

## Chapitre 5 : Liaison de données

/udd/bcousin/Pages-web-Armor/Enseignement/Reseaux-generalites/Cours/5.recover.fm - 5 juin 2001

### Plan

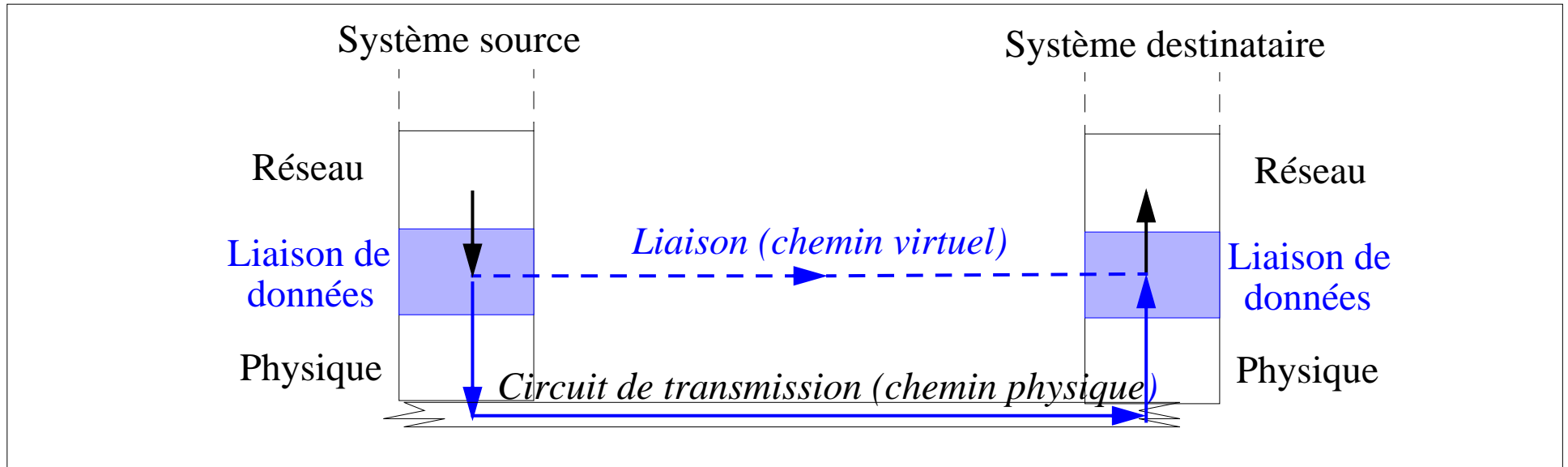
- 1. Introduction p128
- 2. La trame p130
- 3. Mécanismes généraux p134
- 4. Le protocole HDLC (High Level Data Link Control) p142
- 5. Conclusion p161

### Bibliographie

- High-level Data Link Control : ISO 3309, 4335, 7448, 7776, 7809, 8471
- G. Pujolle, Les réseaux, Eyrolles, 1995. Chapitre 5.
- H. Nussbaumer, Téléinformatique, Presses polytechniques romandes, 1987. Tome 1, chapitre 4.
- A. Tanenbaum, Réseaux, InterEditions, 1997. Chapitre 3.

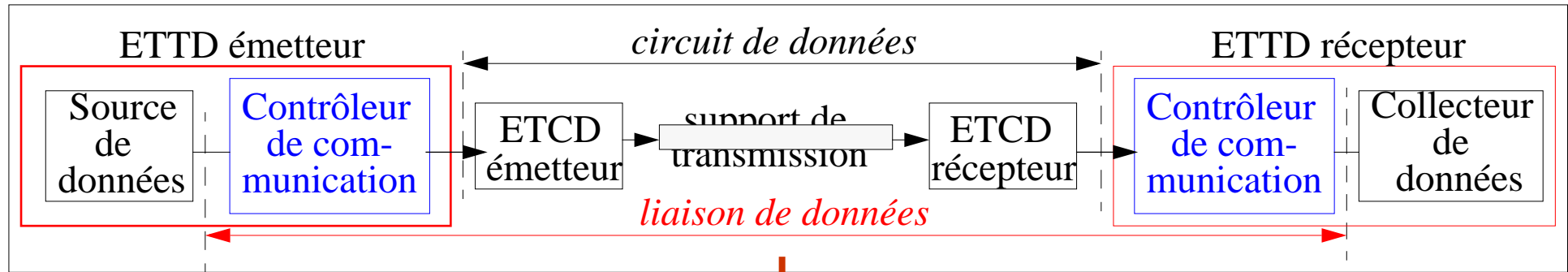
## 1. Introduction

Transmettre des données entre les entités homologues d'un système source et d'un système destinataire *adjacents* de manière *fiable* et *efficace*.



Utiliser un circuit de transmission de données et le transformer en une liaison de données efficace qui paraît exempte d'erreurs de transmission

La couche Liaison de données a pour rôle d'établir une liaison de données entre systèmes adjacents primitivement reliés par un circuit de données.



Nécessité de délimiter les unités de données, de lutter contre les erreurs et la duplication, d'assurer le contrôle de flux et le séquençement, et parfois d'identifier les destinations (et origines) de l'information

Généralement au sein de chaque système (ETTD), les fonctions de la couche Liaison de données sont réalisées par une carte spécifique appelée contrôleur de communication.

- Par exemple : carte HDLC, carte Ethernet, etc.

## 2. La trame

C'est l'unité de données du protocole de niveau Liaison de données (**L-PDU**).

Adaptation (de la longueur des données) aux caractéristiques :

- de l'application
  - de la transmission
  - de protection contre les erreurs
  - du stockage
- ==> longueur variable mais bornée, ou parfois fixe

Identification

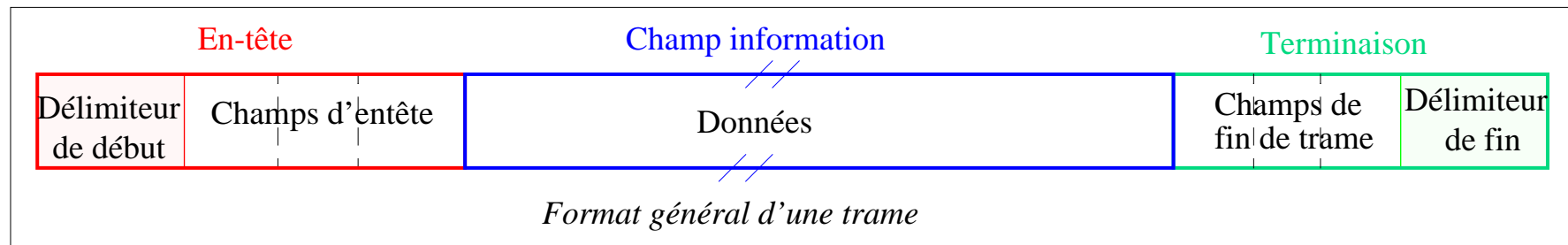
Structuration :

- les champs de contrôle (en nombre fixe ou variable)
- le champ de données

## 2.1. Format général d'une trame

Une **trame** :

- Une suite de bits (d'une longueur variable mais bornée)
- Le début et la fin de trame sont souvent identifiés par des délimiteurs
- Composée d'un certain nombre de champs ayant chacun une signification précise.
- On distingue souvent 3 ensembles de champs : l'entête ("header"), le champ de données, la terminaison ("trailer")



## 2.2. Remarques

Suivant le type de protocoles, une trame peut :

(i) être de taille fixe ou de taille variable (jusqu'à une certaine taille maximum)

- Exemple :

- . **variable** = trame d'HDLC,
- . **fixe** = cellule d'ATM (53 octets)

(ii) ne pas avoir de fin de trame explicite.

- Exemple :

- . SD + ED ("Starting/ending delimiter") = trame d'HDLC ou Token Ring,
- . SD uniquement : trame Ethernet,  
en fait la fin de la trame est détectée par la fin de la porteuse du signal.

Suivant le rôle qui lui est attribué :

- une trame peut ne pas comporter de champ information.
  - . Par exemple : un simple acquittement

### 2.3. Délimitation des trames

La couche Physique (grâce à la technique de codage utilisée) permet généralement au récepteur de reconstituer (à l'aide du signal reçu) l'horloge et donc d'assurer la **synchronisation au niveau du bit** et ainsi de reconstruire la suite binaire envoyée.

La couche Liaison de données à partir de cette suite binaire va reconstituer la trame envoyée en assurant la **synchronisation au niveau de la trame**.

#### Délimiteurs de trame :

Chaque trame commence par un délimiteur de début et se termine par un délimiteur de fin. Un délimiteur peut être :

- soit une **séquence particulière de caractères**,
  - . Exemple : les séquences **(DLE) STX** et **(DLE) ETX** du protocole BSC d'IBM
- soit une **suite particulière de bits**,
  - . Exemple : le fanion **01111110** du protocole HDLC.
- soit un **codage particulier** :
  - . Des séquences, non-utilisées pour coder les éléments binaires, servent à délimiter les trames.
  - . Exemple : les symboles **J** et **K** du protocole Token Ring.

### Utilisation d'un compteur de caractères :

- L'en-tête de trame contient un champ indiquant la longueur de la trame.
- Inapplicable puisque l'on ne sait pas où commence la trame !
- Exemple : codage d'une chaîne de caractères = "string" en Turbo-Pascal.



### 3. Mécanismes généraux

#### 3.1. Techniques de contrôle

“S’assurer que le récepteur a reçu *correctement, en un seul exemplaire et dans l’ordre, les trames émises*”.

Deux phénomènes viennent perturber la transmission :

- la **corruption de trames** : transformation de la suite binaire transmise
- la **perte de trames** due à la congestion, à une panne ou la non-reconnaissance de la trame
  - . ex : corruption du délimiteur
- une corruption se traduit par une perte lorsque le récepteur détruit la trame corrompue

Les solutions :

- Mécanisme de **détection** des erreurs (cf. chapitre précédant)
- Mécanisme d’**acquiescement** positif ou négatif (explicite ou implicite)
- Techniques de **mémorisation** des trames et de correction par **retransmission**
- Utilisation de **temporisateurs**
- **Identification** des trames

### 3.2. Identification des trames

Une numérotation permet d'identifier les trames et leur acquittement

L'unité d'identification varie en fonction des protocoles :

- la trame (par exemple HDLC)
- l'octet (par exemple TCP)
- plus l'unité est petite plus l'identification est précise mais plus c'est coûteux (taille du champ)
  - . ex : HDLC = 3 bits (ou 7 bits), TCP = 16 bits

La numérotation se fait modulo  $N$  ( $= 2^{\text{la\_largeur\_du\_champ\_de\_numérotation}}$ )

- la champ est de taille bornée (car la trame est bornée !)

Au cours d'une même connexion le même numéro va être réutilisé pour identifier plusieurs trames différentes

- il faut lever les ambiguïtés :
  - .  $W < N$
  - . problème similaire à la gestion du tampon circulaire et ses pointeurs lorsque  $P_{\text{debut}} = P_{\text{fin}} \implies$  tampon plein ou vide ?

L'identification permet de détecter :

- la perte de trames :
  - . demande explicite de retransmission ou par temporisateur
- la duplication de trames :
  - . destruction du duplicata
- l'ordonnement des trames :
  - . soit réordonnement si la mémorisation est autorisée
  - . soit destruction des trames n'arrivant pas dans le bon ordre (+ facile).

### 3.3. Protection des trames

#### 1- Vérification par le récepteur de données

Vérification du format des trames :

- longueur, valeurs prédéfinies de certains champs

Détection de la corruption des trames :

- champ de contrôle d'erreur

#### 2- Information de l'émetteur de données

- Soit implicitement par temporisateur
  - . armé à chaque envoi de trame,
  - . désarmé lors de la réception d'un acquittement
- Soit explicitement par acquittement : "Nack"
  - . le **rejet total** : retransmission de toutes les trames à partir de celle spécifiée
  - . le **rejet sélectif** : retransmission de la trame spécifiée

### 3- Retransmission de la trame (perdue ou détruite) par l'émetteur

La liaison de données est maintenue active (et surveillée) par la transmission périodique de trames (d'acquittement) même si aucune donnée n'est échangée.

- lutte contre la perte d'acquittement (par répétition !)
- détecte la rupture du circuit de données.

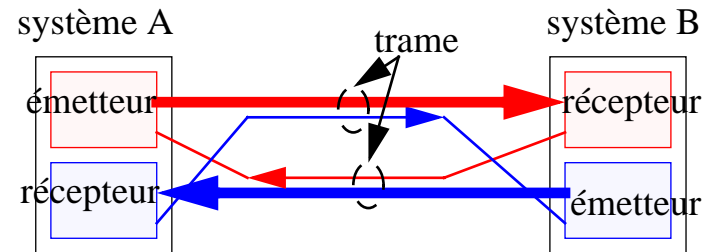
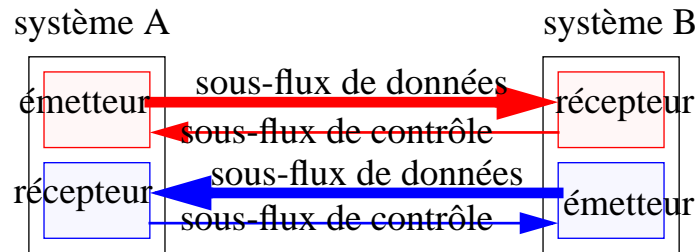
### 3.4. “Piggybacking”

Au sein d’un flot de données unidirectionnel, on peut discerner deux sous-flux : le sous-flux de données proprement dit (de l’émetteur de données au récepteur) et le sous-flux de contrôle (du récepteur vers l’émetteur de données).

Lorsque le flot de données est bidirectionnel, deux sens de transmission des données existent. Les deux systèmes d’extrémité fonctionnent à la fois comme émetteur de données et comme récepteur de données.

Les trames transportant le sous-flux de données d’un sens de transmission peuvent être combinées avec les trames du sous-flux de contrôle de l’autre sens.

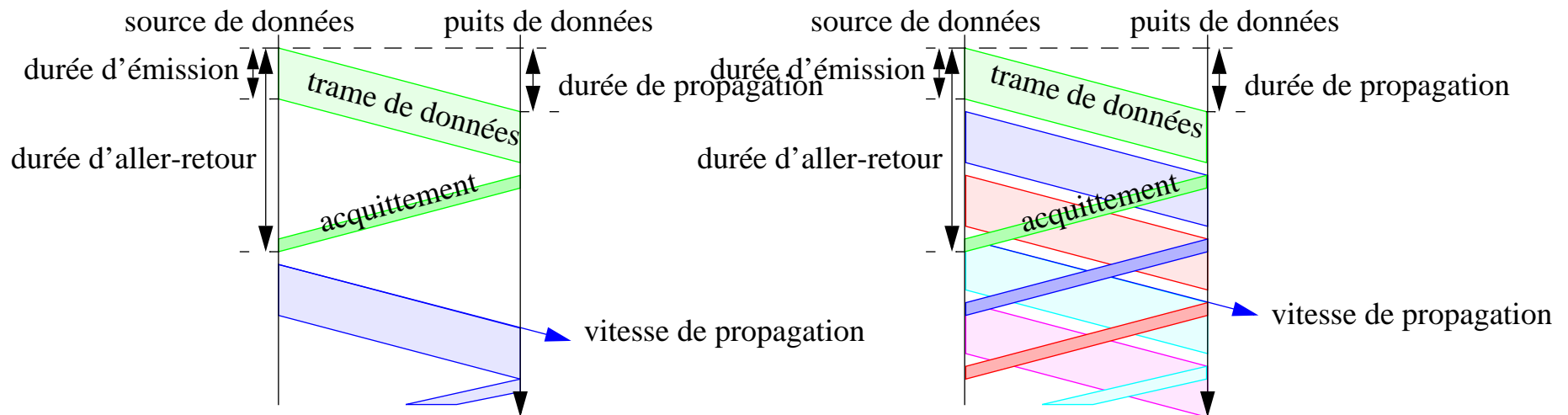
- Par exemple : une même trame peut se comporter à la fois comme une trame de données (elle possède un champ d’information) et un acquittement (elle possède un champ du même nom).



### 3.5. Fenêtre coulissante (“sliding window”)

Les protocoles simples précédents (bit alterné, “send and wait”, “stop and go”) ont comme principal inconvénient de n’autoriser que la transmission d’une seule trame à la fois.

La liaison de données est alors inoccupée la plupart du temps. De même, l’émetteur (resp. le récepteur) passe son temps à attendre l’acquittement du récepteur (resp. la trame de données de l’émetteur)



On autorise l’émission (resp. la réception) **de plusieurs trames d’information consécutives** sans attendre l’acquittement de la première (resp. avant d’envoyer l’acquittement).

On remarque que la source et le puits émettent et reçoivent simultanément

- la liaison est utilisée de manière bidirectionnelle

Le nombre maximum de trames consécutives que l'on peut ainsi émettre (resp. recevoir) est défini par la **largeur de la fenêtre d'anticipation** d'émission (resp. de réception).

- Dans l'exemple : la largeur  $W \geq 4$

Pour que la liaison de données soit totalement utilisée il faut que :

- $W * L \geq T_{a/r} * D$ 
  - . L étant la longueur moyenne d'une trame,  $T_{a/r}$  la durée d'aller/retour et D le débit nominal de la liaison.
  - .  $T_{a/r} * D$  représente la capacité de la liaison

La largeur de fenêtre (W) peut être :

- **fixe**
  - . par exemple : HDLC ou X25.3
- **variable**
  - . par exemple : TP
  - . dans ce cas sa valeur instantanée est appelée crédit

### 3.6. Contrôle de flux

Pour éviter la congestion du récepteur :

- c'est-à-dire le débordement de son espace de stockage, ce qui entraînerait la destruction de trames devant être reçues
- la largeur de la fenêtre doit correspondre à la capacité de stockage du récepteur

Dans le cas d'une fenêtre de largeur fixe :

- le contrôle s'effectue sur la totalité de la fenêtre,
- la fenêtre est ouverte ou fermée.
- exemple : XON/XOFF ou RR/RNR de HDLC.

Dans le cas d'une fenêtre de largeur variable :

- la largeur (le crédit) est adaptée à la capacité de stockage du récepteur,
- les PDU doivent comporter alors un champ supplémentaire pour acheminer le crédit du récepteur de données à l'émetteur de données.
- exemple : champ crédit du protocole Transport.



## 4. Le protocole HDLC

HDLC (“High Level Data Link Control”)

- offre un service de transfert de données fiable et efficace entre deux **systemes adjacents**.
- protocole utilisant le mode connecté

High-level Data Link Control :

- ISO 3309 : HDLC frame structure
- ISO 4335 : HDLC : elements of procedure,
- ISO 7448 : MultiLink procedure (MLP),
- ISO 7776 : LAP-B compatible link control procedure,
- ISO 7809 : Consolidated classes of procedures,
- ISO 8471 : HDLC balanced, link address information

Utilisé comme protocole de la couche Liaison de données dans les normes X.25 (du CCITT) en usage dans les réseaux publics de transmission numérique de données (TRANSPAC, par exemple).

Note : le protocole PPP d’Internet réutilise HDLC

## 4.1. Historique

1960 : BSC (“Binary synchronous communication”) - IBM

- tout premier protocole synchrone :
  - . l’horloge du récepteur est maintenue synchronisée même s’il n’y a pas de transmission de données
  - . transmission plus rapide (sans resynchronisation)
  - . nécessite un contrôleur de communication spécialisé
- l’unité de transmission est le caractère (code ASCII (7 bits) ou EBCDIC (8 bits))
  - . par abus : protocole “orienté” caractère

70 : SDLC (“Synchronous data link control”) - IBM

- l’unité de transmission est la trame
- normalisé par l’ANSI (“American national standard institute”) sous le nom ADCCP (“Advanced data communication - control procedure”)

76 : HDLC (“High data link control”)

- protocole basé sur l’élément binaire (“orienté” bit)
- ISO 3309 : HDLC frame structure
- ISO 4335 : HDLC : elements of procedure

80 : adapté pour l'accès au réseau numérique de données

- LAP-B (“Link access procedure-balanced”) :
  - . rôles équilibrés (symétriques) entre les deux systèmes adjacents
- normalisé : CCITT X25.2 et ISO 7776

85 : adapté aux réseaux locaux

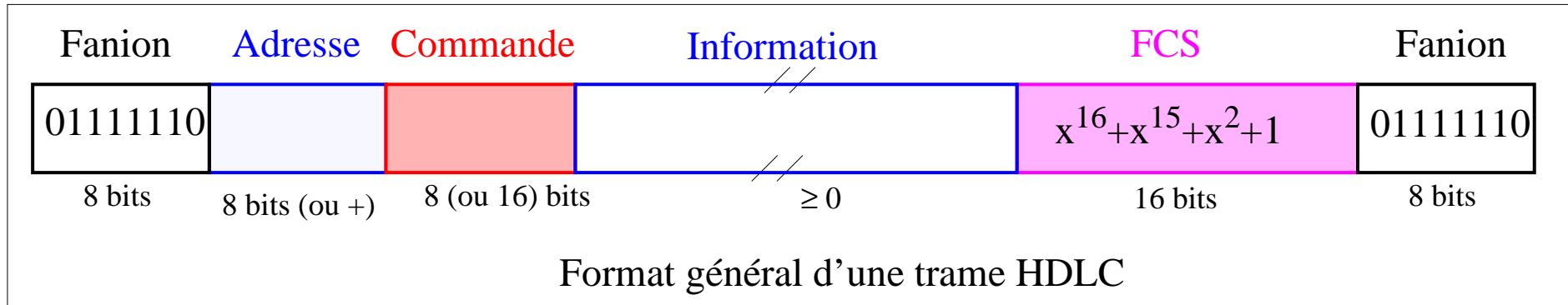
- protocole de la sous-couche d'homogénéisation LLC (“Logical link control”)
- apparition d'un mode de transmission non connecté (LLC classe 1)
- normalisé : IEEE 802.2 et ISO 8802/2

Autres adaptations :

- LAP-X : Téléx - CCITT T71
- LAP-D : canal D du RNIS (Numéris) - CCITT Q921 ou I441
- LAP-F : relai des trames (“frame relay”) - UIT-T Q931 ou I451
- LAPDm : liaison radio GSM

## 4.2. Format général d'une trame HDLC

Trame : L-PDU ("Link protocol data unit")



### 4.3. Le fanion (“flag”)

Délimite la trame : dedans/dehors

- la trame est de longueur variable puisque le champ de données est de longueur variable

Format :

- octet formé de 6 bits consécutifs à 1, préfixés et suffixés par un bit à 0.

Unicité :

- Comment assurer l’unicité de la configuration binaire du fanion à l’intérieur de la trame puisque le champ de données de la trame peut comporter n’importe quel octet (le transport des données est *transparent*) ?

Transcodage :

- la trame (sauf les fanions) est transcodée lors de la transmission
  - toute suite de 5 bits consécutifs à 1 est transcodée en une suite de cinq bits à 1 et d’un bit à 0.
    - . augmentation de la longueur de la trame (6/5 au maximum)
    - . la longueur de la trame transcodée n’est plus un multiple d’octets !
- Rencodage : opération inverse au récepteur

Notes :

- Abandon de la transmission d'une trame :
  - . au moins sept bits consécutifs à 1
- Remplissage de l'espace intertrame :
  - . par des fanions ou des éléments de 7 à 14 bits consécutifs à 1.
- Les trames successives peuvent n'être séparées que par un seul fanion
- Une suite de plus de 15 bits consécutifs à 1 est interprétée comme une liaison hors service
  - . dépend du codage de transmission :
  - . codage bipolaire (avec 1 = état neutre).

Exemple :

- suite binaire originale : 0110001011111110111110
- suite transcodée: 0110001011111**1**11101111**1**0

#### 4.4. Les adresses

Seulement 2 adresses utilisées normalement (LAP-B)

Identification de la station à laquelle la commande est destinée

- commande de l'ETCD vers l'ETTD : @A = 11000000
- commande de l'ETTD vers l'ETCD : @B = 10000000

Identification de la station qui émet la réponse à la commande

- réponse de l'ETCD vers l'ETTD : @B
- réponse de l'ETTD vers l'ETCD : @A

Adresse fictive : 00000000

Adresse de diffusion totale ("broadcast"): 11111111

Champ d'adresse extensible :

- suite quelconque d'octets dont le premier bit est = 0, sauf pour le dernier octet de la suite
- exemple : 0001 0010 0100 1000 1111 0110 = 0010 0101 0010 0011 1011 0

## 4.5. Les différents types de trames

Trois types de trames :

les trames d'**information** (**I** *Information*)

les trames de **supervision** (**S** *Supervisory*)

les trames **non numérotées** (**U** *Unnumbered*)

Elles se distinguent notamment par leur champ Commande :

Types de trame	Champ Commande							
<b>I</b>	0	<b>N(S)</b>			P/F	<b>N(R)</b>		
<b>S</b>	1	0	<b>Type</b>		P/F	<b>N(R)</b>		
<b>U</b>	1	1	<b>M</b>	<b>M</b>	P/F	<b>M</b>	<b>M</b>	<b>M</b>

Exemples de champ Commande :

- I(3, 5, P) : 00**111**101

- REJ(4, F) : 0**101**1100



Deux formats du champ Commande existent :

- le format normal (8 bits)
- le format étendu pour les trames numérotées (16 bits) : négocié lors de l'établissement de la connexion (SABME) pour avoir un champ de commande plus grand et ainsi effectuer la numérotation modulo 128.

## 4.6. Les trames d'information

Acheminent les données. Utilisent le “piggybacking”.

- commande ou réponse

**N(S)** :

- numéro de la trame d'information (modulo 8 ou 128)

**N(R)** :

- numéro de la prochaine trame d'information attendue (modulo 8 ou 128)
- acquitte **toutes** les trames de numéros strictement inférieurs à N(R)
- la perte d'un acquittement peut ainsi être compensée par le prochain acquittement
- exemple : cf paragraphe 4.12

Le bit **P/F** (“Poll/Final”) :

- Dans le mode équilibré du protocole : **LAP-B**
  - . dans une commande : demande de réponse immédiate
  - . dans une réponse : réponse à la demande de réponse immédiate
- Dans le mode normal (historique) du protocole : **LAP**
  - . code l'alternance du droit d'émission (maître/esclave)

## 4.7. Les trames de supervision

4 types de trames de supervision,

- codées dans le sous-champ Type du champ Commande
- commande ou réponse
  - **RR** (“Received & Ready”) - 00 : **acquittement**
    - . confirme la réception des trames de données de  $n^{\circ} < N(R)$
    - . demande la transmission des trames suivantes
  - **RNR** (“Received & Not Ready”) - 10 : **contrôle de flux**
    - . confirme la réception des trames de données de  $n^{\circ} < N(R)$
    - . **interdit** la transmission des trames suivantes
  - **REJ** (“Reject”) - 01 : protection contre les erreurs
    - . confirme la réception des trames de données de  $n^{\circ} < N(R)$
    - . demande la **retransmission des** trames de  $n^{\circ} \geq N(R)$
  - **SREJ** (“Selective Reject”) - 11 : protection contre les erreurs
    - . confirme la réception des trames de données de  $n^{\circ} < N(R)$
    - . demande la **retransmission de la** trame de  $n^{\circ} = N(R)$
    - . **non-utilisée** par LAP-B

## 4.8. Les trames non-numérotées

Toutes les autres trames nécessaires à la gestion de la connexion

- MMMMM : code le type des trames non-numérotées

Trame d'établissement de la connexion (commande) :

- **SABM** ("Set asynchronous balanced mode") [11100] - en format normal
- **SABME** ("Set asynchronous balanced mode extended") - en format étendu

Trame de libération de la connexion (commande) :

- **DISC** ("Disconnection") [00010]

Trame de confirmation (réponse) :

- **UA** ("Unnumbered acknowledgment")[00110]

Trame de récupération des erreurs (réponse) :

- **FRMR** ("Frame reject") [11000]

Trame d'indication de connexion libérée

- **DM** ("Disconnected mode") [10 001]

#### 4.9. Quelques variables des entités du protocole

Chaque entité tient à jour les trois variables suivantes :

- $V(S)$  = numéro de la prochaine trame d'information à émettre,
- $V(R)$  = numéro de la prochaine trame à recevoir,
- $DN(R)$  = numéro du dernier acquittement reçu.

et connaît les constantes suivantes :

- $T1$  = délai de garde au bout duquel une trame non acquittée est réémise.
- $T2$  = délai d'acquittement pendant lequel le récepteur peut retarder l'envoi de l'acquittement d'une trame.
- $N1$  = taille maximum d'une trame.
- $N2$  = nombre maximum de réémissions d'une même trame.
- $W$  = largeur de la fenêtre.
- etc.

## 4.10. Quelques aspects algorithmiques de la procédure (du protocole) HDLC

### 4.10.1 Emission d'une trame

Emission d'une trame **I** :

vérifier que  $V(S) < DN(R) + W$  puis :

- $N(S) = V(S)$  et  $N(R) = V(R)$  ;
- mémoriser la trame;
- incrémenter  $V(S)$  modulo  $N$ ;
- armer le temporisateur (délai de garde **T1**) associé à la trame;
- armer **T2**.

Emission d'une trame **REJ** :

- $N(R) = V(R)$
- armer **T2**.

Emission d'une trame **RR** :

- $N(R) = V(R)$
- armer **T2**.

## 4.10.2 Réception d'une trame

Sur réception d'une trame :

- Si la trame est invalide
  - . alors la trame est ignorée (si FCS incorrect) ou émission d'une trame FRMR (format incorrect).
  - . sinon la trame est décodée

Sur réception d'une trame **I** :

- Si  $N(S) \neq V(R)$
- alors trame non-attendue (déséquencée)
  - . émettre une trame REJ;
- sinon /\*  $N(S) = V(R)$  \*/
  - . armement du temporisateur **T2** (délai d'acquittement) associé à  $N(S)$ ;
  - . incrémentation de  $V(R)$ .
- Si  $DN(R) \leq N(R) < V(S)$  alors
  - . désarmer les temporisateurs T1 des trames de n° compris entre  $DN(R)$  et  $N(R)$ ;
  - .  $DNR(R) = N(R)$ ;

### Sur réception d'une trame **RR**

- Si  $DN(R) = < N(R) < V(S)$  alors
  - . désarmer les temporisateurs T1 des trames de n° compris entre DN(R) et N(R);
  - .  $DNR(R) = N(R)$ .

### Sur réception d'une trame **REJ**

- Si  $DN(R) = < N(R) < V(S)$  alors
  - . désarmer les temporisateurs T1 des trames de n° compris entre DN(R) et N(R);
  - .  $DNR(R) = N(R)$ ;
  - . émettre les trames de numéros compris entre N(R) et V(S) ( $< N2$ ).

#### 4.10.3 Expiration d'un délai

##### A l'expiration du délai **T1** associé à une trame

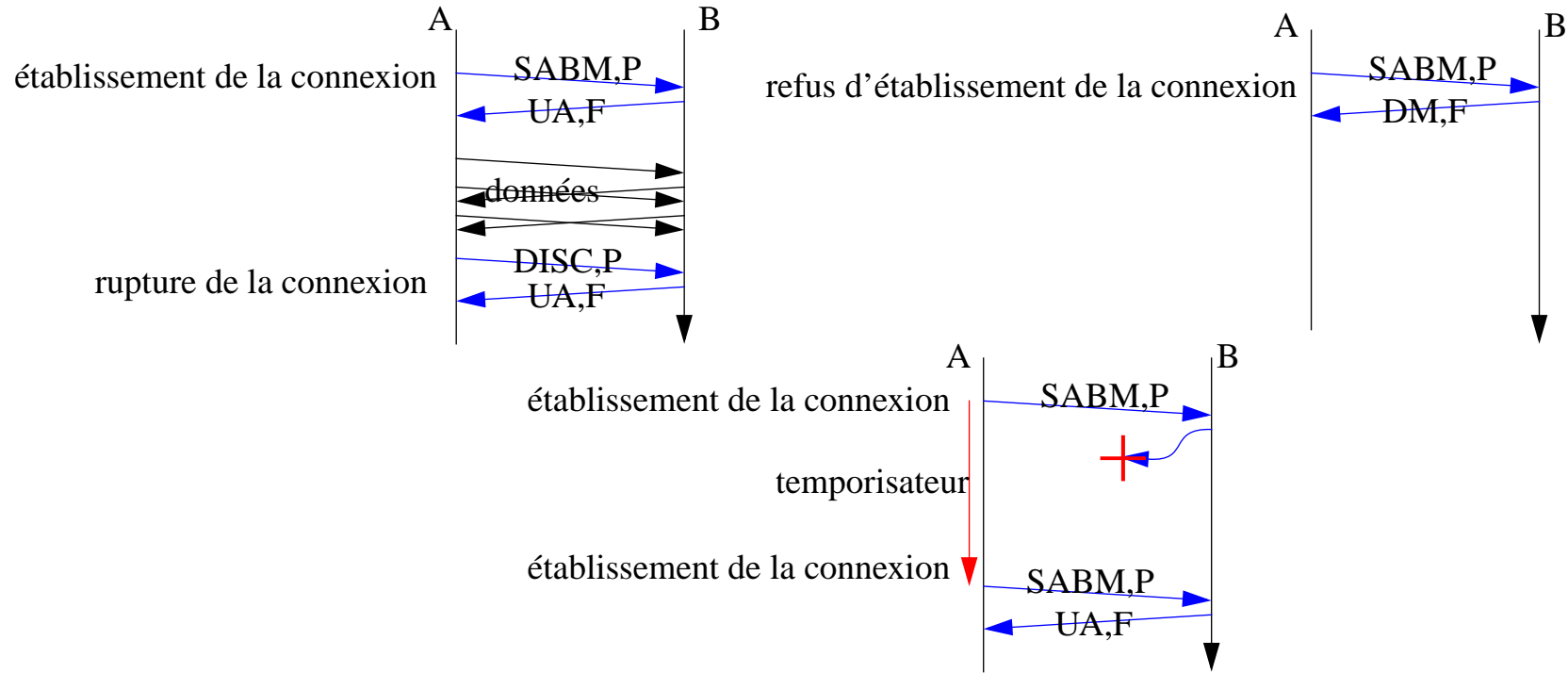
- Si le nombre de retransmissions n'est pas dépassé ( $< N2$ )
  - . alors on réémet la trame **I** telle qu'elle a été mémorisée.

##### A l'expiration du temporisateur **T2**

- émettre une trame **RR**.

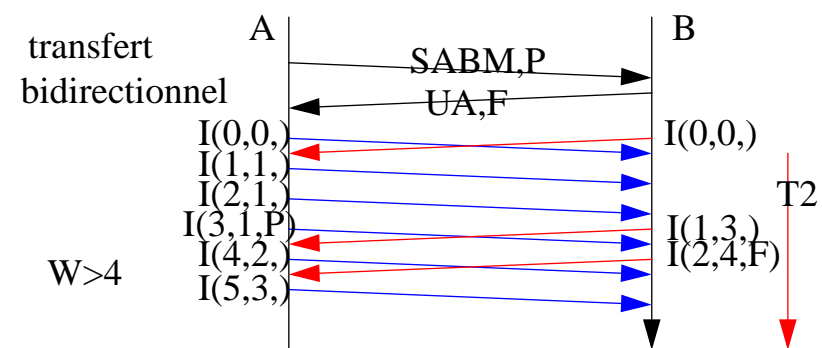
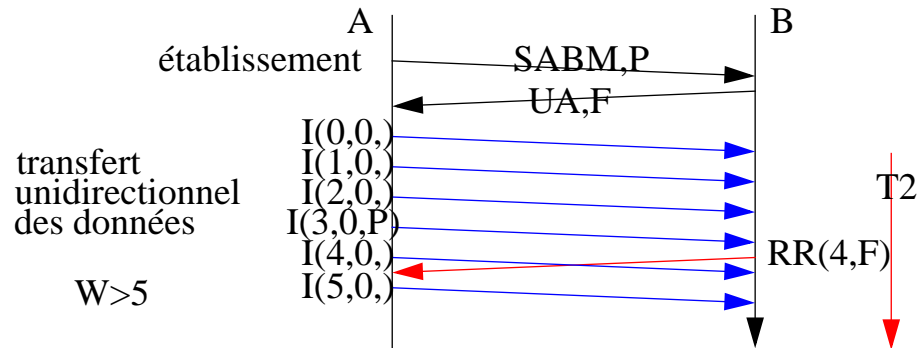


### 4.11. Scénario d'établissement et de rupture d'une connexion

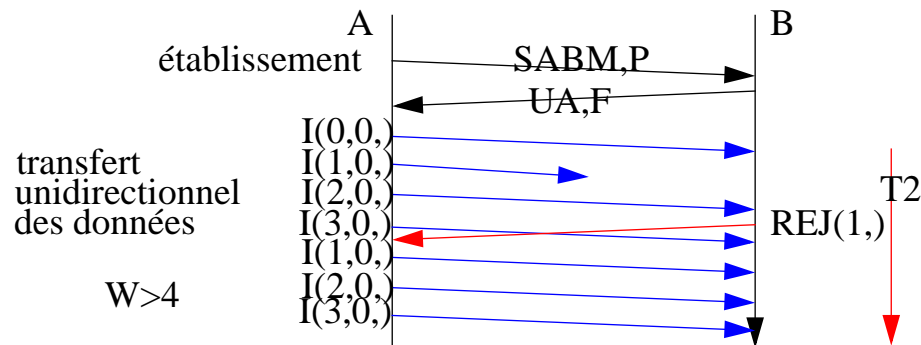


### 4.12. Scénario de transfert de données

#### Unidirectionnel ou bidirectionnel

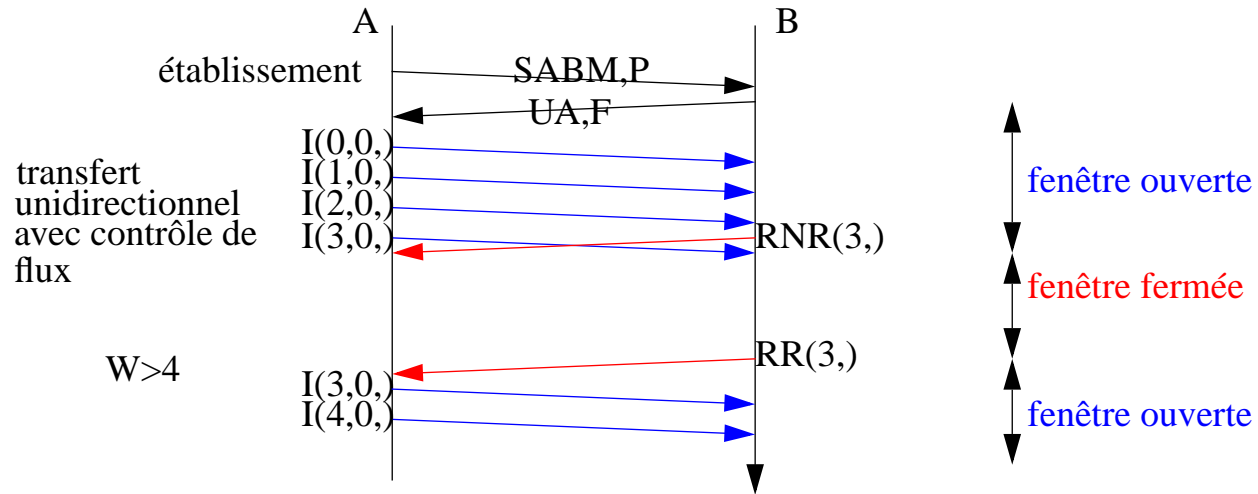


#### Avec perte



Notations : I(N(S),N(R),P/F)  
 REJ(N(R),P/F)  
 RR(N(R),P/F)  
 RNR(N(R),P/F)

### Avec contrôle de flux



## 5. Conclusion

L'ensemble des mécanismes précédemment cités nécessitent l'échange d'information, la création d'un contexte commun (partagé et réparti). Ces mécanismes se développent facilement au sein de ce qui est appelé une “**connexion**”.

Généralement les mécanismes d'augmentation de la fiabilité de la transmission sont développés au sein d'un protocole utilisant le mode connecté.

- ex : HDLC

Généralement les protocoles utilisant le mode non connecté ne proposent pas de mécanisme augmentant la fiabilité de la transmission.

- ex : Ethernet

Par abus de langage : on parle souvent de protocole fiable à la place de protocole implémentant des mécanismes chargés d'augmenter la fiabilité !

En effet, la fiabilité n'est jamais totalement assurée, elle n'est que probable (probabiliste) à cause des mécanismes de détection des erreurs.

De même par abus : on associe très souvent fiabilité au mode connecté.

On parle aussi de mode assuré, ce qui indique l'utilisation d'acquittements.

Le mécanisme de la **fenêtre coulissante** combine l'ensemble des techniques citées (contrôle de flux, d'erreur, de déséquence, et de duplication, numérotation, protection contre les erreurs, etc.)

### Attention

- il y a deux fenêtres coulissantes par connexion bidirectionnelle :
  - . une pour chaque sens de transmission
- pour chaque fenêtre on peut distinguer 3 jeux de variables :
  - . celui de l'émetteur, celui du récepteur et celui des trames
  - . en fait ces variables implémentent de manière répartie la gestion du tampon circulaire
  - . les exemplaires de la même variable dans les différents jeux sont synchronisés !

De nombreux protocoles d'autres niveaux ou d'autres environnements protocolaires utilisent le mécanisme de la fenêtre coulissante :

- Par exemple : TCP - monde Internet, TP - couche Transport de l'OSI

D'autres protocoles de niveau Liaison de données existent :

- Ils doivent répondre à des besoins spécifiques notamment au sein des réseaux locaux (MAC) :
  - . l'adressage et la diffusion
  - . la gestion du conflit d'accès multiple au support physique unique et le multipoint.

La famille des protocoles de type HDLC est vaste :

- type de connexion : LAP ou LAP-B (SNRM, SARM, SABM)
- LAP-D (RNIS) ou LAP-X (Télex), SDLC (IBM) ou LLC (Réseaux Locaux)

La version LAP-B est elle-même adaptable :

- Procédure multi-liaison ou à liaison unique:
  - . éclatement d'une liaison dans plusieurs circuits de données.
- Format normal ou étendu (SABM, SABME)