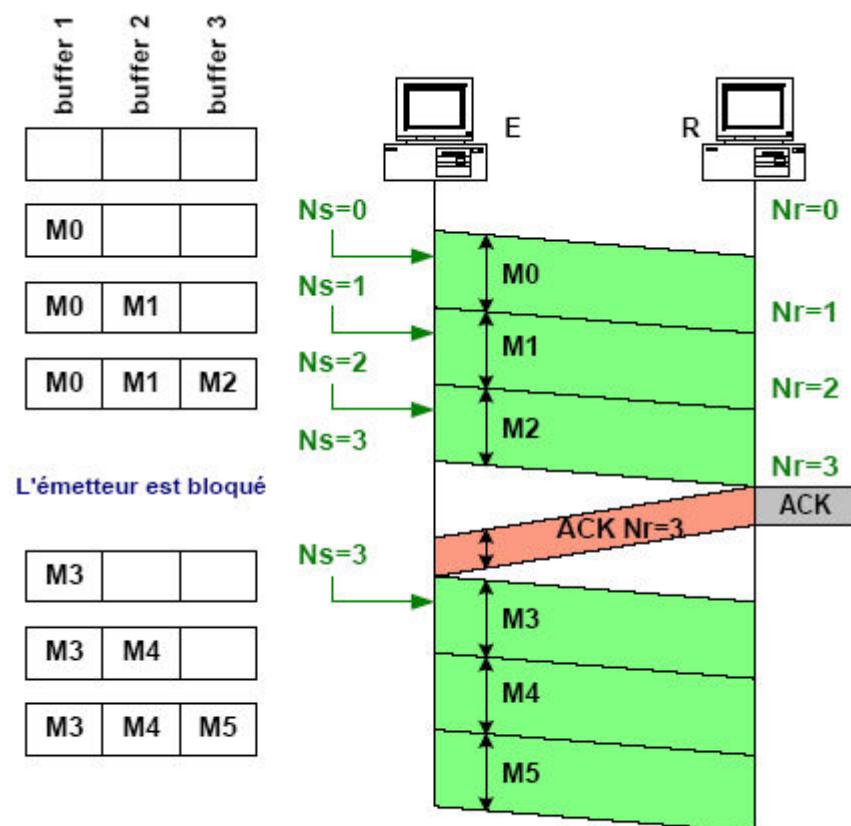


Les protocoles à anticipation

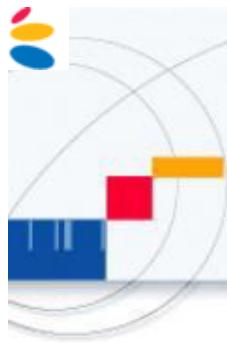
Gestion de la fenêtre avec W=3



Gestion glissante de la fenêtre



Gestion sautante de la fenêtre



Les protocoles à anticipation

Les politiques de reprise sur erreur

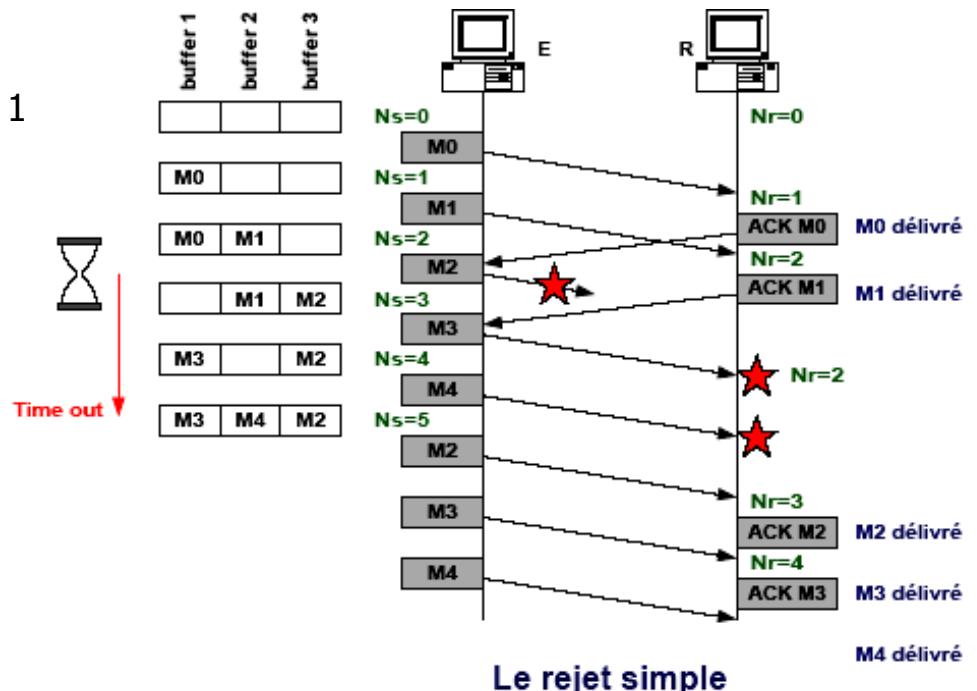
Rejet simple

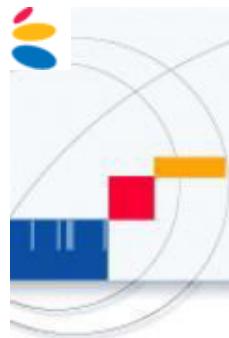
tous les blocs reçus hors séquencement sont rejettés

le protocole est dit "Go Back N"

l'émetteur reprend la transmission à partir du message perdu

mémoire du récepteur minimisée, $W_{réception} = 1$





Les protocoles à anticipation

Les politiques de reprise sur erreur

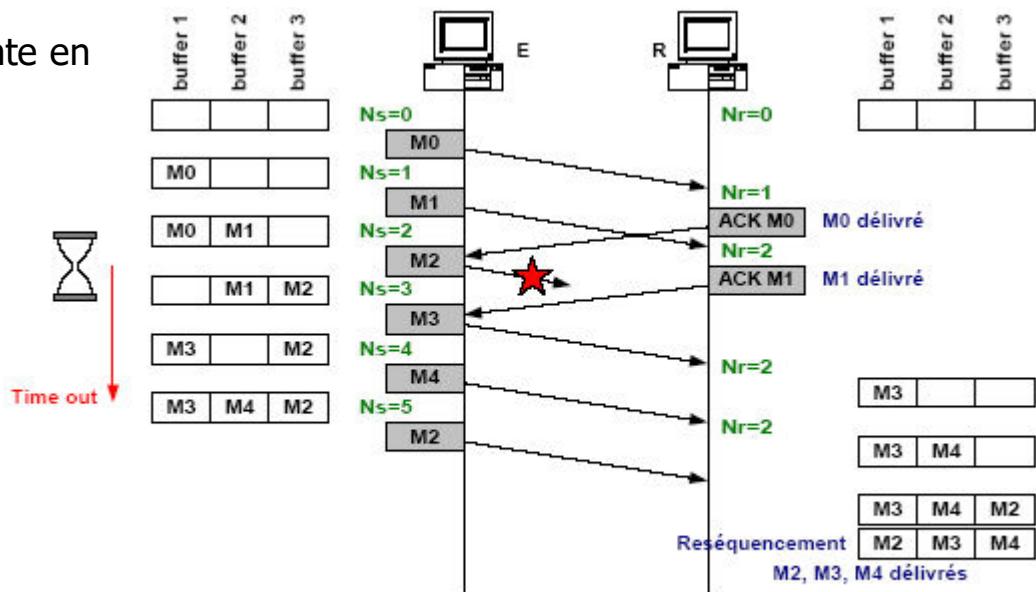
Rejet sélectif

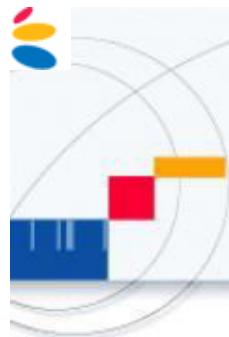
Le récepteur mémorise les messages hors séquencement

l'émetteur ne retransmet que les messages erronés

W réception = nombre de messages déséquencés pouvant être reçus

transmission optimisée - mémoire importante en réception





Le contrôle de flux

Le Nombre de buffer (tampons mémoire) sur le récepteur est limité, l'émetteur ne doit pas émettre plus de données que le récepteur ne peut en accepter sinon les paquets seront perdus.

Le contrôle de flux est le mécanisme qui consiste à asservir **la cadence d'émission** de l'émetteur sur les capacités de réception du récepteur.

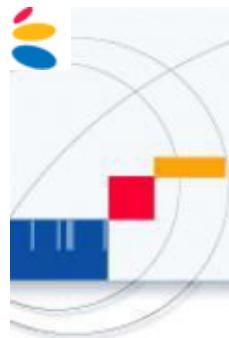
On appelle **crédit d'émission (Ct)** le nombre de blocs que l'émetteur est autorisé à transmettre
Il y a deux types de contrôle de flux :

Contrôle de flux implicite

- le nombre de crédits est fixé une fois pour toute ;
- quand l'émetteur a épuisé ses crédits, il attend l'autorisation du récepteur pour reprendre l'émission.

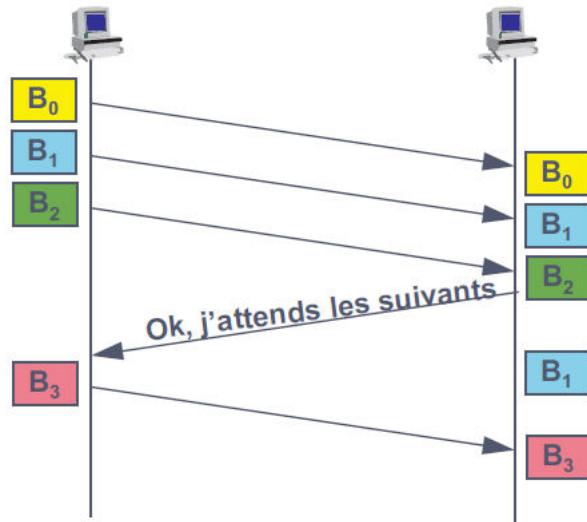
Contrôle de flux explicite ou dynamique

- le récepteur informe en permanence l'émetteur sur ses capacités de réception ;
- le message du récepteur contient le nouveau nombre de crédits disponibles.



Le contrôle de flux

Contrôle de flux implicite



Contrôle de flux explicite ou dynamique

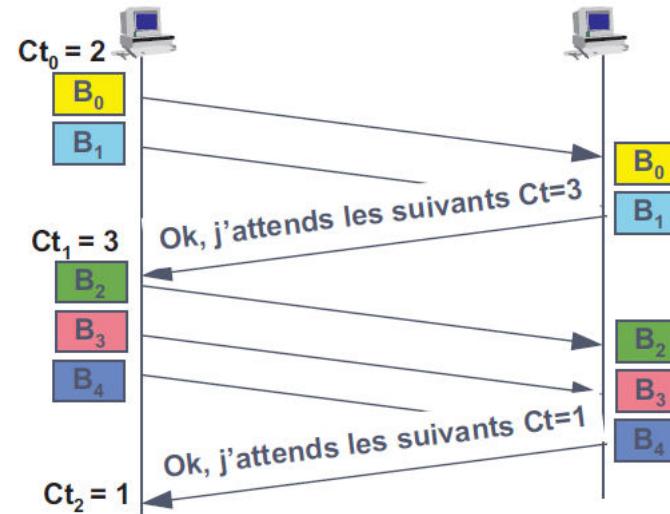
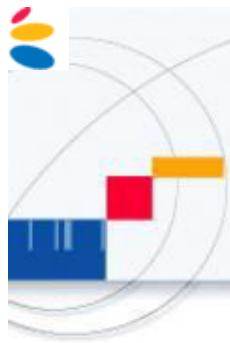


Figure Contrôle de flux



Les modes de communication

On peut distinguer deux grands modes de communication:

- communication en mode connecté (appelé aussi "with connection")
- communication en mode non connecté (appelé aussi "connectionless" ou par abus de langage "datagramme")

Le mode non connecté :

1 seule phase : le transfert de données

- chaque unité de transfert de données est acheminée indépendamment
- les entités communicantes ne mémorisent rien ("memoryless").
- les messages échangés sont auto-suffisants ("self-content")

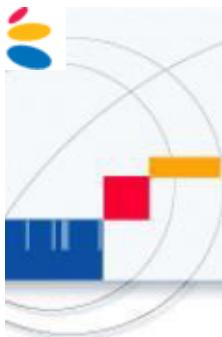
Le mode connecté :

3 phases :

- phase d'établissement de la connexion
- phase de transfert de données
- phase de libération de la connexion

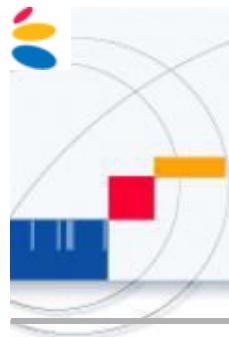
un contexte (réparti) est partagé par les membres de la connexion

- permet (facilite) le contrôle et la gestion du transfert de données :
contrôle d'erreur, contrôle de flux, maintien en séquence, etc.



Le protocole HDLC

- HDLC - High Level Data Link Control, Protocole de niveau 2 OSI
- Premier protocole moderne, normalisé en 1976 par le CCITT (UIT-T)
- Repose sur la **transmission synchrone orientée bit**.
- Nécessite une liaison physique synchrone full-duplex
- Il met en œuvre un mécanisme de transparence par fanion
rend le protocole totalement indépendant de la taille des données
- Sa variante la plus connue est le mode LAP-B
utilise un mécanisme de contrôle de flux.
Il est équilibré ou symétrique : les deux stations ont les mêmes prérogatives.
Il utilise des moyens de transmission duplex intégral sur une liaison point à point exploitée en full-duplex ;



Le protocole HDLC

Différentes modes de HDLC :

Le mode normal ou synchrone

NRM - Normal Response Mode
ou LAP - Link Access Protocol

relation maître/esclave
(sollicitation du primaire)

Le mode asynchrone

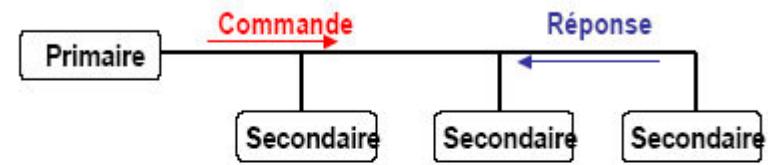
ABM - Asynchronous Balanced Mode

chaque extrémité est primaire en émission
et secondaire en réception (mode équilibré)

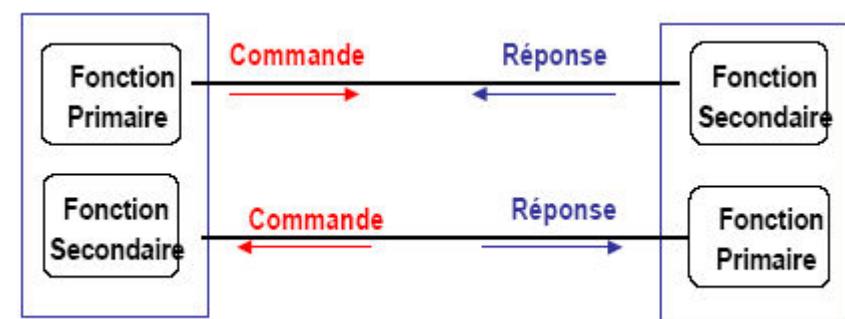
liaison point à point uniquement

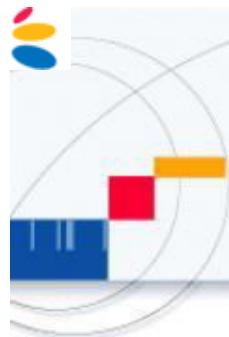
full duplex (LAP Balanced - RNIS)

- Multipoint



- Point à point





Le protocole HDLC

Toutes les informations sont transportées dans une structure unique : la trame. (Frame)

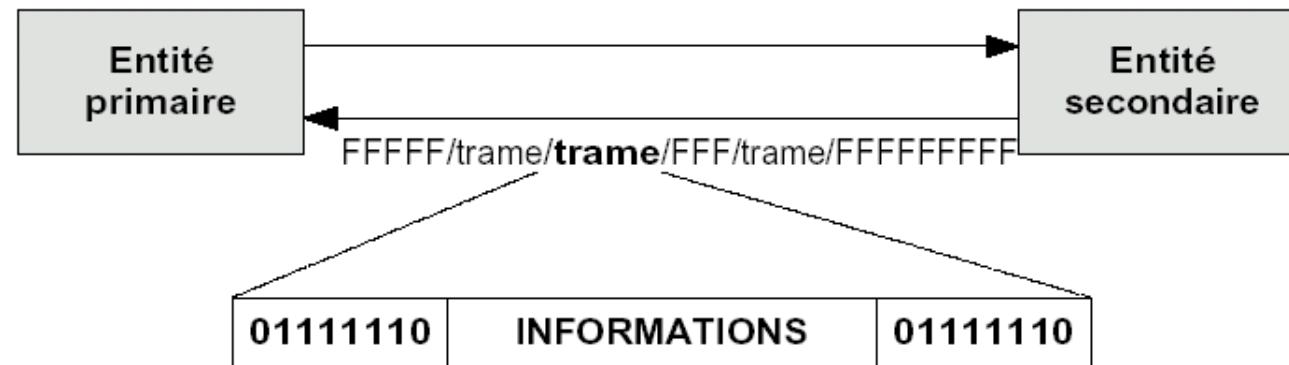
La trame est de longueur variable et délimitée par un fanion (flag);

En l'absence de données, le fanion est envoyé pour maintenir la synchronisation entre les trames;

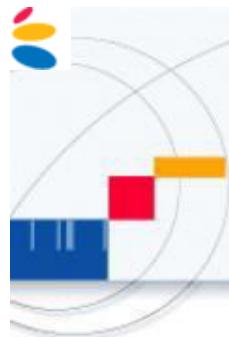
En cas d'émission consécutive de trames, le fanion marque la fin d'une trame et le début de la suivante;

La transparence est assurée par la technique du bit de bourrage;

Contrôle d'erreurs très efficace par CRC avec reprise sur erreur.



L'entité est dite primaire si elle initie la communication
protocole basé sur l'élément binaire ("orienté" bit)

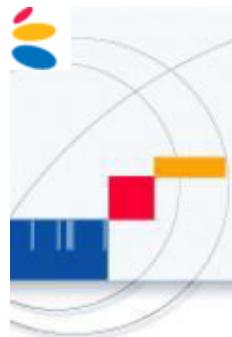


Structure d'une trame HDLC

Format de la trame (Frame)

Fanion deb	Adresse	Commande	INFORMATIONS	FCS	Fanion fin
1 octet	1 octet	1 ou 2 octets		2 octets	1 octet

- Fanion** = 01111110
- Le champ **Adresse** s'étend sur un octet et identifie une des extrémités de la liaison. fut conçu pour des lignes multipoints. Il peut supporter jusqu'à 256 terminaux par ligne.
- Le champ **Commande** décrit le type de la trame : il s'étend sur 1 octet mais peut être porté à 2 octets dans le mode appelé mode étendu.
- Le champ **Information** est un champ facultatif contenant un nombre quelconque d'éléments binaires représentant les données de l'utilisateur.
- Le champ **FCS** (Frame Check Sequence) est une séquence de contrôle de trame (elle est obtenue par un contrôle polynomial de polynôme générateur
$$x^{16} + x^{12} + x^5 + 1$$
).
- Le champ de gauche est le premier transmis, le champ de droite est le dernier.



Structure d'une trame HDLC

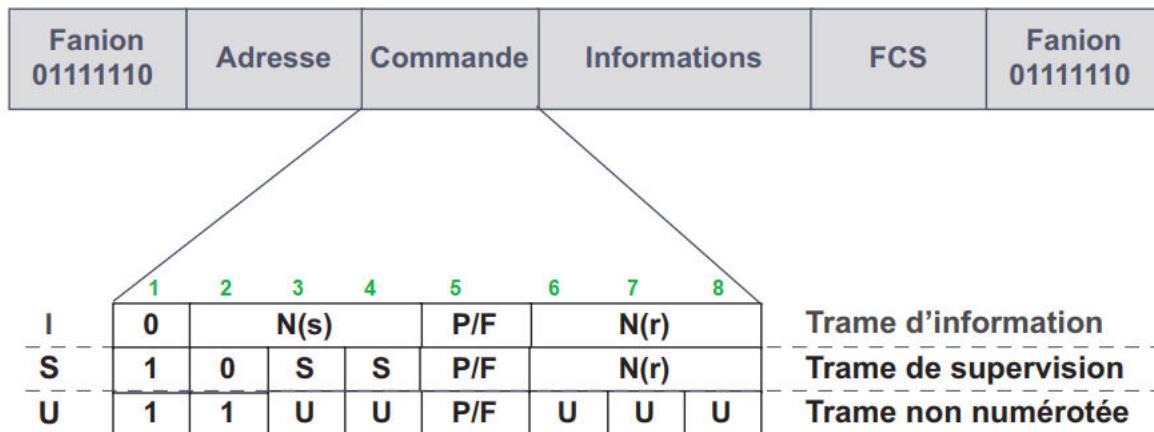
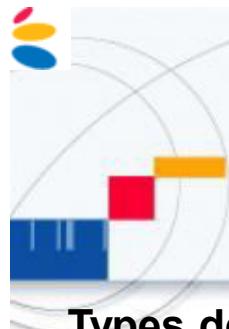


Figure Structure du champ de commande.



Structure d'une trame HDLC

Types de trames :

Il existe trois types de trames qui sont identifiés par le champ Commande.

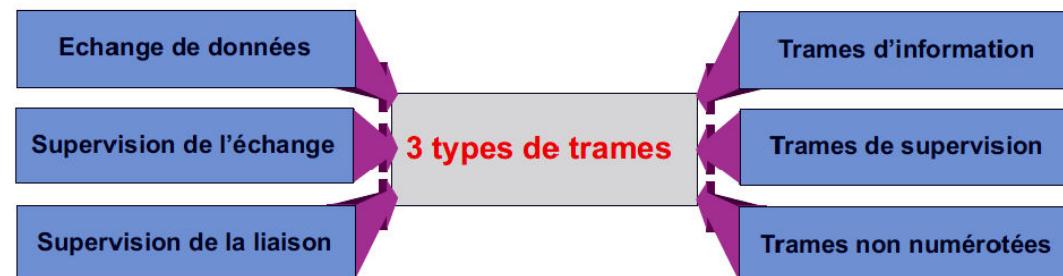
La trame d'information ou trame I : permet la transmission de données de l'utilisateur.

Les trames de supervision ou trames S : permettent l'acquittement et le contrôle de flux.
Elles ne transportent pas de données.

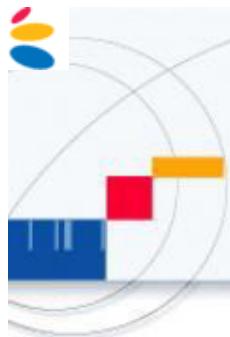
Les trames non numérotées ou trames U (Unnumbered) : sont utilisées pour toutes les fonctions de contrôle de la liaison telles que l'initialisation, la libération... Elles ne transportent pas de données.

Le champ Commande :

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
	Nr		P/F		Ns		0	trame I
	Nr		P/F	s	s	0	1	trame S
U	U	U	P/F	U	U	1	1	trame U



Les fonctions et trames correspondantes d'HDLC.

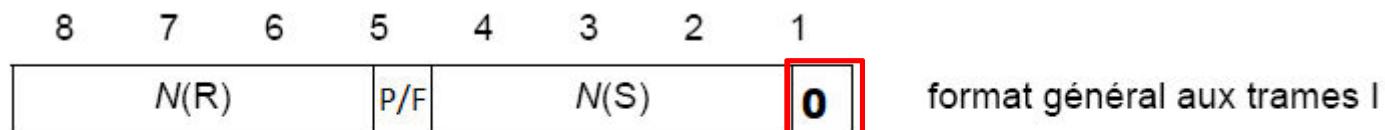


Le champs commande

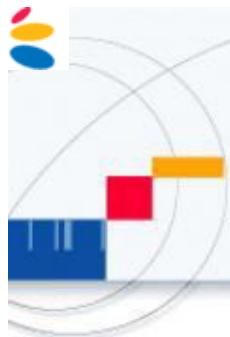
La trame I (Information) :

La trame I permet la transmission des données. Elle est numérotée "le compteur Ns". Elle permet également l'acquittement des trames échangées dans le sens inverse grâce au compteur Nr.

- Ns - compteur des trames I émises
- Nr - compteur des trames I reçues
- Nr contient le numéro de la prochaine trame attendue
- Nr = n acquitte les (n-1) trames précédentes

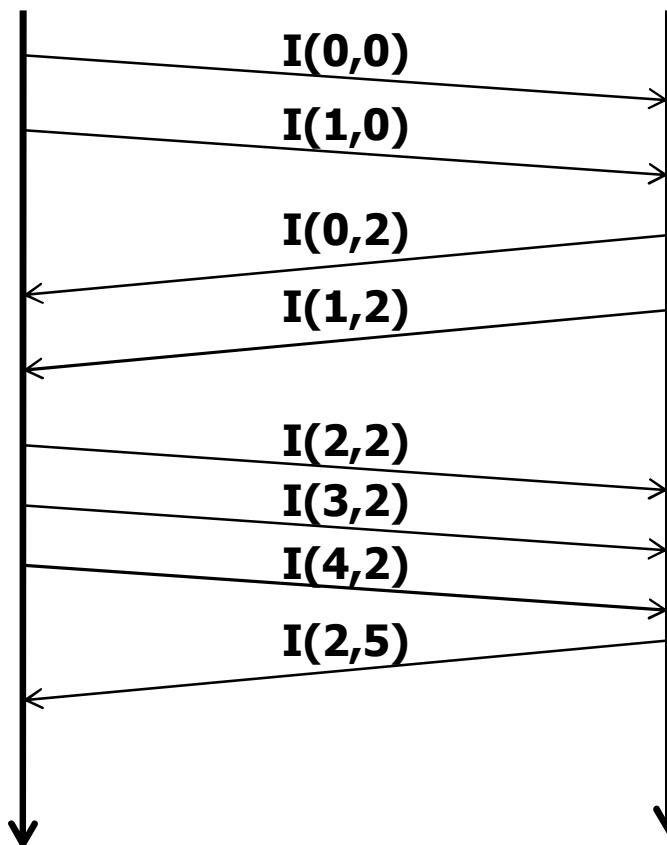


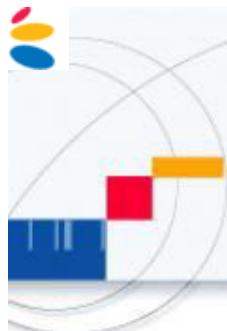
le bit 1 de valeur 0 est spécifique à la trame I



Le champs commande

La trame I (Information) :





Le champs commande

Les trames S (Supervision) :

Permettent l'acquittement et l'indication de l'état de disponibilité des stations (aptitude ou non à recevoir de nouvelles trames).

Servent au contrôle d'erreur et au contrôle de flux. Contiennent un numéro Nr

Les quatre trames de supervision sont :

la trame RR (Receiver Ready) indique que l'équipement est prêt à recevoir de nouvelles trames d'information. Le numéro de séquence Nr indique le numéro de la prochaine trame attendue. Il indique donc que toutes les trames d'information de numéro Ns strictement inférieur à Nr ont été bien reçues.

la trame RNR (Receiver Not Ready) indique que l'équipement n'est pas en mesure de recevoir de nouvelles trames d'information. Le numéro Nr a la même signification que pour RR.

la trame REJ (Reject) indique que l'équipement demande l'arrêt immédiat des émissions en cours de trame d'information et la reprise de la transmission. Le numéro de séquence Nr indique où reprendre la transmission.

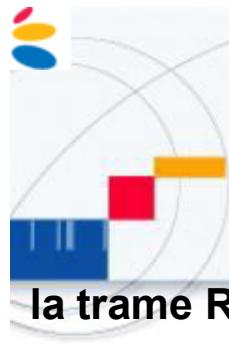
La trame SREJ (Selective Reject) : protection contre les erreurs confirme la réception des trames de données de $n^o < N(R)$

demande la retransmission de la trame de $n^o = N(R)$
non-utilisée par LAP-B

8	7	6	5	4	3	2	1
N(R)	P/F	S	S	0	1		
N(R)	P/F	0	0	0	1		
N(R)	P/F	0	1	0	1		
N(R)	P/F	1	0	0	1		
N(R)	P/F	1	1	0	1		

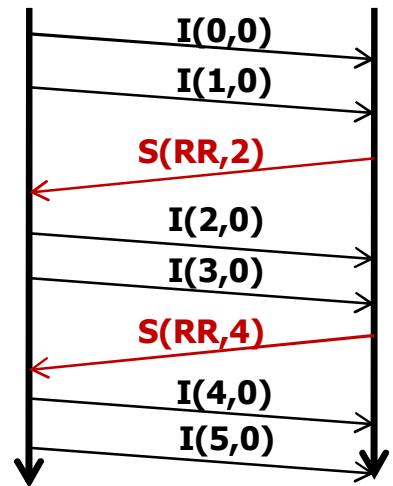
format général aux trames S
RR : Receiver Ready
RNR : Receiver Not Ready
REJ : Reject
SREJ : Selective Reject

les bits 1 et 2 sont spécifiques des trames de supervision S
les bits 3 et 4 définissent le type de trame de supervision

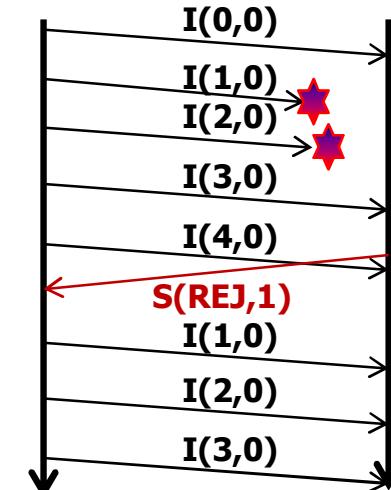


Le champs commande

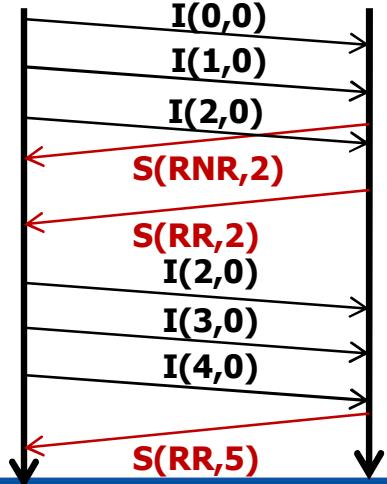
la trame RR (Receiver Ready)



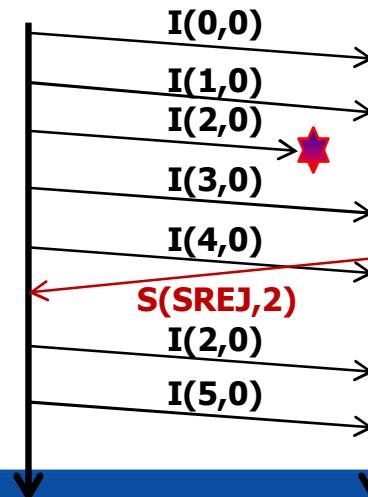
la trame REJ (Reject)

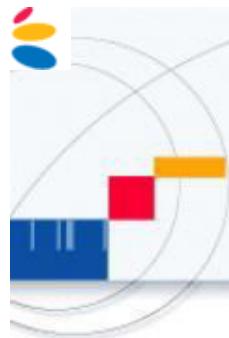


la trame RNR (Receiver Not Ready)



La trame SREJ (Selective Reject)





Le champs commande

Les trames U (Les trames non numérotées) (Unnumbered) :

Les trames U sont utilisées pour effectuer des fonctions supplémentaires de commande de la liaison :

SABM (Set Asynchronous Balanced Mode) permet d'initialiser le fonctionnement en mode équilibré ;

DISC (DISConnect) permet de rompre logiquement la liaison entre deux stations ;

UA (Unnumbered Acknowledgement) permet d'acquitter les commandes SABM ou DISC ;

FRMR (FRaMe Reject) permet de rejeter une commande invalide (correcte du point de vue de la détection des erreurs mais incohérente par rapport à l'état du dialogue) ;

DM (Disconnect Mode) permet d'indiquer l'état de déconnexion d'une station. Il est utilisé, en particulier, pour répondre négativement à une initialisation SABM.

8	7	6	5	4	3	2	1
U	U	U	P/F	U	U	1	1
0	0	1	P/F	1	1	1	1
0	1	0	P/F	0	0	1	1
0	1	1	P/F	0	0	1	1
1	0	0	P/F	0	1	1	1
0	0	0	P/F	1	1	1	1

format général pour trames U

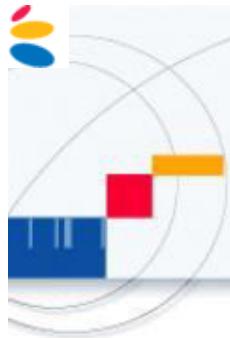
SABM : Set Asynchronous Balanced Mode

DISC : DISConnect

UA : Unnumbered Acknowledgment

FRMR : FRaMe Reject

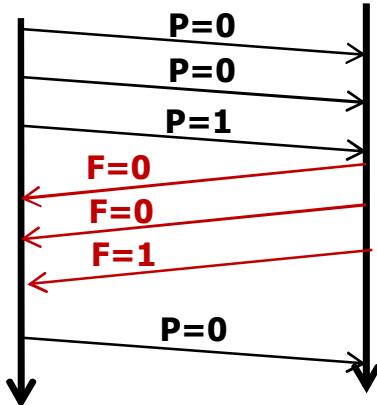
DM : Disconnect Mode

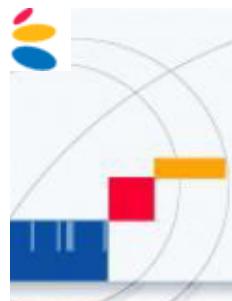


Le champs commande

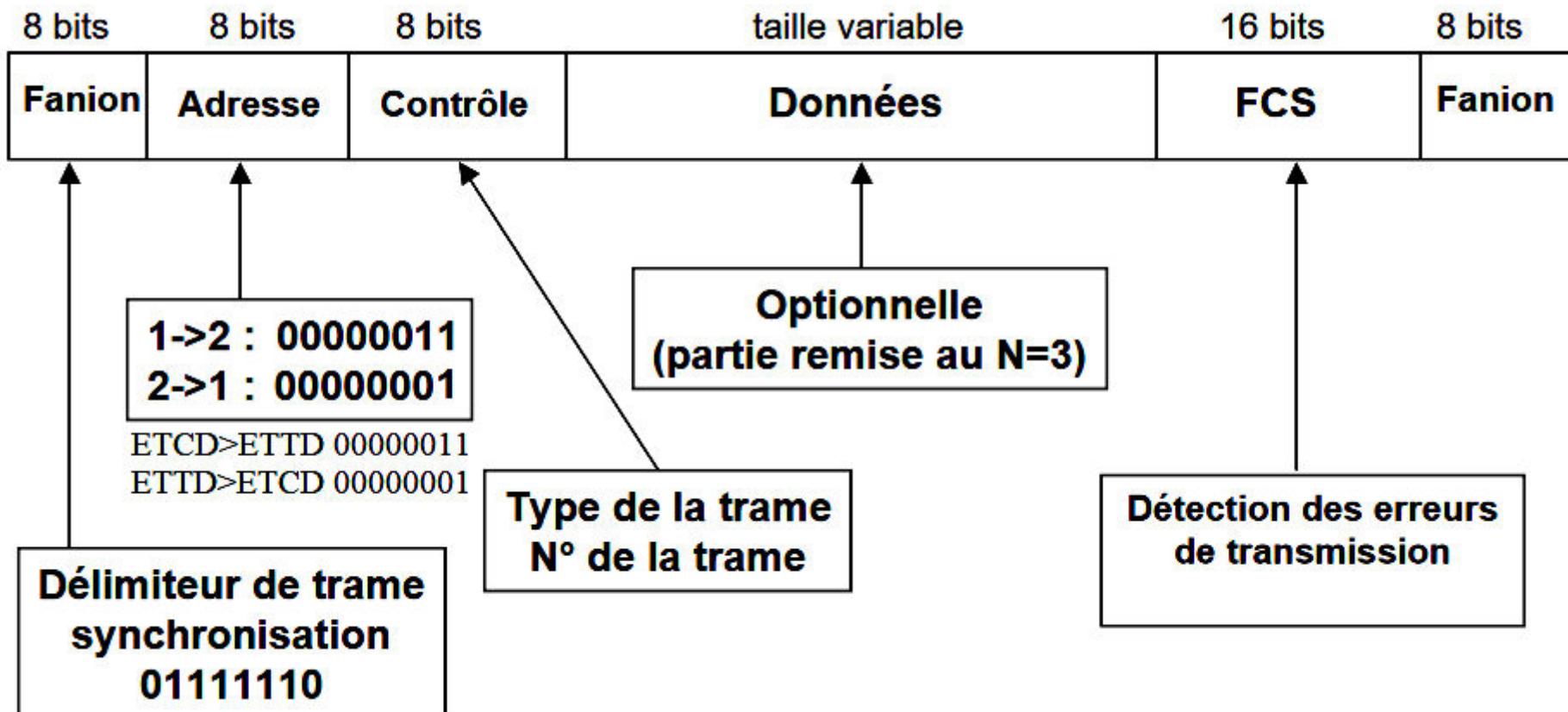
Le bit P/F - bit de contrôle

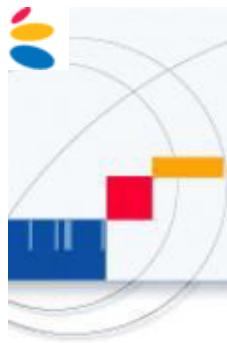
- Il est appelé P dans une trame de commande, F dans une trame de réponse
- P (Poll) = 1 demande de réponse explicite ou non émise par une station primaire
- F (Final) = 1 indication de réponse explicite émise par une station secondaire, suite à une demande explicite émise par un primaire
- P=1 - sollicite une réponse explicite du secondaire
- Réponse à P=1 par F=1 - le secondaire répond par un acquittement
- Une station qui reçoit une trame de commande avec le bit P/F=1 doit répondre avec P/F=1





Trame HDLC





Structure d'un dialogue HDLC

Connexion et libération:

Une connexion bipoint se déroule en trois étapes: **initialisation** du mode, **échanges** de trames d'information avec contrôle d'erreur et de flux, et enfin **fermeture** de la connexion.

SNRM : Set normal response mode

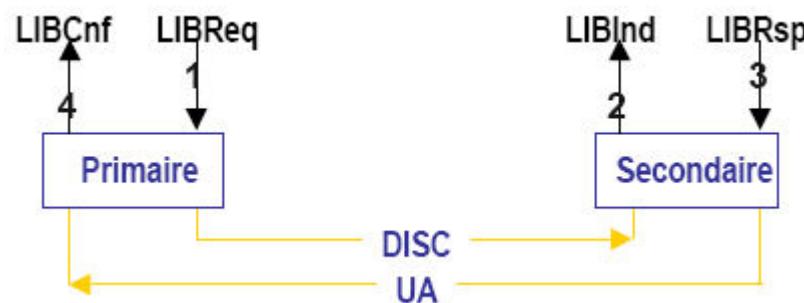
SABM : Set Asynchronous Balanced Mode

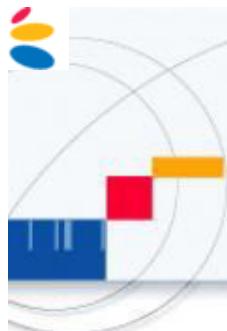
UA : Unnumbered Acknowledgement



DISC : DISConnect

UA : Unnumbered Acknowledgement





Structure d'un dialogue HDLC

Règles de reprise pour le mode LAP_B

Débordement (contrôle de flux)

- la station qui ne peut plus recevoir de **trames I** émet un **RNR** dont le **Nr** indique la **première trame non acceptée**
- elle émet une trame **RR** quand elle est de nouveau prête à recevoir de nouvelles trames

Erreur de transmission

- toute trame dont l'analyse du **FCS** indique une erreur de transmission est ignorée aucune autre action n'est reprise

Erreur de numéro de séquence **Ns**

- la trame dont le **Ns** n'est pas celui attendu est ignorée ainsi que les suivantes
- la station recevant cette trame émet une trame **REJ** dont le numéro **Nr** indique la trame attendue

Deux temporisateurs

- temporisateur de retransmission (**T1**) initialisé par l'émetteur à chaque trame émise
- temporisateur d'acquittement (**T2**) correspondant au délai maximum au bout duquel le récepteur doit, s'il n'a pas de données à émettre, envoyer un acquittement



Structure d'un dialogue HDLC

Règles de reprise pour le mode LAP_B

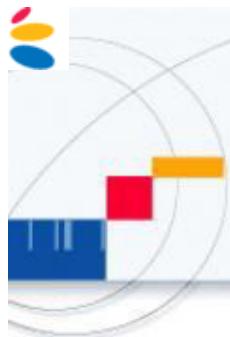
Reprise sur time-out

- la station qui ne reçoit aucun acquittement des trames d'information émises au bout d'un temps **T1** reprend l'émission de la première trame non acquittée en positionnant le bit **P/F à 1**
- la station peut poursuivre normalement l'émission des autres **trames I** dès la réception d'un acquittement portant le bit **P/F à 1**

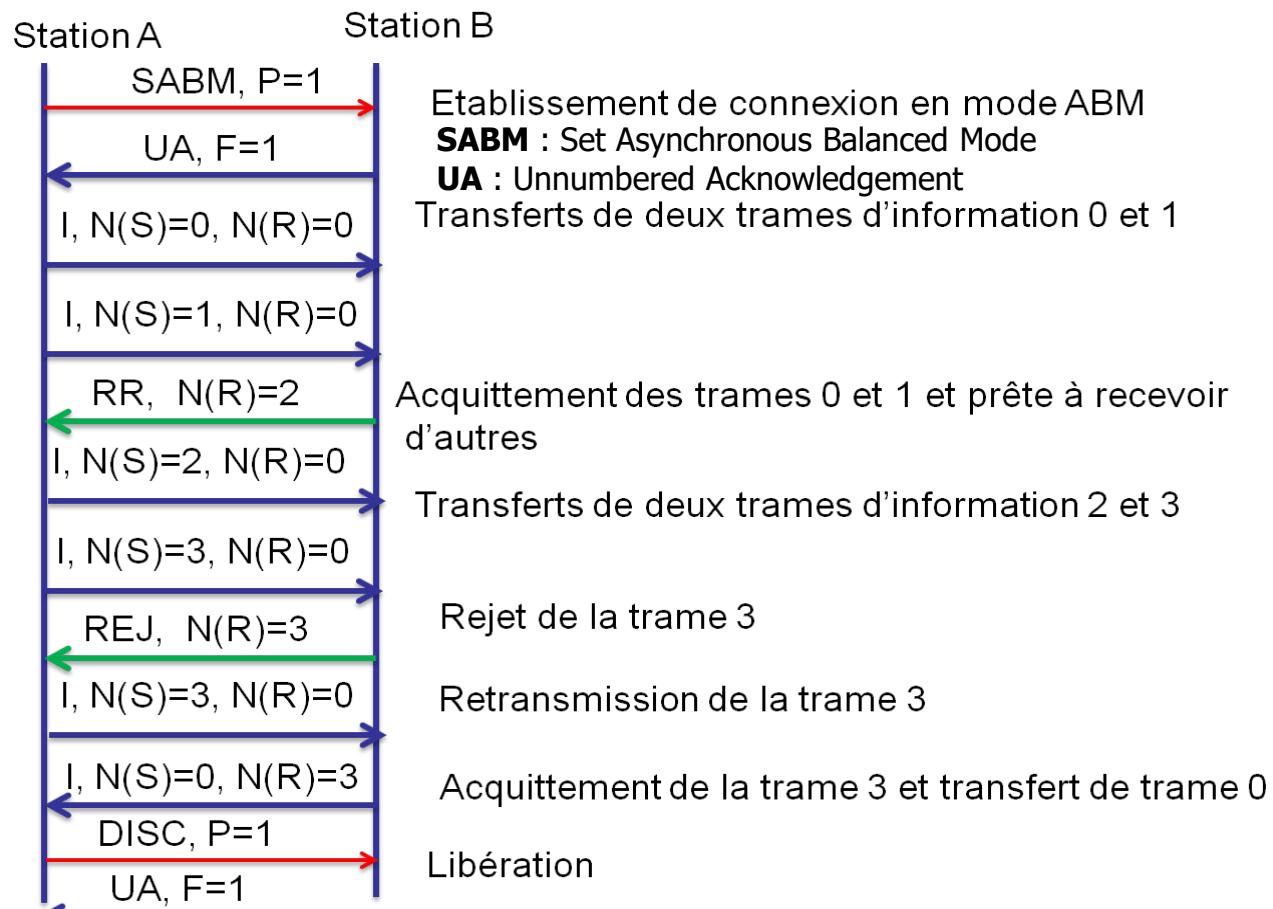
Des variables d'état **V(s)**, **V(r)** et **DN(r)**

Dans chaque station, la mise en œuvre des procédures comprend la gestion du contrôle des temporiseurs et des variables d'état appelées **V(s)**, **V(r)** et **DN(r)** définies par:

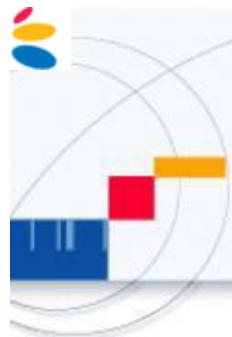
- **V(S)** : numéro (modulo 8) de la prochaine trame d'information que la station est prête à émettre.
- **V(R)** : numéro (modulo 8) de la prochaine trame d'information que la station est prête à recevoir.
- **DN(R)**: "numéro plus un" (modulo 8) de la dernière trame d'information que la station avait émise et qui a été acquittée par la station distante.



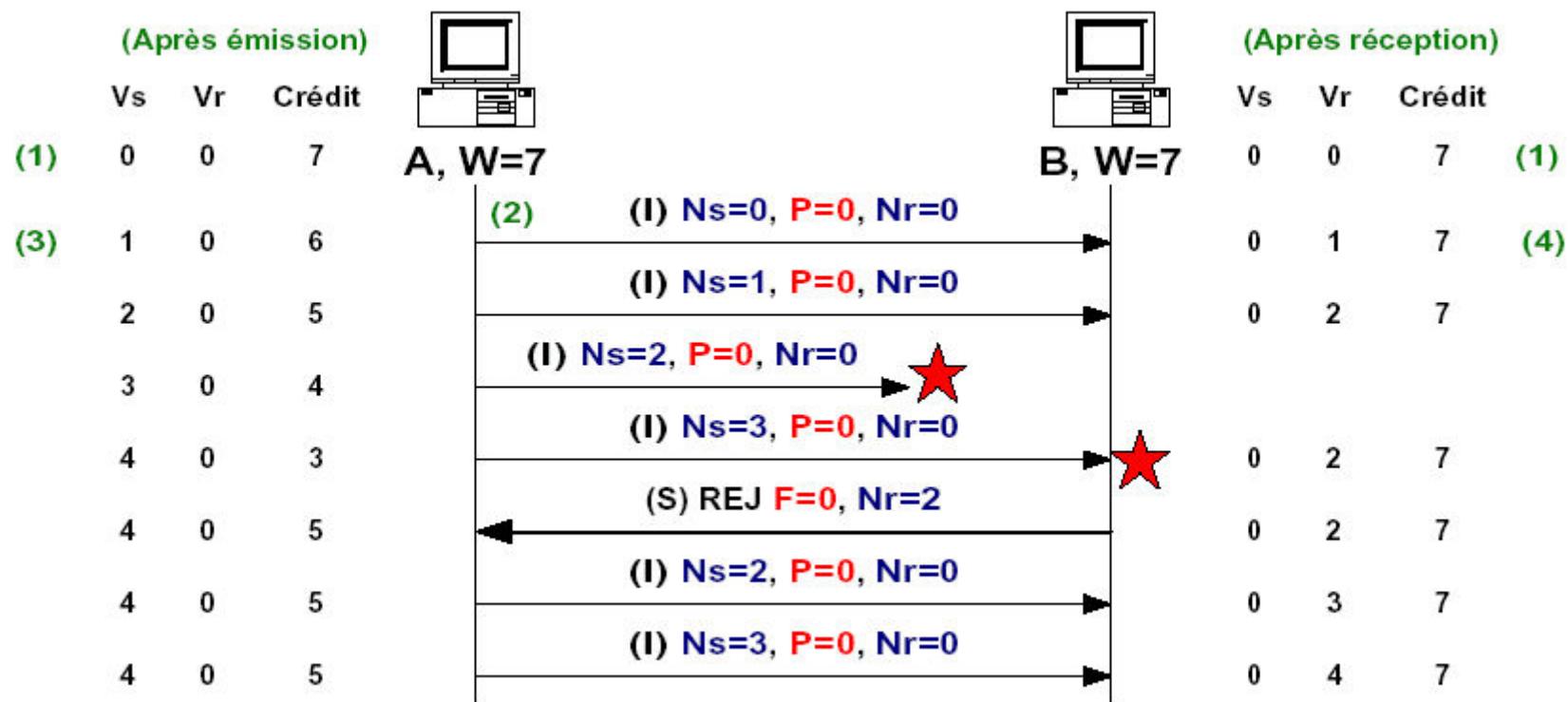
Exemple d'échange de données



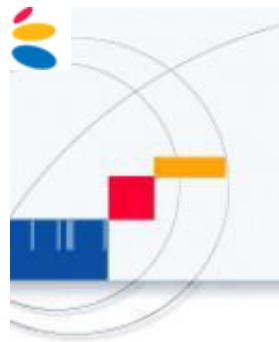
Chronogramme d'une transmission



Exemple d'échange de données



Chronogramme d'une transmission



Exemple d'échange de données

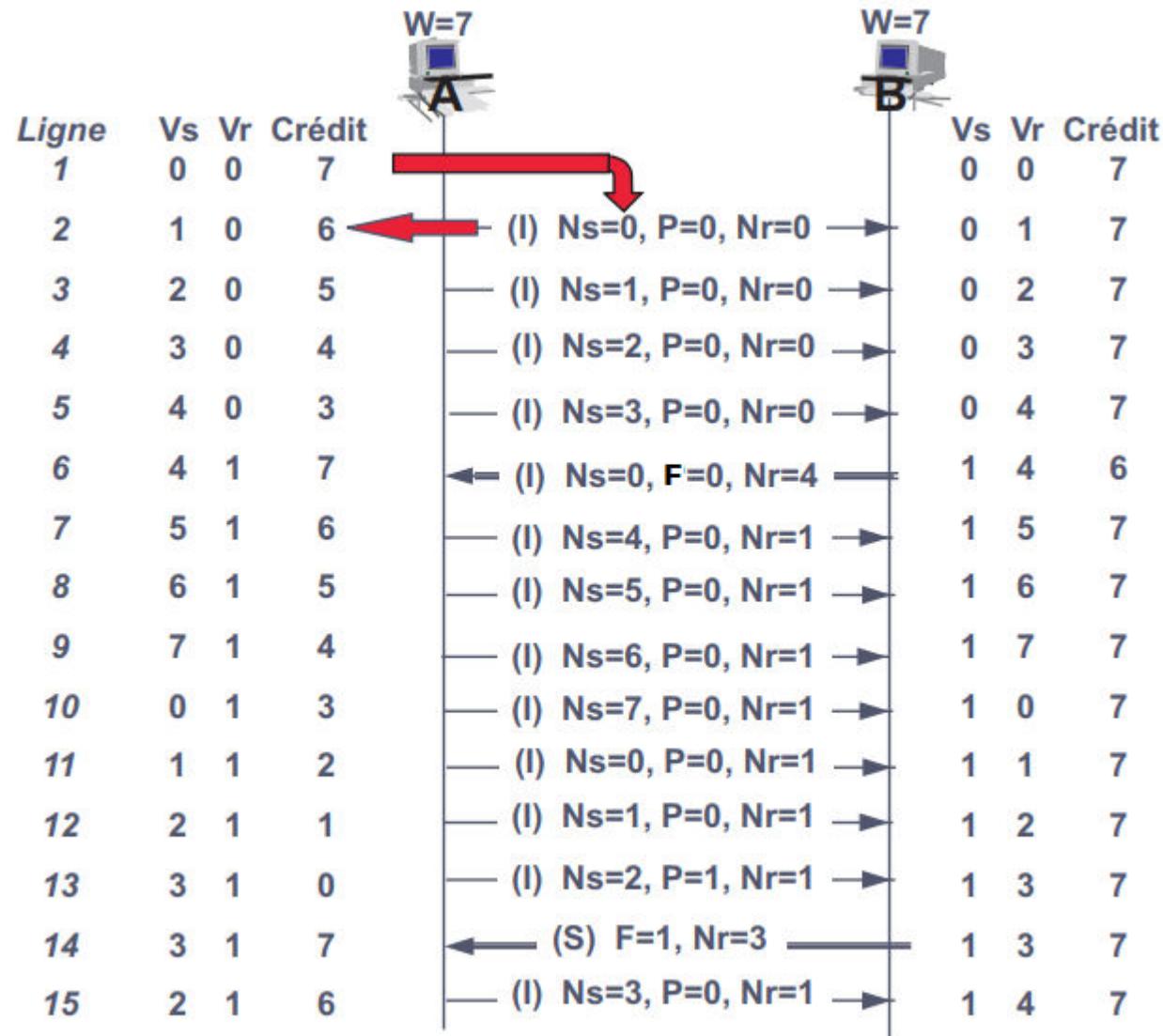
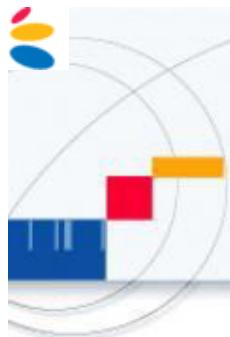


Figure L'échange de données et la gestion de la fenêtre.



Exemple d'échange de données

