Niveau Liaison
"Data link Layer"

Protocoles Point à Point

Réseaux locaux

Niveau Liaison
"Data link Layer"

Protocoles Point à Point
"Point to Point Protocols"

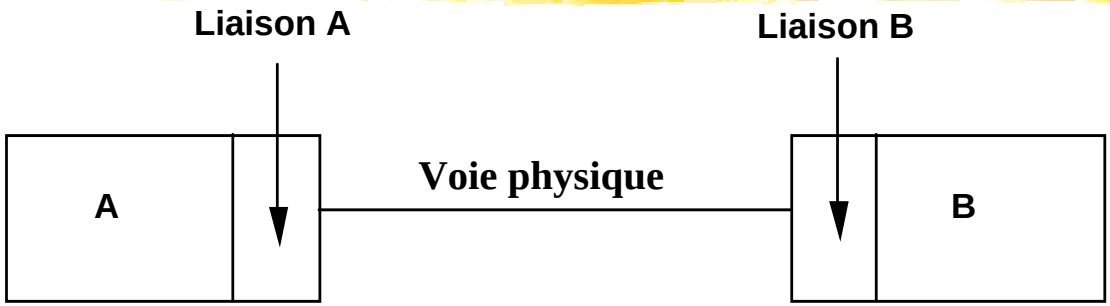
Introduction.
I Solutions générales.
II Protocoles industriels
Conclusion.

Niveau Liaison En Point à point

Introduction

Problèmes résolus dans les protocoles de niveau liaison en point à point

Situation du sujet: le niveau liaison point à point



- **Le niveau liaison contrôle les communications** réalisées au niveau physique entre des sites reliés par une voie point à point (implantation logicielle).
- **Le niveau liaison masque les caractéristiques** de la couche physique (tous types de voies) aux entités de réseau.

  Exemple type: PPP dans l'Internet
- **Le niveau liaison point à point résout des problèmes non résolus au niveau physique** (transparent suivant).
- **Niveau liaison point à point: très stabilisé** (théorie 1960/1970, normes 1970/1990)

Principaux problèmes résolus dans le niveau liaison point à point

Selon les choix de conception on trouve **tout ou partie** de solutions au problèmes **suivants :**

- 1 **Délimitation** des trames.
- 2 Contrôle **d'erreurs.**
- 3 Contrôle **de flux.**
- 4 Contrôle **de séquence.**
- 5 Gestion **des connexions.**
- 6 **Multiplexage.**
- 7 Fonctions **d'administration.**

1 - Délimitation des trames ("tramage", "framing").

- **Fonction basique de tous les protocoles de liaison : acheminer des trames** (des suites binaires structurées).
  - **Nécessite** de retrouver la **synchronisation trame** en relation avec **les erreurs** (trames bruitées).
- **Fonction considérée comme de niveau liaison**
  - Synchronisation bit => niveau physique (horloges).
  - Synchronisation trame => plutôt niveau liaison (délimiteurs)
- **Exemple type :** Fonction quasi unique dans les protocoles de liaison sans connexion
  - Un seul type de trame existe qui doit essentiellement être délimité

    => Protocole **SLIP** ("Serial Line Interface Protocol")

2 - Détection et correction des erreurs : A) Contrôle d'erreurs ("Error control")

- **Bruits au niveau physique** : Solution niveau liaison
  => codes détecteurs d'erreurs.

- **A) Si le taux d'erreurs est inacceptable**

Taux d'erreur de la voie physique : exemple 10^{-3} à 10^{-5} par bits.

=> Le protocole de liaison amène **le taux d'erreur résiduel à un niveau acceptable** au moyen de retransmissions (exemple > 10^{-12} par bits).

=> Protocoles avec contrôle d'erreurs : **LAPB** ('Link Access Protocol B'), IEEE 802.11 (**WiFi** "Wireless Fidelity").

2 - Détection et correction des erreurs : B) Destruction silencieuse

- **B) Si le taux d'erreurs du au bruit est considéré comme acceptable** :

  - **Pas de contrôle d'erreur** (le contrôle d'erreur est renvoyé au niveau transport).
  - => **Destruction silencieuse** des trames erronées ("Silent Discard").
  - => **Exemple type : PPP** ('Point to Point Protocol' en mode standard, **Ethernet** IEEE802.3).

3 - Contrôle de flux ('Flow Control')

- **Cas 1 : Contrôle de flux jugé indispensable**
  - **Problème : adapter la vitesse de l'émetteur à celle du récepteur**.
  - **Si le débit d'un émetteur est trop important** relativement aux performances d'un récepteur
  - On doit **réguler** les échanges : ralentir l'émetteur pour éviter les **pertes** de trames dues à **l'impossibilité de stocker** les trames entrantes en cas de **trafic élevé**.
  - => Exemples : Protocoles **LAPB**, **IEEE 802.3x** (Contrôle de flux en Ethernet)

- **Cas 2 : Si les pertes de trames d'un récepteur sont jugées acceptables :** pas de contrôle de flux

  => Exemples : Protocoles **SLIP**, **PPP** (mode standard).

4 - Livraison en séquence (respect de la causalité)

- **Voie physique : médium 'causal' :**
  - Causalité de propagation des ondes: les bits ne peuvent **remonter le temps** (arriver avant d'être émis) ou se **dépasser** sur les câbles.
- **Problème:** les erreurs de transmission et les retransmissions produisent des **modifications de la séquence émise (lacunes, déséquencements)** ou des **duplications.**
  - Le contrôle de séquence assure, en liaison avec le contrôle d'erreur et le contrôle de flux, la délivrance des trames au récepteur dans l'ordre **exact** de la soumission par l'émetteur
- **Exemples :**
  - Protocoles de liaison avec contrôle de séquence : Protocoles **LAPB, IEEE 802.11 Wifi.**
  - Protocole sans contrôle de séquence : **SLIP, PPP** en standard.

5 - Gestion des connexions

Protocoles en mode connecté

=> Identification de flots séparés par échanges préalables de messages de connexion (à la fin messages de déconnexion).

- Exemples de mode connecté **BSC** (Binary synchornous Communications) **, LAPB, LLC2** (Logical Link Control 2), **PPP**.

Protocoles en mode non connecté

=> Les échanges ont lieu avec des trames de données qui peuvent être transmises à tous moments.

- Exemples d'implantation du mode sans connexion: **SLIP**, **LLC1** (Logical Link Control 1 sur réseau local), **Ethernet**, **Wifi**.

6 - Multiplexage

- **Multiplexage** : gestion de plusieurs flots de données identifiés et délivrés à des entités de réseau différentes.
  - **Utilisation 1 :** Coexistence de trafics pour des architectures de réseaux différentes sur la même voie physique :

    Exemples : Coexistence Internet, Novell, Apple Talk , SNA, OSI ….
  - **Utilisation 2** : coexistence d'un trafic pour des entités de réseaux différentes dans une même architecture:

    Exemples : En Internet protocoles ICMP, ARP , RARP…
- **Solution statique** : Utilisation d'une zone identifiant les différents flots selon des valeurs fixes (normalisées).
- **Exemples : PPP** , **Ethernet/DIX** (Digital/Intel/Xerox), **LLC/SNAP** (Logical Link Control/Subnetwork Access Protocol)

7 - Administration du niveau liaison

- **Fonctions d'administration ('Network Management')** : généralement selon une approche normalisée et par niveaux
  - Exemple : **SNMP** 'Simple Network Management Protocol'.
- **Cas du niveau liaison** : des fonctions spécifiques sont définies dans les protocoles (réglages, adressage, ...)
- **Essentiellement valable pour le protocole PPP Internet:** Un protocole qui développe beaucoup cet aspect
  - **Gestion des configurations** : négociation de paramètres de fonctionnement, fourniture d'adresses.
  - **Gestion des pannes** : mécanismes de diagnostic (bouclage).
  - **Gestion de la sécurité** : authentification d'un nouveau client.
  - **Gestion des performances** : pas d'exemple au niveau liaison.
  - **Gestion de facturation des ressources** : pas d'exemple de liaison.

Niveau Liaison En Point à point

Chapitre I

Solutions générales pour la réalisation des protocoles de liaison en point à point

Délimitation des trames

Contrôle d'erreur, de flux, de séquence

I.1
Délimitation des trames

Délimitation des trames
Position du problème

- **Retrouver la synchronisation trame :**
  - **mise en correspondance "une pour une**"
  - entre une trame émise et une trame reçue.
- **Difficultés du problème**
  - Le bruit peut faire perdre la synchronisation trame.
  - Le bruit peut créer des trames fictives (à une trame émise peut correspondre une trame ou plusieurs trames reçues en raison du bruit découpant ou créant des trames parasites).
- **La solution de base:**
  - Se donner un mécanisme de **séparation des trames**.
  - Associer à une trame **un code polynomial détecteur d'erreur.**
  - Quand on décide qu'**une trame est délimitée en réception** on vérifie la **correction** au moyen du code polynomial.
  - Si elle est **incorrecte ou séparée** en plusieurs trames incorrectes **on abandonne les informations reçues** (destruction silencieuse).

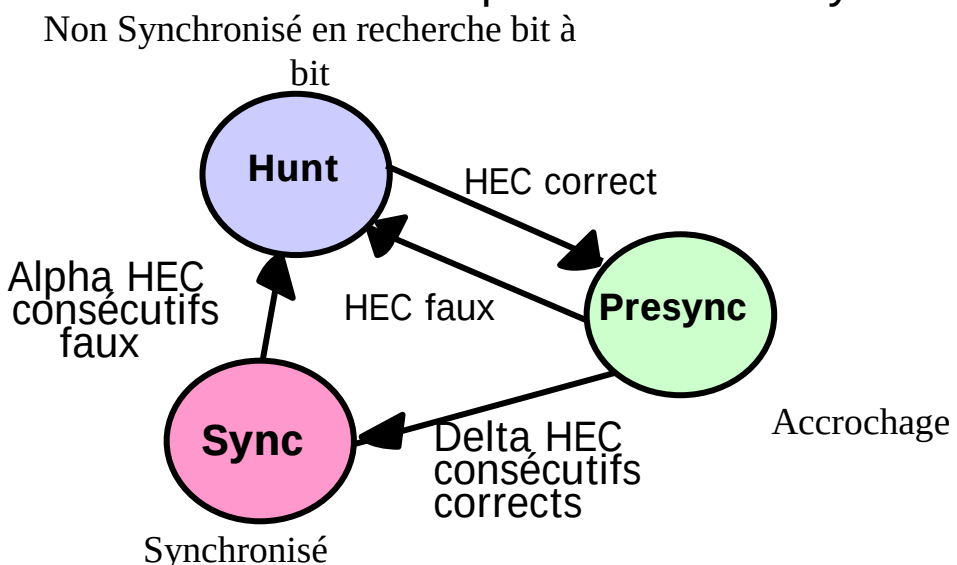
Trames sans délimiteurs (dans un flux continu synchrone)

- **Solution utilisée pour des trames de longueur fixe en flux continu:**
  - Pour chaque trame : un code polynomial correct **se répète en position fixe**.
  - On fait fonctionner un automate qui se **recale sur chaque bit considéré comme un début de trame** et vérifie l'exactitude du code polynomial.
  - **Approche probabiliste :** On peut se recaler une fois par hasard mais pour delta trames successives correctes => très faible risque d'erreur de synchro.

  - **Exemple ATM**

- Cellules de 53 octets

- HEC : Header Error Control (code polynomial)



- **Pour des trames de longueur variable (pas d'exemple industriel)**
  - Difficultés supplémentaires

Trames avec délimiteurs: Transparence caractère ('Character Stuffing')

- **Trames constituées de caractères** d'un alphabet normalisé (ex ASCII,EBCDIC) => les trames sont multiples de 8 bits.
- **Caractères de contrôle** : des codes caractères particuliers.
- **Solution retenue en** : **BSC**, **SLIP**, **PPP** mode standard.

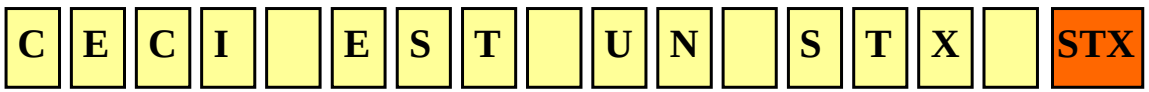
Exemple dans les protocoles BSC ('Binary Synchronous Communication')

- **STX** ("**S**tart of **T**e**X**t") - Délimiteur début de bloc de texte
- **ETX** ("**E**nd of **T**e**X**t") - Délimiteur fin de bloc de texte
- **DLE** ("**D**ata **L**ink **E**scape") - Échappement de liaison

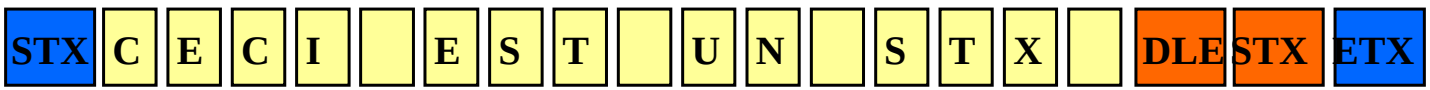
- **Pour une trame purement alphanumérique: pas de problème d'ambiguïté** entre caractères de contrôle et données dans les blocs texte.
- **Si une trame comporte des caractères de contrôle** parmi des caractères alphanumériques d'un bloc => **transparence caractère**
  - Tout caractère de contrôle (qui n'est pas le délimiteur début ou fin) apparaissant dans le bloc est précédé de DLE.
  - ETX -> DLE ETX; STX -> DLE STX; DLE -> DLE DLE
  - A la réception les DLE ajoutés pour la transparence sont retirés.

Exemple de transparence caractère en BSC

- **Bloc à transmettre :**



- **Bloc transmis** :



- **Pas de problème :** pour restituer le bloc initial
  - Pour distinguer les délimiteurs réels en début et fin de bloc (sans DLE devant)
  - Pour distinguer les codes caractères identiques transmis dans la charge utile (ils sont précédés d'un DLE que l'on supprime à la réception).

Trames avec délimiteurs: Transparence binaire (‘Bit Stuffing’)

- **Les trames sont constituées de suites binaires**.
- **Chaque trame est délimitée** par une suite binaire réservée.

- **Exemple de la famille des protocoles SDLC/HDLC/LAPB:**
  Délimiteur : drapeau, fanion ou "flag" huit bits **01111110**.
- **Le fanion ne doit jamais apparaître** dans la suite binaire d’une trame sous peine de décider la fin de trame.
- **Procédure de transparence binaire**

  - En sortie quand on a **5 bits 1 consécutifs** => on insère automatiquement **un bit 0** après

  - **En entrée le bit 0 suivant 5 bit 1 est enlevé** (sauf bien sur pour les fanions début et fin qui sont interprétés comme des délimiteurs).

Exemple de transparence binaire en SDLC/HDLC/LAPB

- **Suite binaire à transmettre :**

01001000001111101111100

- **Suite binaire après adjonction des bits de transparence :**

01001000001111101011111000

- **Trame transmise avec ses délimiteurs (fanions) :**

01111110 01001000001111101011111000 01111110

- **En réception : suppression des bits de transparence et des fanions.**

Trames avec délimiteurs: Violation de code

- **Problèmes des techniques de transparence** basées sur l'utilisation de délimiteurs formés de **configurations binaires légales** (caractères de contrôle, fanions, ...).
  - **Allongement des trames** du aux informations de transparence (quelques pour cent).
  - **Temps perdu à l'émission et à la réception** (surtout en logiciel).
- **Autres solutions :** définir des modulations utilisées comme délimiteurs (en plus de la modulation des données trame 0,1) .
  - **Augmenter la valence** des signaux de niveau physique pour créer des délimiteurs.
  - Ces signaux **ne peuvent donc apparaître dans le flot normal** des données d'une trame.
- **De nombreuse variantes** de ce principe ont été développées.
  - Les problèmes de la transparence sont résolus.
  - Mais le modulateur doit être plus complexe.
  - Solution très utilisée (par exemple dans les réseaux locaux).

Exemple de délimiteurs dans le protocole IEEE 802.3u Ethernet 100 Base T

- **Code 4B/5B**
  - Données binaires utilisateur: groupes de 4 bits
  - Données transmises: pour chaque groupe de 4 bits un groupe de 5 bits.
  - 16 configurations nécessaires pour les données utilisateurs.
  - 16 configurations disponibles pour des besoins protocolaires : symboles inter trames, délimiteurs etc
- **Délimiteurs**
  - Symbole J 11000 First Start Of Frame (Invalid code)
  - Symbole K 10001 Second Start of Frame (Invalid code)
  - Symbole T 01101 First End Of Frame (Invalid code)
  - Symbole R 00111 Second End of Frame (Invalid code)
  - Symbole I 11111 Idle (Invalid code)
- **Structure d'une trame**

I I I I | J K ..... Groupes données de trame ..... T R | I

- **Solutions en FDDI, Ethernet 1000 , 10G : très similaires.**

I.2
Contrôle d'erreur, de flux, de séquence

Introduction : présentation générale des protocoles étudiés

- **Solutions de complexité croissante** aux problèmes:
  - Contrôle d'**erreur.**
  - Contrôle de **flux**
  - Respect de la **causalité** (livraison en séquence).
- **Traitement conjoint des problèmes**
- **Codage des solutions en langage style ADA**
- **Protocoles examinés** (selon l'approche de A. Tannenbaum)

- Protocole 1 **"Sans contrôle d'erreur et de flux"**
- Protocole 2 **"Envoyer et attendre"**
- Protocole 3 **"Bit alterné"**
- Protocole 4 **"A fenêtre glissante et réception ordonnée"**
- Protocole 5 **"A fenêtre glissante et rejet sélectif"**

Protocole 1 : Sans contrôle d'erreur ni contrôle de flux

- **Pas de contrôle de flux**
  - **La couche réseau du récepteur est toujours prête à recevoir:** les pertes de trames dues au manque de contrôle de flux sont négligeables (temps de calcul négligeables, mémoire toujours disponible pour stocker).
  - **Ou le contrôle de flux est prévu ailleurs** (assuré dans un autre niveau).
- **Pas de contrôle d'erreur**
  - **Les pertes de trames dues au bruit sont tolérables**.
  - **Ou le contrôle d'erreur est prévu ailleurs** (assuré dans un autre niveau).

- **Nature de la solution**
- **Solution de base** d'un protocole sans connexion qui se contente d'acheminer des trames et laisse aux niveaux supérieurs toutes les tâches.
- **Mise en place de la programmation** pour les solutions suivantes.
- Le code ne décrit qu'une transmission **unidirectionnelle.**

- **Solution des protocoles Internet :** SLIP, PPP en mode standard.

Protocole 1 : Déclarations

```
-- Zone de données utilisateur (paquet réseau)
-- Par exemple: taille de paquet 1500 octets.
type paquet is array (integer range 1..1500) of character ;
-- Type de trame de niveau liaison utilisée.
-- Une seule information : la charge utile (le paquet).
type trame is record
     info : paquet ;
end record;
-- Type événement en entrée.
-- Un seul événement : l'arrivée d'une trame
type Type_Evénement = (Arrivée_Trame ) ;
```

Protocole 1 : Codage de l'émetteur

```
procedure émetteur_1 is
s           : trame ;      -- La trame liaison en émission
tampon      : paquet ;     -- Le paquet réseau à émettre
begin
   loop
   recevoir_couche_réseau (tampon) ;   -- Paquet à envoyer
   s.info := tampon ;                  -- Préparer une trame
   envoyer_couche_physique(s) ; -- La faire émettre
   end loop                            -- Boucle infinie
end emetteur_1 ;
```

Protocole 1 : Codage du récepteur

```
--
-- Procédure exécutée par le récepteur
--
procedure récepteur_1 is
événement    : Type_Evénement ;-- Événement à traiter;
r            : trame ;         -- La trame en réception
begin
   loop
   attendre (événement) ;          -- Attendre une arrivée
   recevoir_couche_physique (r) ;-- Prendre trame arrivée
   envoyer_couche_réseau(r.info);-- Passer à l'utilisateur
   end loop                        -- Boucle infinie
end récepteur_1;
```

Protocole 2 : Arrêt et attente ("Stop and Wait")

- **Solution simpliste uniquement de contrôle de flux**.

- **Idée de base :** pour ne pas saturer le récepteur **freiner l'émetteur**.
- **Solution de rétroaction du récepteur sur l'émetteur**:

  . Le récepteur informe l'émetteur qu'il peut **accepter un nouvelle trame** en envoyant une trame de service ne contenant pas de données.

  . Cette trame s'appelle en réseau **un crédit (CDT)** : un crédit d'une unité donne un droit pour émettre une nouvelle trame.

  . L'émetteur **doit attendre d'avoir un crédit** pour envoyer une trame.

  => Famille des solutions de contrôle de flux: **Solutions basées crédits.**

Protocole 2 : Codage de l'émetteur

```
--
-- Code présenté => Solution unidirectionnelle
-- Une voie de retour pour des trames de service (toujours des crédits).
-- Déclarations : pas de changement par rapport au protocole 1
--
procedure émetteur_2 is
-- Les variations par rapport au code du protocole 1 sont en italique.
événement : Type_Evénement ;              -- Un événement à traiter
s         : trame ;                       -- Trame en émission
tampon    : paquet ;                      -- Paquet à émettre
begin
    loop
    recevoir_couche_réseau (tampon) ;     -- Un tampon à envoyer
    s.info := tampon ;                    -- Préparer une trame
    envoyer_couche_physique(s) ;          -- La faire émettre
    attendre(événement) ;                 -- Attendre un crédit
    end loop                              -- Boucle infinie
end émetteur_2 ;
```

Protocole 2 : Codage du récepteur

```
--
procedure récepteur_2 is
--
-- Les variations par rapport au code du protocole 1 sont en italique
événement: Type_Evénement;          -- Evénement à traiter
r          : trame;                 -- Une trame en réception
s          :trame;                  --Une trame de crédit
begin
    loop
    attendre (événement) ;          -- Attendre arrivée de trame
    recevoir_couche_physique (r);   -- Prendre trame arrivée
    envoyer_couche_réseau(r.info) ; -- Passer à l'utilisateur
    envoyer_couche_physique(s);     -- Envoyer le crédit
    end loop                        -- Boucle infinie
end récepteur_2;
```

Protocole 3 : PAR "Protocole avec Acquittement et Retransmission"

- **Solution globale: problèmes de contrôle de flux**, **d'erreur**, **de séquence**
  - Pour une voie physique **bruitée** (utilisation d'un code détecteur d'erreur)
  - et avec un récepteur à **capacité de traitement** et de **mémoire limitée.**

- **Nombreuses appellations pour une classe de protocoles voisins.**
  - Protocole **PAR ("Positive Acknowledgment with Retry").**
  - Protocole du **Bit alterné ABP ("Alternate bit protocol").**
  - Protocole **ARQ ("Automatic Repeat Request").**

- **Introduction des notions suivantes**
  - A) **Acquittement positif** (d'une trame correcte).
  - B) **Délai de garde** (pour retransmission sur erreur).
  - C) **Identification des trames** (par un numéro de séquence).

- **Exemples d'implantation de protocoles de ce type :** BSC 'Binary Synchronous Communications', GSM 'Global System for Mobile', WIFI

A) Acquittements positifs "Positive Acknowledgments"

- **Acquittement positif** : une information protocolaire qui circule pour indiquer la bonne réception d'une trame de donnée.
- **Utilisation des acquittements positifs**

Règle 1 : Une trame n'est **acquittée positivement** que si elle est **reçue correctement** (code détecteur correct).

Règle 2 : Toute **trame correcte doit être acquittée positivement** afin que l'émetteur ne la retransmette plus.

- **Remarques**

- Stratégie de reprise : en acquittement positif la reprise sur erreur est **confiée à l'émetteur** qui doit s'assurer qu'une trame a bien été reçue.

- Acquittements positifs et crédits

=> L'acquittement positif a une signification dans **le contrôle d'erreur** alors que le crédit sert dans le **contrôle de flux**.

=> Une trame unique (baptisée acquittement "ACK") est souvent utilisée (exemple dans le protocole PAR) à double sens pour signifier :

. **dernière trame correcte** (acquittement positif)

. **acceptation d'une autre trame** (crédit de une trame).

Acquittements négatifs "Negative Acknowledgments"

- **Acquittement négatif** : une information protocolaire indiquant la mauvaise réception d'une trame de donnée.
- **Utilisation des acquittements négatifs**

- **Règle 1** : Une trame n'est acquittée négativement que si le destinataire ne l'a pas reçue alors qu'il sait qu'elle a été émise.

Apprentissage : . Signal indiquant une trame en erreur (rare).

. Absence d'une trame dans une suite numérotée.

- **Règle2** : Un acquittement négatif est une demande de retransmission.

- **Remarques:**

- **Stratégie de reprise : en acquittements négatifs** le traitement des erreurs est confié au récepteur qui doit découvrir l'absence d'une trame, en demander la retransmission pour que celle-ci soit bien reçue.
- **Le protocole PAR n'a pas besoin** des acquittements négatifs.
- On peut concevoir de **multiples variantes** de protocoles utilisant à la fois des stratégies d'acquittements négatifs et positifs.
- Des protocoles probabilistes **basés uniquement sur** les acquittements négatifs sont possibles.

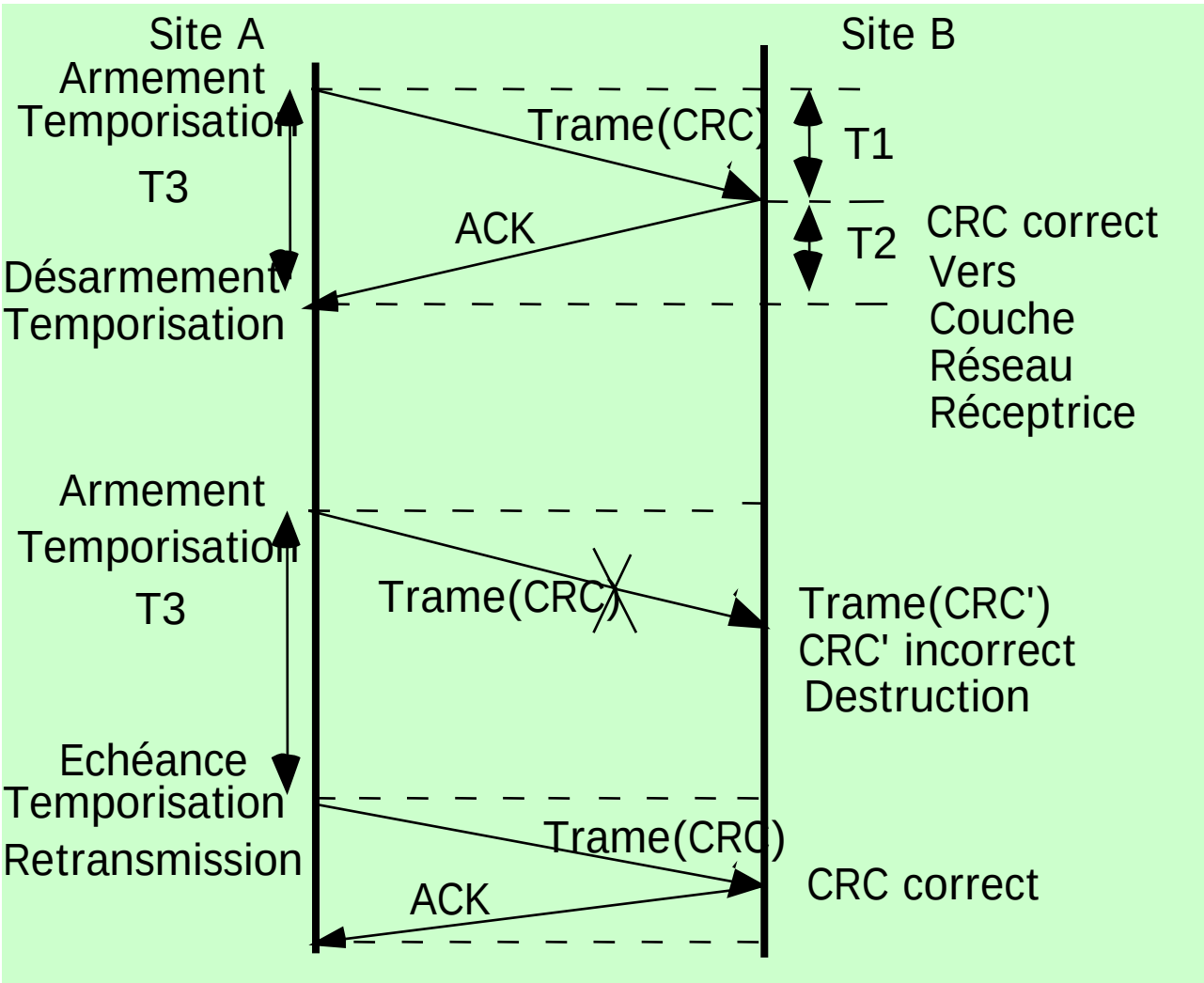
B) Délais de garde : Temporisateurs , "Timers"

- Nécessité de **conserver copie d'une trame** .

  . Si la trame est bruitée on doit la retransmettre.

  . Si la trame est correcte mais l'acquittement positif bruité on ne sait pas si la trame a été reçue => on retransmet.
- Problème: en l'absence d'acquittement positif on ne peut conserver **indéfiniment** les copies.
- Utilisation **d'un délai de garde** (temporisateur)
  - **Réveil de l'émetteur à échéance.**
  - **Retransmission de la trame**
- Protocole PAR : seul mécanisme de reprise, l'émetteur **retransmet systématiquement une trame** sur échéance du délai.
- **Remarque** : L'usage du délai de garde ralentit la reprise sur erreur car le délai de garde doit être nettement supérieur à la somme des délais d'émission et de retour d'acquittement.

Fonctionnement avec délais de garde (temporisateurs)

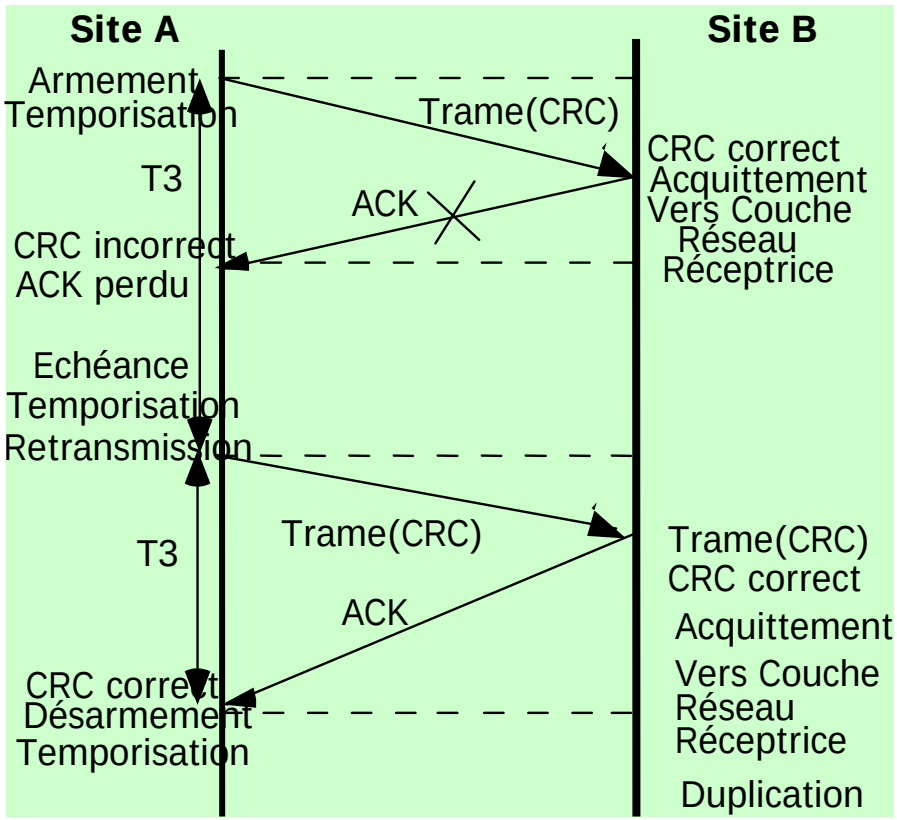


C) Identification des trames : numéros de séquence ("Sequence numbers")

Exemple de problème sans identifiant

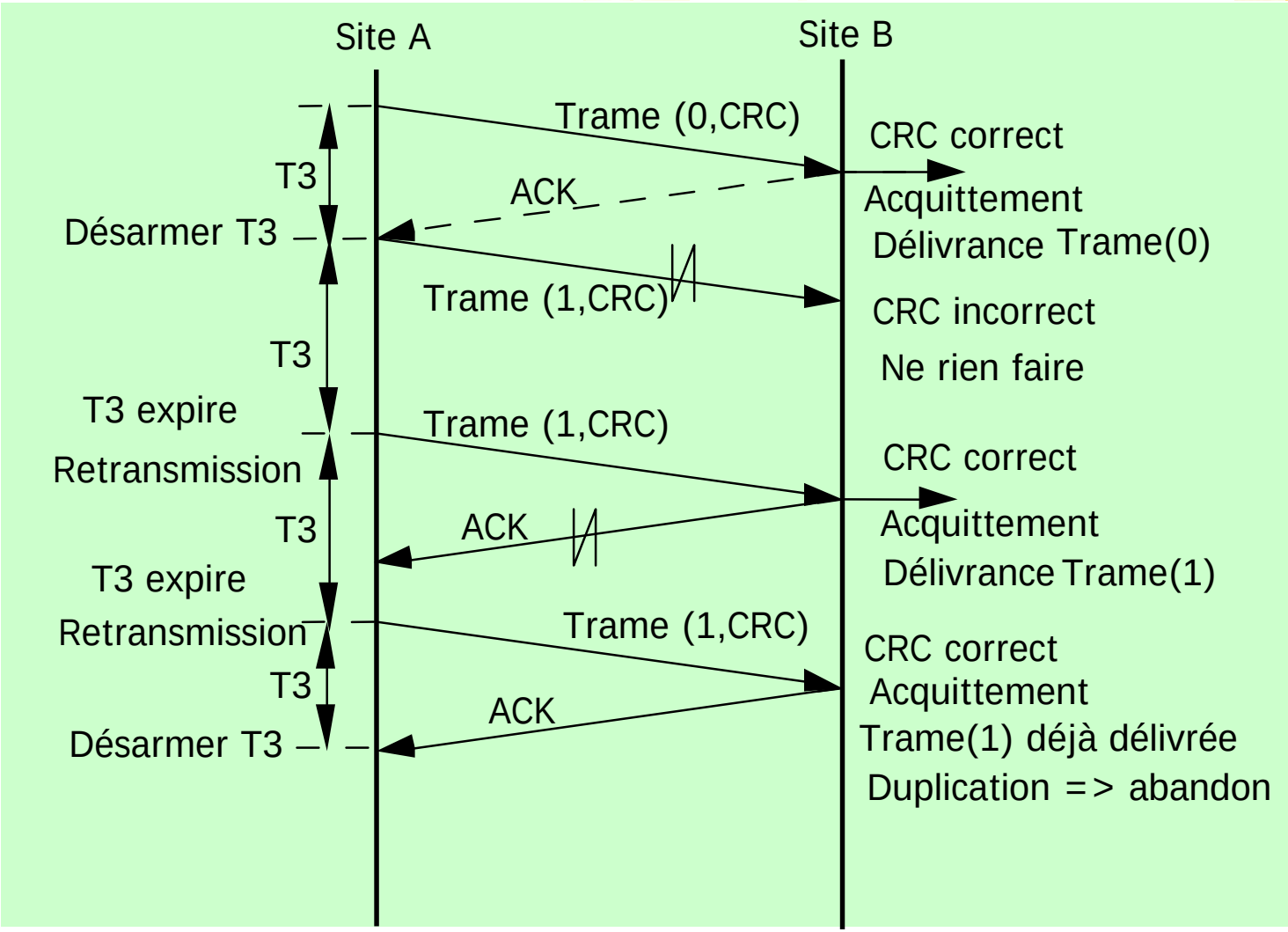
- Une trame est reçue **correctement** mais l'**acquittement positif correspondant se perd ...**
- La couche liaison récepteur reçoit **deux fois la trame** puisque l'émetteur réémet.

=> **Duplication non détectée (non respect de la séquence)**

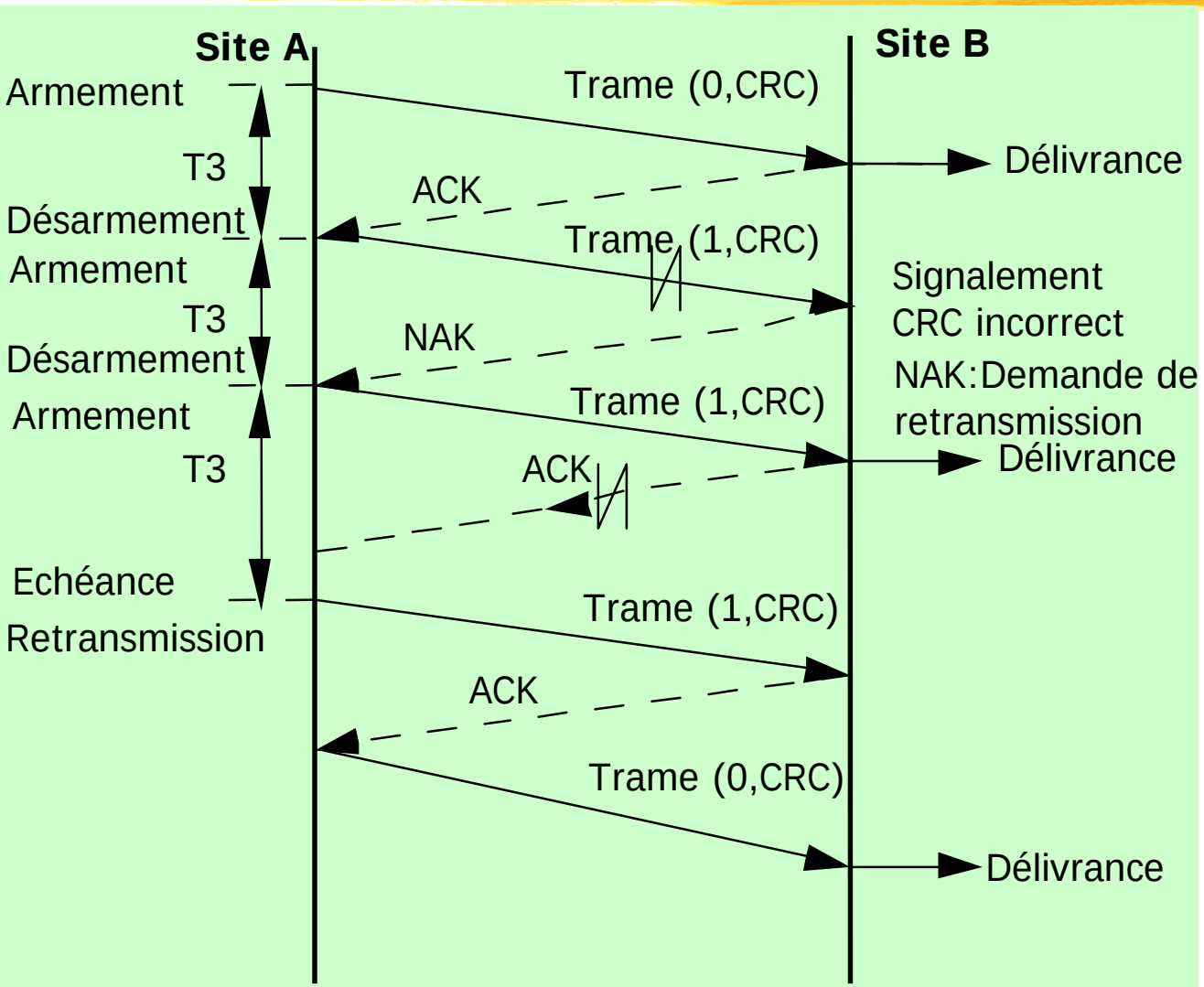


- Nécessité **d'un numéro de séquence** de trame (identifiant entier de la trame) pour éviter les **duplications** et assurer le **respect de la séquence d'émission** (causalité).

Exemple de fonctionnement du protocole PAR



Fonctionnement en utilisant en plus des acquittements négatifs



- Les acquittements négatifs **ne sont pas indispensables** car le fonctionnement de base est fourni par **les acquittements positifs, les temporisateurs et les numéros de séquence**.
- Les acquittements négatifs, servent à **accélérer les retransmissions** en cas d'erreur.

Protocole 3 PAR : Codage
Présentation de la solution codée

- **Utilisation unique des acquittements positifs**.
- Utilisation d'un **délai de garde** (fonctions d'armement et de désarmement d'un temporisateur).
- **Identification des trames** sur un bit (numéros de séquence 0 et 1) => Protocole de bit alterné.
- Solution **unidirectionnelle** (un seul sens est décrit)
- Utilisation d'une voie de retour pour la **circulation des acquittements positifs.**

Protocole 3 PAR : Codage des déclarations

```
--
-- Variations par rapport au protocole 2 en italique
--
-- Déclarations globales
--
-- Type numéro de séquence d'amplitude 0..maxseq=1
maxseq: constant :=1;
type numero_sequence is integer range 0..maxseq;
type paquet is array ( integer range 1..128 ) of character ;
type trame is record
        seq : numero_seq ;
        info : paquet ;
end record;
type Type_Evénement = (Arrivée_Trame, Erreur_Trame, Horloge);
```

Protocole 3 PAR : Codage de l'émetteur

```
procedure emetteur_3 is
événement          :Type_Evénement;          -- Evénement à  traiter;
s                  :trame;                   -- Trame en émission;
tampon             :paquet;                  -- Paquet à émettre
Proch_Trame_A_Envoyer  :  numero_sequence;   -- Num prochaine trame émise
begin
Proch_Trame_A_Envoyer := 0;                  -- Init pour le premier message
recevoir_couche_réseau (tampon);             -- Un tampon est à envoyer
loop
    s.seq := Proch_Trame_A_Envoyer ;         -- Préparer numéro  de trame
    s.info := tampon ;                       -- Partie information usager
    envoyer_couche_physique(s) ;             -- Faire partir la trame
    démarrer_horloge (s.seq) ;               -- Armer un délai de garde
    attendre(événement) ;                    -- Attendre un crédit / un acquit
    if événement = arrivée_trame then        -- C'est l'acquittement attendu
            désarmer_horloge(s.seq) ;        -- Plus besoin d'un réveil
            inc(Proch_Trame_A_Envoyer:);     -- +1 pour le prochain message
            recevoir_couche_réseau (tampon); -- Un tampon est à envoyer
    endif   -- Cas d'une retombée de garde : on ne fait rien donc on réemet
end loop                                     -- Boucle infinie
end emetteur_3;
```

Protocole 3 PAR : Codage du récepteur

```
procedure récepteur_3 is
événement: Type_Evénement;          -- Un événement à traiter;
r          : trame;                 -- Une trame en réception
s          : trame;                 -- Une trame de crédit /acquit (ack)
trame_Attendue: numero_sequence     -- Numéro de séquence prochaine trame à recevoir
begin
trame_Attendue := 0 ;               -- Initialisation attente de la trame 0
loop
attendre (événement) ;              -- Attendre arrivée trame
-- Deux cas possibles: la trame est correcte ou en erreur
if événement == Arrivée_Trame then  -- Cas d'une trame correcte
    recevoir_couche_physique (r);   -- Prendre la trame arrivée
    if r.seq ==Trame_Attendue then  -- C'est la bonne trame
        envoyer_couche_réseau(r.info) ;  -- La passer à l'utilisateur
        inc (Trame_Attendue) ;      -- Préparer trame suivante
    endif;
-- Dans tous les cas : en séquence ou pas on doit acquitter
envoyer_couche_physique(s);         -- Envoyer le crédit / acquit
endif     -- Dans le cas d'une erreur on ignore la trame reçue
end loop                            -- Boucle infinie
end récepteur_3;
```

Protocole 3 PAR : Conclusion
1) Problèmes de robustesse

- Fonctionnement **correct** dans toute configuration:

  . de **perte de messages** ou **d'acquittements.**

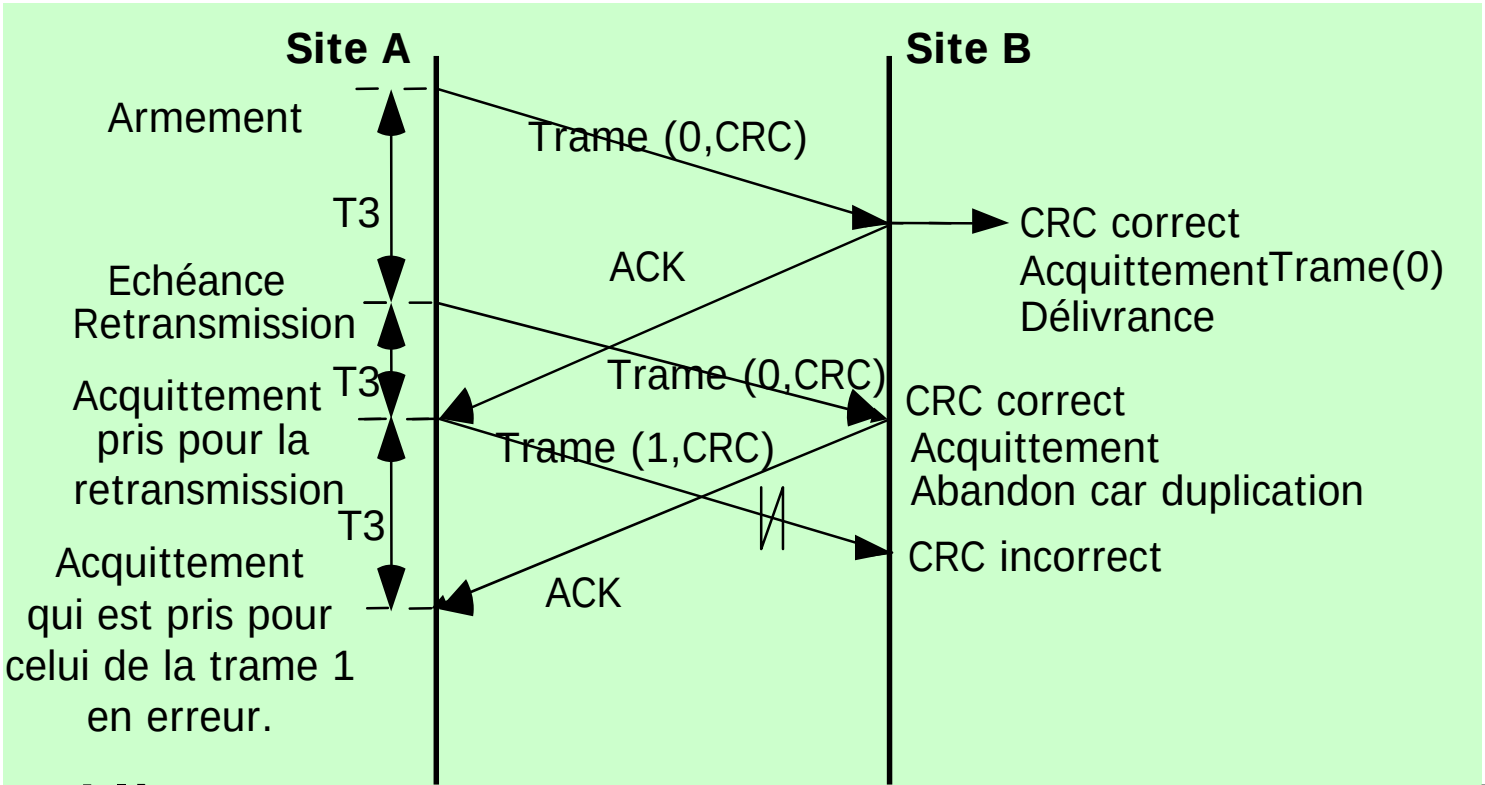
  . de **réglage des délais de garde.**

  . de **délai d'acheminement**.

- **Besoin de preuve d'un protocole :**
  - Sémantique (preuve de programme)
  - Temporelle (contraintes de synchro et de temps réel)
- Difficulté: Le protocole **n'est pas généralement pas reconfiguré** à chaque fois que l'on change de support physique ou de débit.

Exemple de problème de robustesse

- **Combinaison d'un délai de garde court et d'une perte de message.**



- **Le problème:** les acquittements ne sont pas correctement associés aux messages.
- **Solution :** Identifier dans les acquittements le numéro du message acquitté. => Solution des protocoles à fenêtres glissantes.

Protocole 3 PAR : Conclusion
2) Problèmes de performance

- **Chaque acquittement positif occupe une trame** uniquement dédiée à cette fonction => Relativement coûteux pour une opération très simple.

. ***Gestion d'une interruption*** *pour chaque acquittement*.

Sauvegarde du contexte + Traitant d'interruption + Déroutement couche liaison + Restauration du contexte.

. ***Réservation puis libération d'un tampon*** *pour chaque acquittement.*

- **Si les délais de transmission sont longs, l'attente d'un acquittement est longue => émetteur inactif**

. **Le taux d'utilisation** de la voie peut devenir très faible**.**

. **Exemple des voies satellite** (temps de propagation 250 milliseconde): Le taux d'utilisation de la voie peut tomber à quelques pourcent.

Protocole 4 : A fenêtre glissante avec réception des trames en séquence

- **Objectifs : corriger les défauts** du protocole PAR.

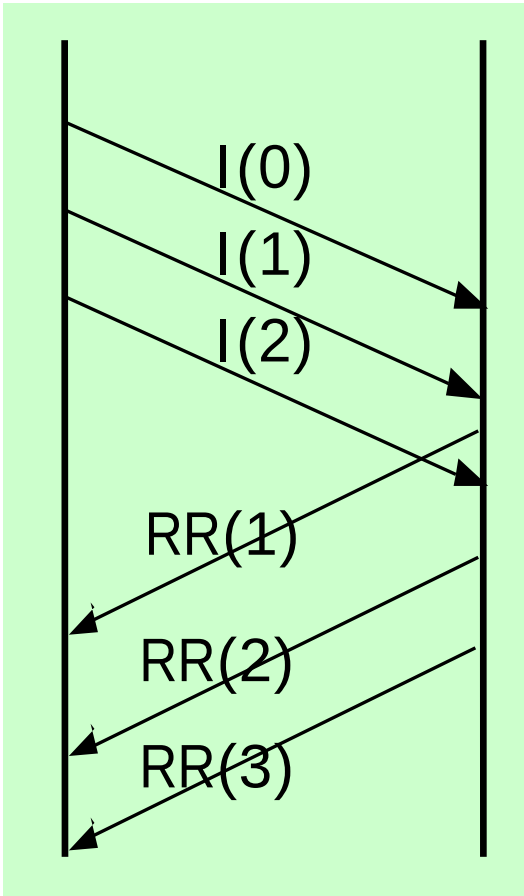
- **Protocole robuste** qui traite correctement le problème de contrôle d'erreur, de flux, de livraison en séquence
  . en présence de perte de messages ou d'acquittements
  . pour tout réglage des délais de garde

- **Protocole performant** qui optimise l'utilisation de la voie en fonction **des temps de transmission**
  . Protocole performant pour différents modes de fonctionnement.

- **Reprise** des outils déjà vus:
  - **Acquittements positifs**, **délais de garde**, **numéros de séquence** de message.
  - Utilisation des notations des protocoles type **LAPB**.

Amélioration 1 : Emission en anticipation ("Pipelining")

- **Anticipation des émissions :** ne pas attendre l'acquittement d'une trame avant d'envoyer les suivantes.
  - Pour éviter d'être **bloqué en attente** d'acquittement: arrêt & attente.
  - Pour résoudre le **problème d'inactivité** de la voie si les temps de transmission sont **longs** avec des messages **courts**.



- **Remarque :** L'anticipation **augmente la longueur** des trames en enchaînant la transmission de plusieurs trames consécutives. On **minimise** donc l'importance relative du temps de **propagation aller retour** et on **améliore le taux d'utilisation du canal**.

- **Exemple d'échange :** I(n) ("Information") : Trame d'information numéro n
  RR(n) ("Receiver Ready") : Trame d'acquittement pour I(n-1)

Anticipation : Règles de fonctionnement (1)

- **Règle 1** - L'émetteur doit **conserver copie des trames** jusqu'à réception de l'acquittement correspondant.

  => Pour retransmission si les trames ont été bruitées.

- **Règle 2** - Chaque trame est **identifiée par un numéro de séquence.**

  Les trames **successives** sont numérotées circulairement (modulo Maxseq+1) par des entiers **successifs**.

  => Pour respecter à la réception l'ordre d'émission.

  => Pour pouvoir éventuellement détecter des erreurs de transmission par des lacunes dans la suite des numéros reçus.

  Fonctionnement avec la règle 2:

  - L'expéditeur maintient une **variable d'état V(s)** qui définit le numéro de **la prochaine trame à émettre**.
  - Chaque trame est transmise avec **un numéro de séquence en émission N(S)** qui est la valeur courante de V(S) au moment de l'émission.

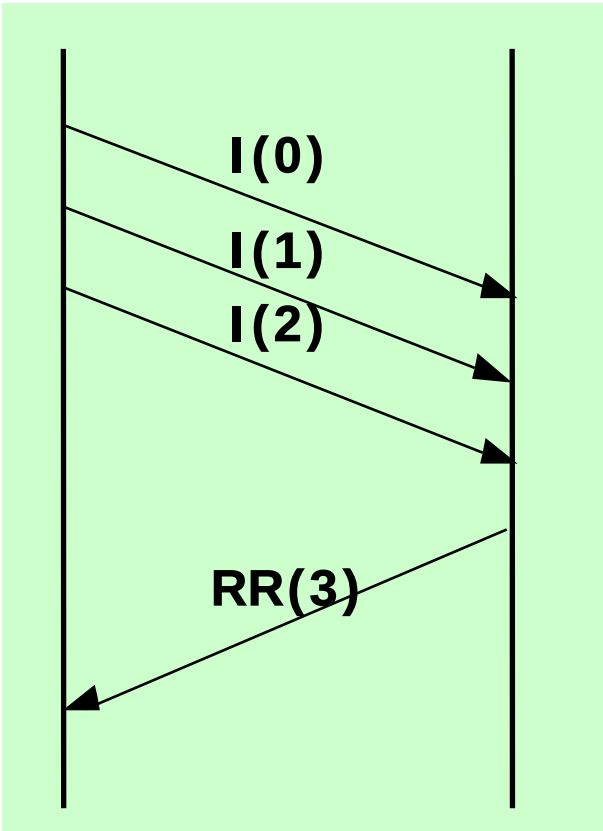
Anticipation : Règles de fonctionnement (2)

- **Règle 3** - Utilisation indispensable d'un ensemble de numéros de séquence de cardinal plus élevé que pour le bit alterné (2 numéros) pour permettre une anticipation réelle.
  - **Sur n bits** : 2^n numéros différents. Trames numérotées de 0 à Maxseq = 2^n - 1.
  - **Choix Maxseq=7** **n = 3** Peu de possibilités d'anticipation.
  - **Choix Maxseq=127** **n = 7** Encombrement dans les messages.

- **Règle 4** - L'anticipation ne peut pas être **autorisée sans limite**.
  - On n'exercerait **aucun contrôle de flux**.
  - On ne disposerait pas de la **capacité mémoire suffisante** pour les copies de trames en attente d'acquittement.
  - **Notion de crédit maximum statique** (taille maximum de la fenêtre d'anticipation).

Amélioration 2 : Regroupement des acquittements

- Il est **inutile et coûteux** d'envoyer **un acquittement pour chaque trame** d'information.
- **On peut acquitter plusieurs trames d'information I par une seule trame RR** d'accusé de réception à condition d'adopter une convention:
- **Acquittement pour I(n-1) vaut pour I(n-2), I(n-3), ... Etc toutes les trames en attente d'acquittement.**



Exemple d'émission avec anticipation et de regroupement des acquittements

Regroupement des acquittements : Règles de fonctionnement (1)

- **<u>Règle 1</u>** - **Le récepteur maintient une variable d'état V(R)** qui désigne le **numéro de séquence de la prochaine trame attendue**

  => Cette variable est incrémentée de 1 chaque fois qu'une trame est reçue en séquence sans erreur.

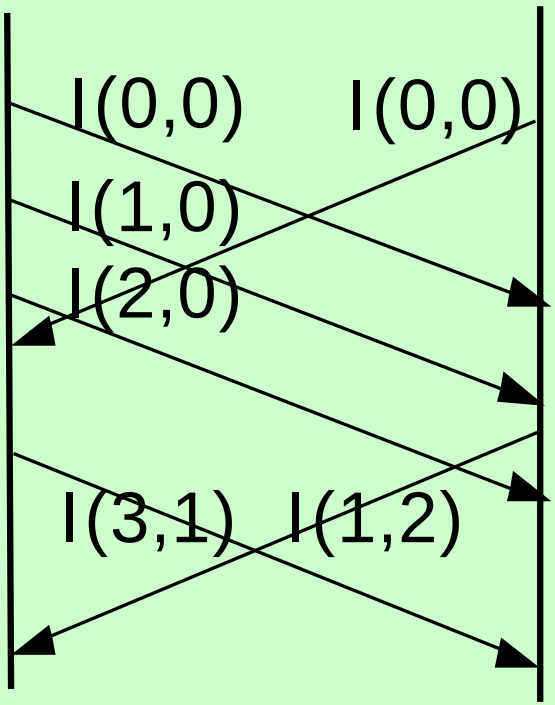
- **<u>Règle 2</u>** - **La variable de réception V(R) est reportée dans le champ N(R)** (le numéro de séquence de réception porté dans les acquittements retournés à l'émetteur RR(N(R)).

- **<u>Règle 3</u> - Cohérence initiale** des numéros de séquence et numéros d'acquittement: N(S) et N(R) = 0 au début de la communication.

- **<u>Règle 4</u> - La signification de l'acquittement:** RR(N(R)) acquitte toutes les trames en attente d'acquittement dont le numéro N(S) est inférieur ou égal à N(R)-1 => **Non pas une seule trame dont le numéro serait N(S)=N(R)-1.**

Amélioration 3 : Acquittements insérés dans les trames infos (piggybacking)

- **Insertion d'un champ acquittement** (N(R)) dans l'entête des trames d'infos.
  => Toute trame d'information devient **un acquittement positif pour des trames du trafic échangé en sens inverse**.
- **I(N(S),N(R)) acquitte** toutes les trames d'information transmises dans l'autre sens avec des numéros de séquence N(S) inférieur ou égal à N(R)-1
- **L'acquittement inséré coûte quelques bits par trame d'information**
- **Peu de trames** d'acquittement explicites.
- **Beaucoup plus de possibilités** d'acheminer des acquittements, sauf si le trafic d'informations est très faible dans un sens: retour à un acquittement explicite.

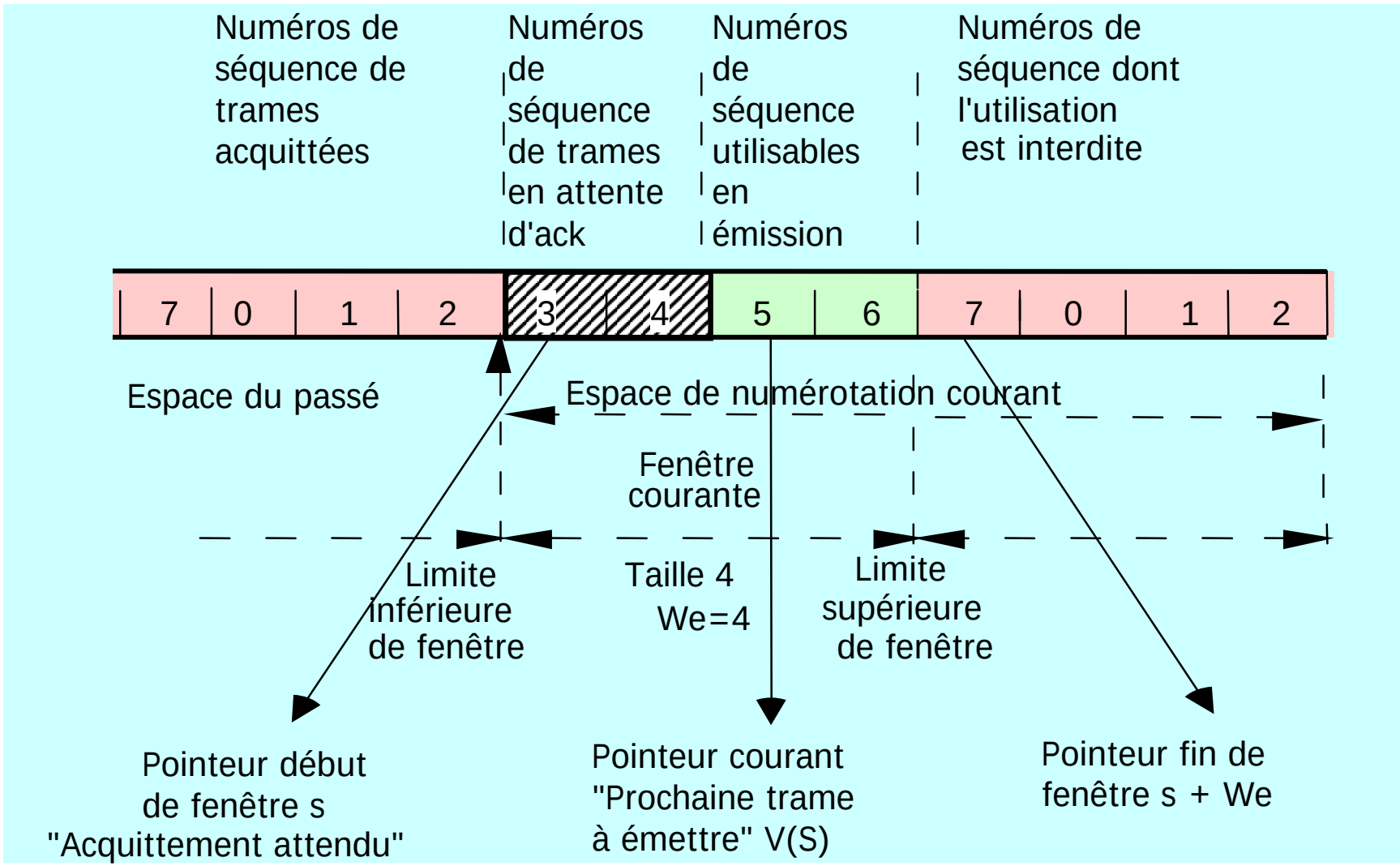


Exemple d'émission avec anticipation, acquits insérés et regroupés

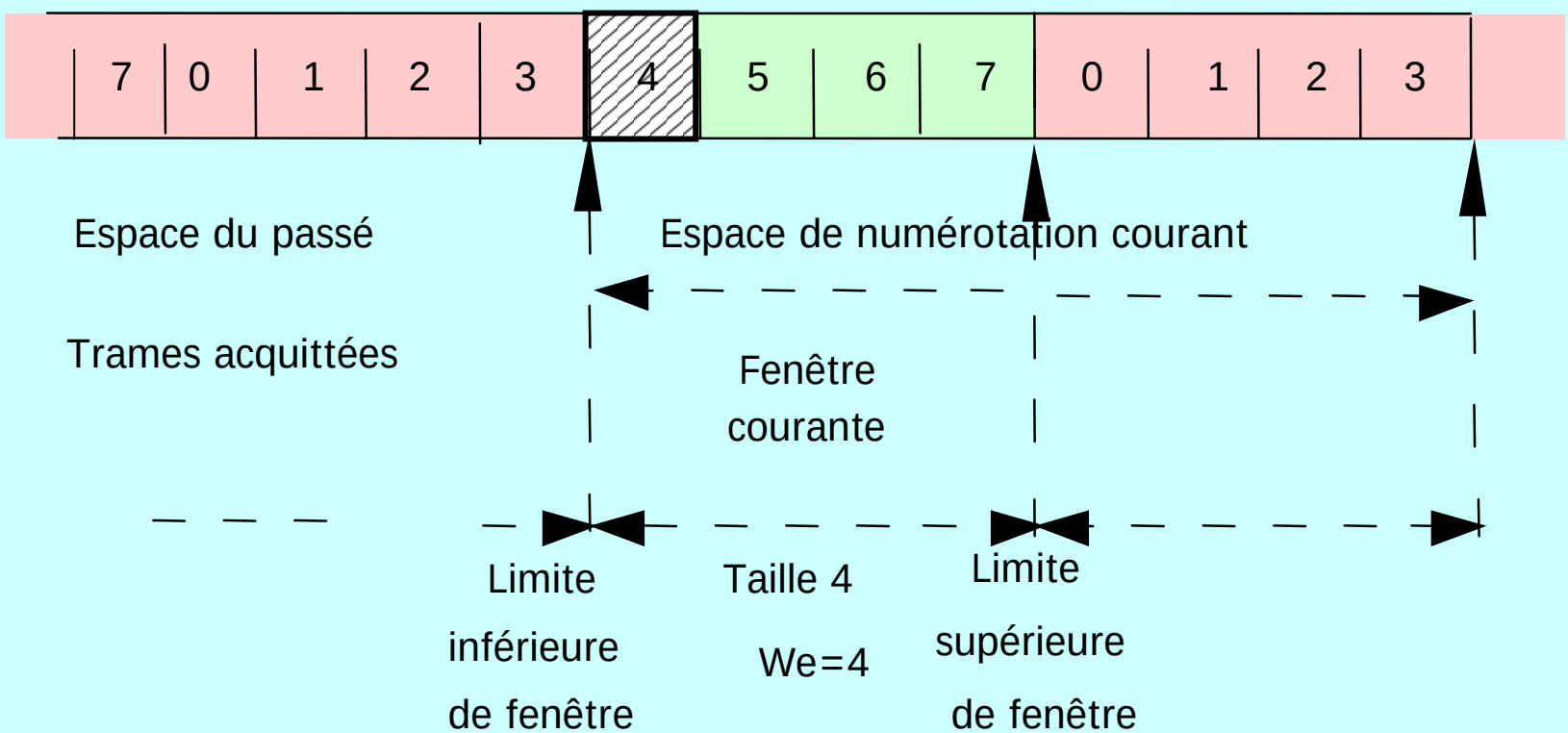
Réalisation des améliorations : Notion de fenêtre d'émission

- **La fenêtre d'émission** ("Sender's Window") est l'ensemble des numéros de séquence des trames dont l'émission en anticipation est autorisée.
  - Trames d'information déjà émises et en attente d'acquittement.
  - Trames à émettre prochainement.
- **Définie par** : **s =< N(S) < s+ We**:
  - **s** est le numéro de la **plus ancienne trame non acquittée**, qui est la limite inférieure de la fenêtre.
  - **s+We** est la limite supérieure de la fenêtre, qui est le **numéro de la première trame** dont l'envoi est **interdit**.
  - **We crédit maximum constant:** taille max de la fenêtre en émission**.**
- **Quand une (ou plusieurs) trames sont acquittées** la fenêtre d'émission glisse circulairement vers le haut.
  - Notion de protocole à **fenêtre glissante** (ou coulissante) (**"Sliding Windows Protocols")**

Protocoles à fenêtres glissantes : Exemple de fenêtre d'émission



Protocoles à fenêtres glissantes : exemple de glissement après RR(4)



Réalisation des améliorations : Notion de fenêtre de réception (1)

- **Fenêtre de réception** ("Receiver's Window") : l'ensemble des numéros de séquence des trames que le récepteur est **autorisé à recevoir**.

  => Toute trame dont le numéro de séquence correspond à un numéro de la fenêtre de réception est **acceptée**.

  => Toute trame dont le numéro de séquence est à l'extérieur de la fenêtre de réception est **détruite**.

Réalisation des améliorations : Notion de fenêtre de réception (2)

- Une **trame reçue correctement et dont le numéro de séquence correspond au niveau bas** de la fenêtre en réception:

  => peut-être **délivrée à l'utilisateur** car elle est en séquence (respect de l'ordre d'émission),

  => La fenêtre en réception peut **glisser d'une unité** vers le haut,

  => La trame peut-être **acquittée** vis à vis de l'émetteur,

- Ces opérations sont réalisées **de façon plus ou moins rapide** sans incidence sur le fonctionnement correct du protocole.
- La fenêtre d'émission et la fenêtre de réception **peuvent être de tailles différentes.**

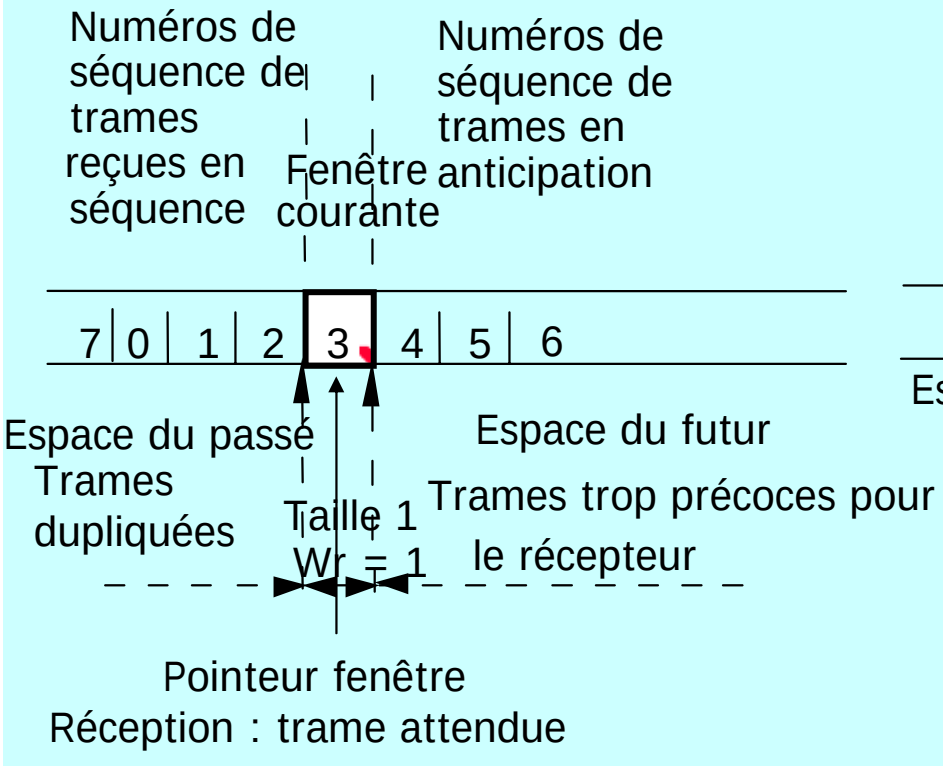
Fenêtre de réception dans le protocole 4

- **Taille de la fenêtre en réception : 1**
  - Le récepteur est donc obligé de **recevoir** les trames correctement les unes après les autres **exactement dans l'ordre d'émission** (un seul tampon suffit).
  - Quand une trame est en **erreur** le récepteur (qui persiste a ne vouloir qu'une seule trame) **perd toute la série** de trames émises en anticipation par l'émetteur :

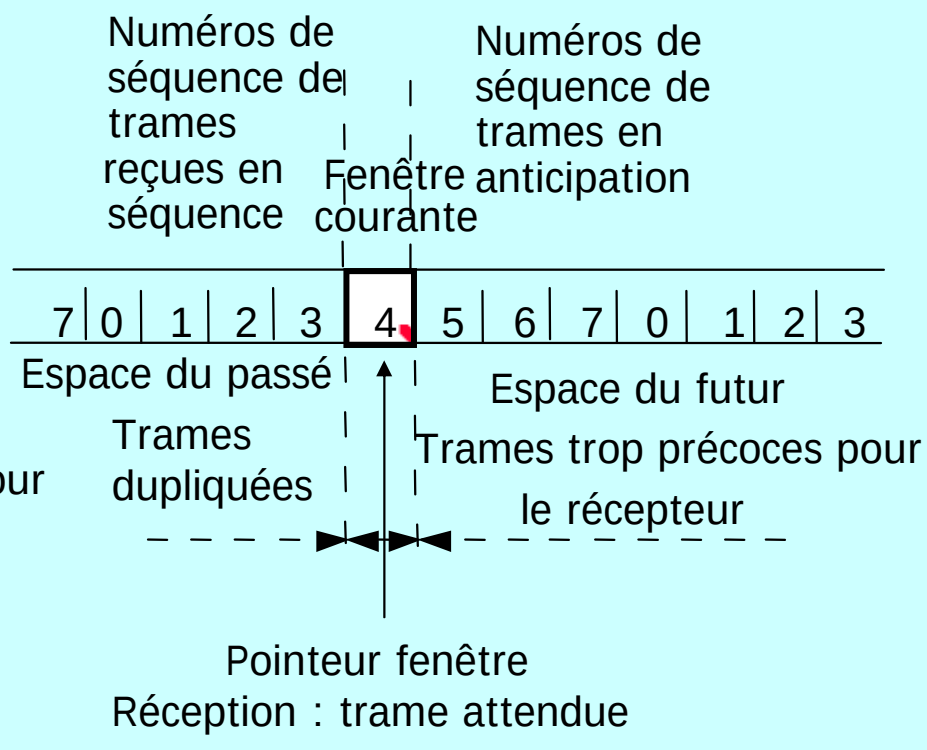
    => Effort d'anticipation perdu (optimisation à venir protocole 5)
- **Stratégie de reprise sur erreur**
  - Stratégie de délai de garde et acquittement positif
  - Stratégie d'acquittement négatif
- **Technique retour arrière de n : "Go back n"**
  - Lorsque le récepteur **constate une lacune** dans la séquence des trames et il **demande la retransmission** de toutes les trames non acquittées avec : N(S) > N(R)-1

Exemple de fenêtre en réception dans le protocole 4

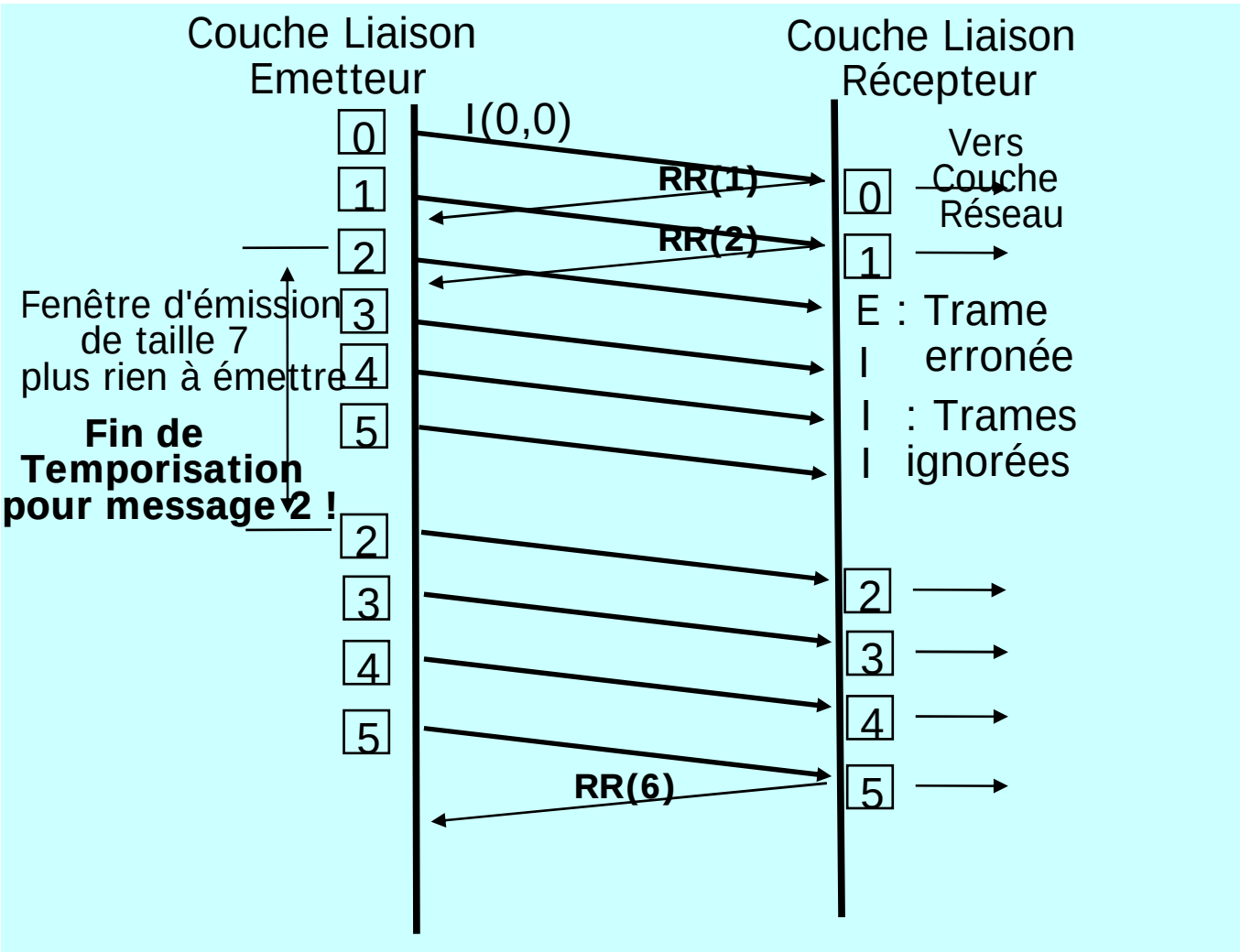
Fenêtre en réception



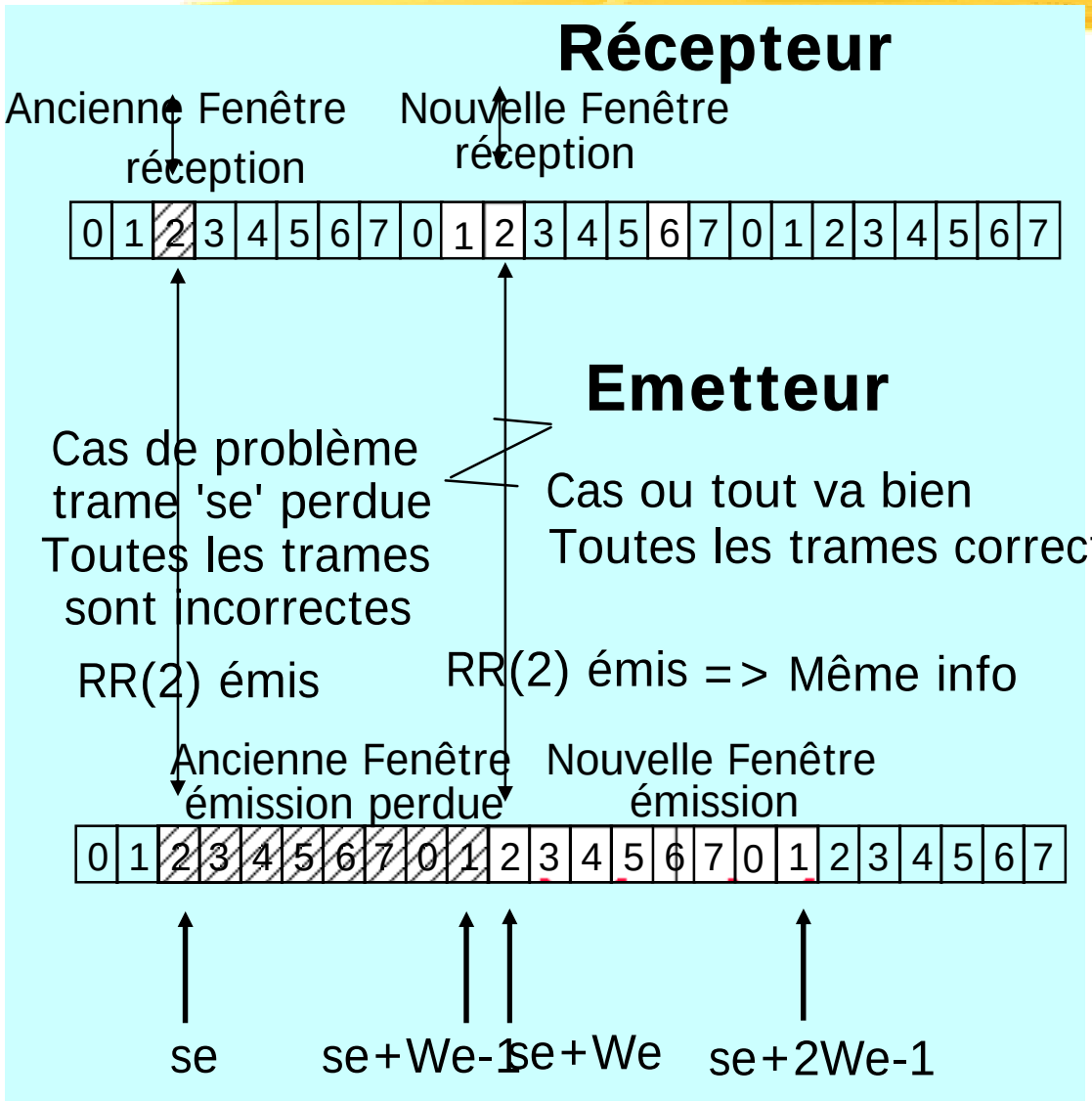
Fenêtre en réception après arrivée de I(3)



Exemple de fonctionnement du protocole 4



Approfondissement 1 : taille maximum de la fenêtre en émission



- **Hypothèse:** 8 numéros de séquence et taille de fenêtre 8
- Si RR(2) se perd pour une fenêtre correcte : retransmission acceptée des trames des trames 2, 3, 4

=> En fait duplication.

- Il y ambiguité quand: se = se + We mod (maxseq +1)

=> **On ne peut pas utiliser une fenêtre comportant la totalité des numéros**.

Approfondissement 1 : formalisation problème de taille maximum (1)

- **Hypothèse : We = Maxseq +1**
- Un émetteur émet toute sa fenêtre: **s < N(S) < s+We-1**
- ***Cas 1 : La première trame s est en erreur***

  Si s est en erreur, le destinataire retourne un acquittement portant N(R) = s => Dans ce cas l'acquittement signifie qu'aucun message n'est reçu correctement (attente de s).
- ***Cas 2 : Toutes les trames sont reçues correctement***

  Le destinataire retourne un acquittement N(R) = s+We

  => Dans ce cas l'acquittement signifie que toutes les trames sont reçues correctement (attente de s+We).

Approfondissement 1 : formalisation problème de taille maximum (2)

- **Pour qu'il n'y ait pas d'ambiguïté possible sur la signification il faut que les deux acquittements:**

  N(R) = s et N(R) = s+We

  soient distinguables (soient des valeurs différentes)
- Si We = Maxseq +1: Pas de distinction entre cas 1 et cas 2

  N(R) = s = s+We = s + Maxseq +1 mod (Maxseq+1)
- **Pour que** les We+1 nombres allant de s à s+We soient tous distincts modulo Maxseq+1 il faut We < Maxseq+1.

  => On doit prendre au plus We = Maxseq
- **Exemples** : n = 3 , 2^n = 8, We=7 trames en anticipation.

  Cas n = 7 , 2^n = 128, We = 127 trames en anticipation.

Approfondissement 2 : Niveaux de contrôle de flux

- **1 Contrôle de flux par fenêtre d'émission (rappel)**

  Avec une fenêtre glissante, si le destinataire s'abstient de retourner des acquittements, il est assuré de ne pas recevoir plus de We trames d'informations s'il retient ses acquittements.

  Problème - Cette technique ne peut plus s'appliquer lorsqu'un site peut obliger l'autre à acquitter (mode commande réponse).

- **2 Suspension temporaire des échanges**

  Permet au destinataire de **demander l'arrêt** temporaire (**puis la reprise**) du protocole.

  Exemple : RNR (non Prêt à recevoir) demande la suspension
  RR ou REJ permettent de reprendre

  Autres exemples : (XOFF, XON) ou (WACK, ENQ) , Ethernet 802.1q

  Ne convient pas si un nombre important de trames peuvent être émises pendant la transmission de la demande d'arrêt.

Protocole 5 : A fenêtre glissante et rejet sélectif

- **Objectif :** Conserver le **bénéfice de l'anticipation** en **cas d'erreur**.
- **Solution :** utiliser une fenêtre de réception de **taille supérieure à 1**

  => Wr>1 définit **la plage des numéros N(S)** de trames d'informations **acceptables par le destinataire**.

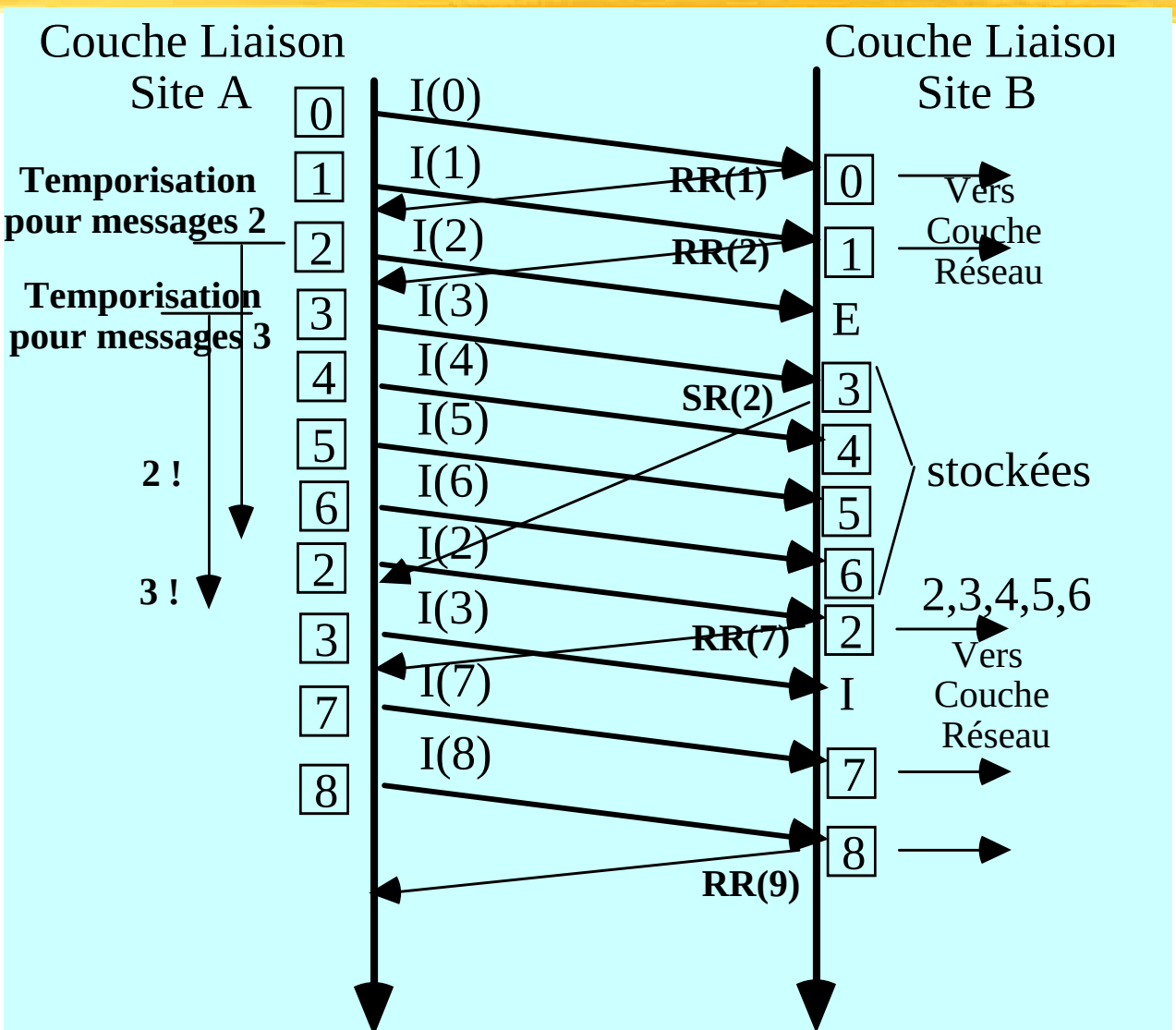
  => Le récepteur accepte **des trames déséquencées** (avec des lacunes dans la numérotation).

  => Le récepteur doit donc **gérer pour chaque trame de la fenêtre un booléen indiquant l'arrivée correcte**.

  => Le récepteur reconstitue la séquence complète des trames émises par **retransmission sur échéance de délai de garde ou sur acquittement négatif**.

Exemple de fonctionnement en mode de rejet sélectif

- L'acquittement négatif est baptisé également ***rejet sélectif*** (SR(N(R) "Selective Reject")
- C'est une **demande de retransmission d'une seule trame d'information en erreur** (de numéro de séquence N(R)).



Approfondissement 1 : Problèmes de taille de fenêtres en rejet sélectif

- **Dimensionnement des fenêtres émission et réception**

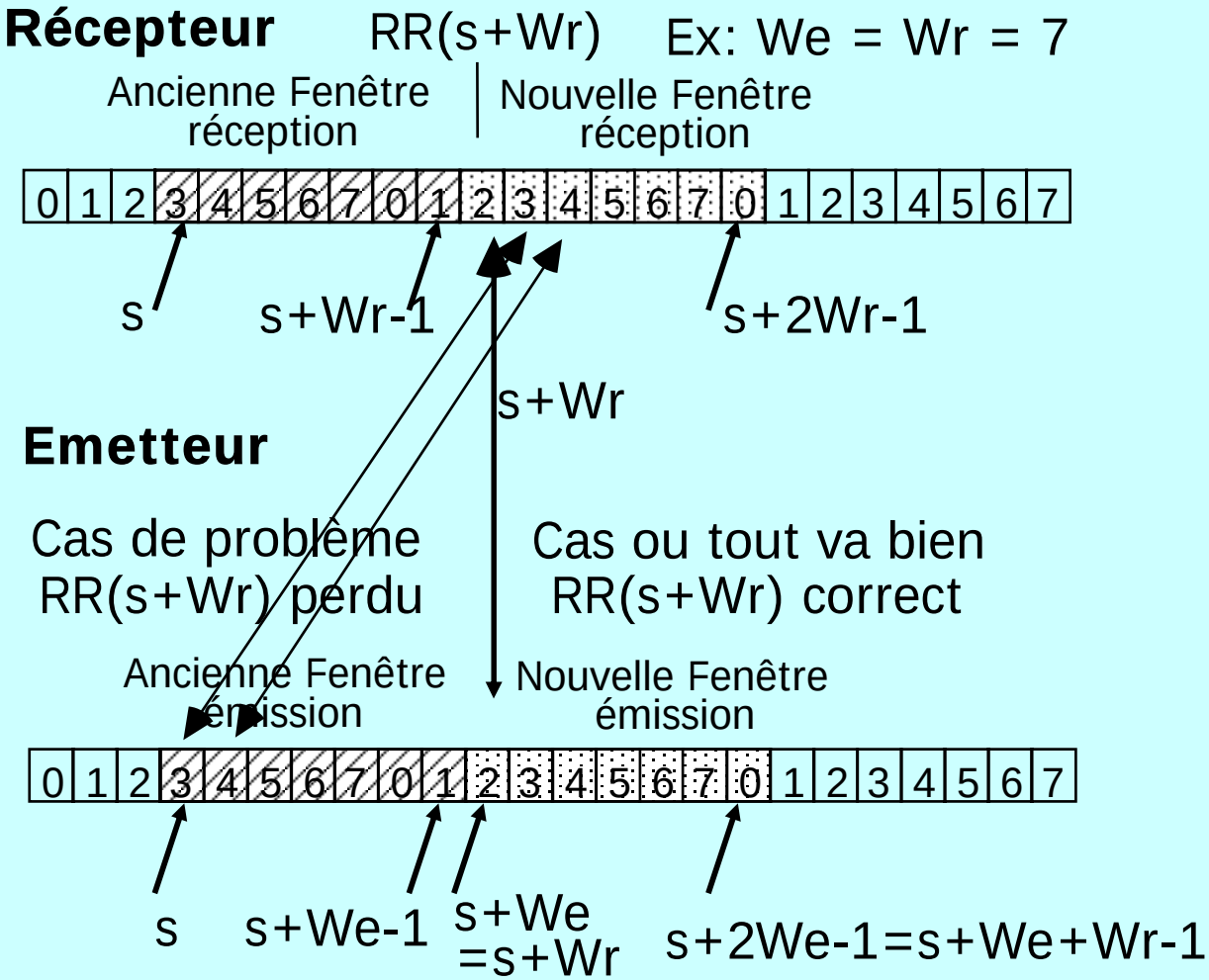
  - We < Wr : pas très utile car Wr-We tampons en réception ne sont jamais utilisés.

  - We > Wr : on ne traite pas complètement le problème de perte du bénéfice d'anticipation en cas d'erreur.

  - We = Wr : choix rationnel.

- **Problème :** le protocole en rejet sélectif fonctionne t'il avec We = Wr = maxseq (la valeur déterminée pour le protocole 4).

Approfondissement 1 : Perte d'un acquittement pour une fenêtre pleine



Retransmissions acceptées des trames 3,4,.. : duplication

Approfondissement 1 : Formalisation du problème de taille en rejet sélectif

- **Si We = Wr = W** : il y ambiguïté entre deux fenêtres d'émission successives quand

  s = s + 2*We -1 mod (maxseq +1)

  => W = maxseq/2 +1

  => On ne peut pas utiliser plus de la moitié des numéros de séquence disponibles.

- **Relation plus générale** : si We différent de Wr

  s = s + We + Wr -1 mod (maxseq +1)

  We + Wr - 1 = maxseq +1

  Ambiguïté si We + Wr = maxseq +2

Approfondissement 2 : Gestion de temporisateurs multiples

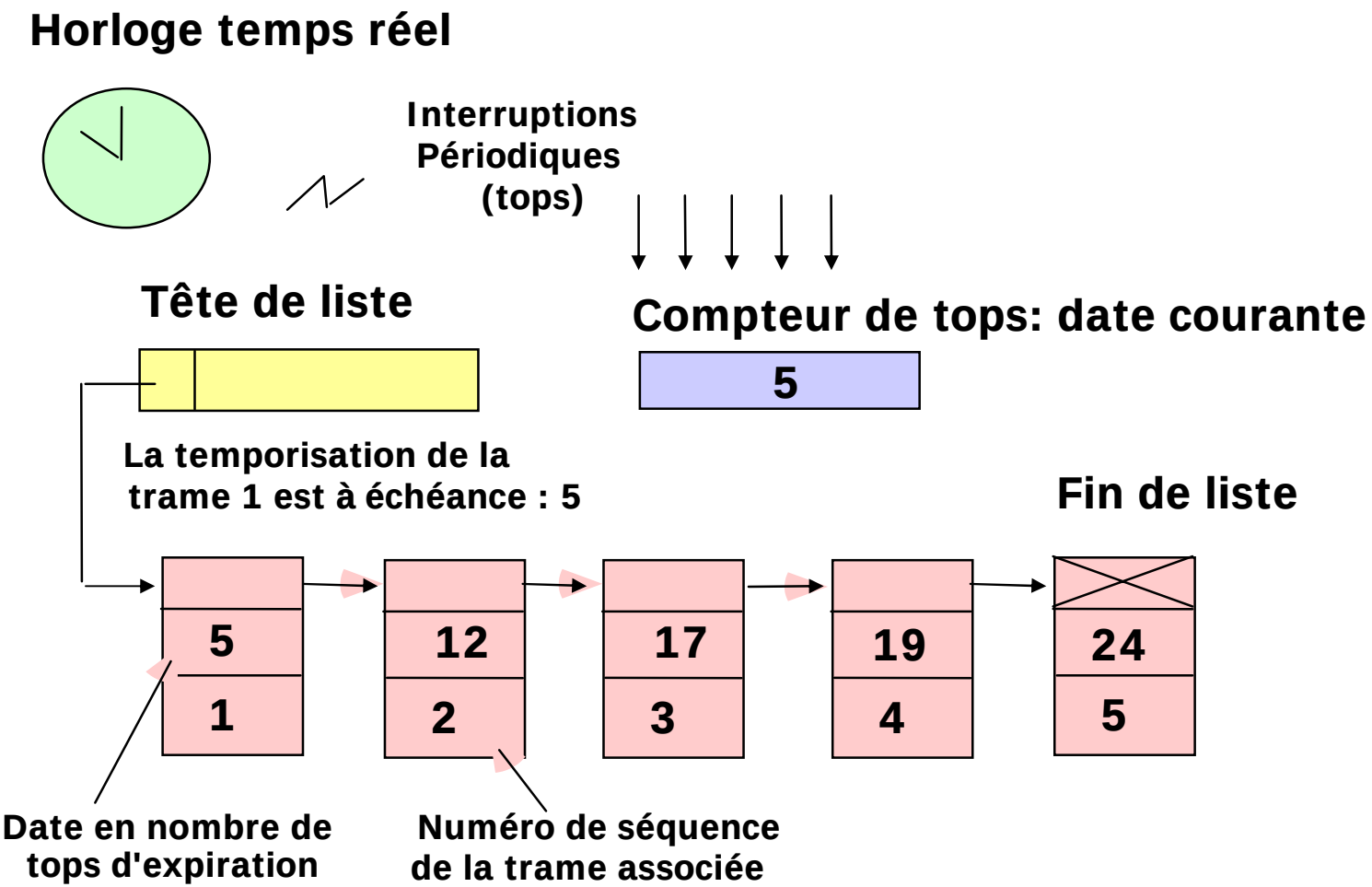
- **Les protocoles à fenêtres glissantes à rejet sélectif impliquent une temporisation associée à chaque message**.
- **Gestion d'un échéancier** (une liste chaînée par ordre croissant de dates d'échéances).
- **Armement d'un temporisateur ("Start_timer")**

  . Insertion d'événements à générer dans la liste.
- **Désarmement d'un temporisateur ("Stop_timer")**

  . Retrait de la liste de l'événement associé
- **Échéance d'un temporisateur ("Alarm")**

  . Déclenchement de traitement de temporisateur sur arrivée en échéance du temporisateur de la tête de liste.

Approfondissement 2 : Schéma avec temporisateurs multiples



Conclusion : Solutions générales aux problèmes des protocoles de liaison

- **Existence de solutions satisfaisantes** construites pour des voies physiques bruitées à bas débit.

  . Solution au problème du **contrôle d'erreur, de contrôle de flux, de contrôle de séquence.**
- **Réutilisation** de ces solutions dans les réseaux sans fils (bruités et d'un débit pas très élevé).
- **Renouvellement du problème** :

  A) Liaison sur câble **coaxial ou fibre** optique : **faible bruit et haut débit.**

  B) Les applications visées sont de plus en plus multimédia: besoin de solutions de communication à **qualité de service**

  => La couches liaison traite **la délimitation, le multiplexage, l'administration** mais **pas de contrôle d'erreur ni de flux** (reportés au niveau transport)

Niveau Liaison En Point à point

Chapitre II

Protocoles industriels

II.1 Protocole à trames de bits

II.2 Protocole PPP

Protocoles de liaison en point à point : Exemples industriels

II.1
Protocoles à trames de bits

Introduction : Rappel des protocoles synchrones en mode caractère

- **Les premiers protocoles de liaison implantés** (définis avant 1970).
- **Protocoles en mode caractère BSC "Binary Synchronous Communication"**
  - L'unité d'information **est le caractère**.
  - Certains caractères sont réservés aux besoins du protocole : les ***caractères de contrôle subsistent dans le jeu de caractères ASCII)***
  - Il a existé **de multiples versions de protocoles basées sur des principes** et une utilisation des caractères de contrôle souvent **très voisins**.
    - Exemple : Protocole BSC 3780

Evolution vers les protocoles à trames de bits

- **Volonté de construire des protocoles indépendants d'un jeu de caractères**.
  - Si l'on transporte des caractères 10 bits ou des mots 60 bits **il faut découper et coder** les données au format caractère du lien.
  - Il faut traiter à chaque fois le problème de l**a transparence en fonction** du code caractère et traiter spécifiquement les caractères de contrôle.
- **Conséquence pour les protocoles à trames de bits:**
  - a) La charge utile devient **une suite de bits**.
  - b) Abandon des caractères de contrôle et **définition d'un format de trame au moyen de zones** ayant un rôle et un codage binaire (structure de message).

Protocoles à trames de bits : Historique et normes associées (1)

- IBM a développé vers 1970 **SDLC** ("**Synchronous Data Link Communication**") pour SNA.
- SDLC a été soumis à l'ISO pour normalisation. Modifié il est devenu **HDLC "High-level Data Link Communication".**
- SDLC a été soumis à l'ANSI pour devenir le standard américain qui l'a modifié et est devenu **ADCCP ("Advanced Data Communication Control Program")**
- Le CCITT a adopté et modifié HDLC qui est ainsi devenu le **LAP ("Linkage Access Protocol")** pour le standard de réseau **X25**.
- Le CCITT a modifié X25 dont le LAP qui est devenu le **LAPB** ("**Linkage Access Protocol**" **B** ou Balanced)

Protocoles à trames de bits : historique et normes associées (2)

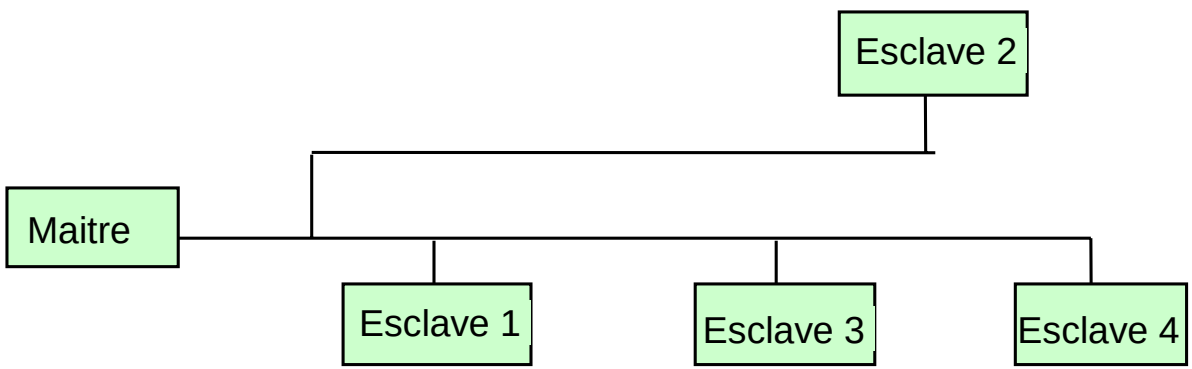
- Les IEEE ont normalisé comme l'un des standards de liaison sur les réseaux locaux une version modifiée légèrement : le **LLC2 ("Logical Link Control type 2")**
- Une autre version : le **LAPD ("Linkage Access Protocol on the D channel")** est définie pour servir de protocole de liaison sur les canaux D du RNIS.
- Une version **LAPX** est spécifiée pour fonctionner avec des échanges à l'alternat (**LAP semi-duplex**).
- Une version **LAPM** (‘**LAP Modem**’) est spécifiée pour la corrections d’erreurs dans les modems.
- Le standard Internet **PPP (‘Point to Point Protocol**’) en version de base reprend différents idées des protocoles à trames et possède une version avec contrôle d’erreur qui est un version de LAPB.

Protocoles à trames de bits : Principes généraux

- **Nombreux points communs** : protocoles à fenêtres.
- Mais **variantes de détail**, liées à des contextes d'utilisation qui rendent ces protocoles incompatibles
- **Quelques choix communs**
  - Utilisation du **principe d'anticipation.**
  - **Numéros de séquence**
    - En général sur 3 bits (au maximum 7 trames en anticipation).
    - Variantes sur 7 bits (exemple LAPD).
  - **Regroupement des acquittements.**
  - **Acquittements insérés** ("piggybacking").
  - Choix dans la plupart des cas **d'une fenêtre en réception de taille 1** (sauf HDLC, ADCCP).

Protocoles à trames de bits : Rappel voies multipoint

- **Deux modes** de fonctionnement :
  - **Symétrique (‘Balanced’)** : point à point.
  - **Dissymétrique (‘Unbalanced’) :** en **scrutation** maître esclave (mode multipoint) **("polling- selecting").**
- **Organisation arborescente (en grappe) des voies multipoint**
  - **Site maître** (calculateur): qui dispose de toutes les fonctions (autre terme: primaire) et supervise les communications.
  - **Sites esclaves** (terminaux): qui ne disposent pas de certaines fonctions (autre terme: secondaire) et qui obéissent au maître.



Protocoles à trames de bits : la gestion des voies multipoint

- **Une classification des fonctions** est effectuée : **fonctions primaires et fonctions secondaires (**pour caractériser les modes maître et esclave).

- Sont considérées comme **primaires :**

  - Mise **en ligne ou hors ligne** (mise d'une station en l'état d'émettre ou de recevoir).

  - **L'initiative dans la validation des données** (sollicitation des acquittements).

  - Les traitements en **cas d'erreur de protocole**.

- Sont considérées comme **secondaires :**

  - **L'initiative d'émission de données** (une fois le mode d'échange établi)

  - ... **les autres fonctions**

Protocoles à trames de bits : les différents modes d'échange

Distinction dans les modes d'utilisation des fonctions primaires et secondaires

- **Mode normal de réponse NRM** ("Normal Response Mode") : **Configurations dissymétriques**

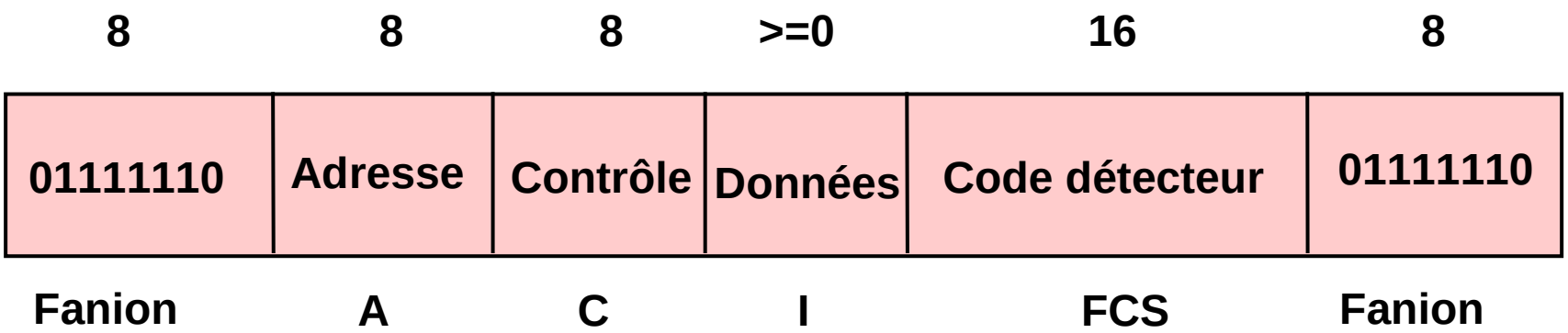
  - L'initiative de l'attribution du mode d'échange est réservé uniquement à la station primaire.

- **Mode asynchrone ABM** ("Asynchronous Balanced Mode") : **Configurations symétriques**

  - Mode asynchrone symétrique applicable aux liaisons point à point en réseau..

  - Une station recevant une commande doit y répondre immédiatement (voir plus loin bit P/F).

Protocoles à trames de bits : la structure des trames (1)



- **1) Les fanions et la transparence binaire (bit stuffing)**

  - Une méthode pour délimiter les trames qui préserve **les propriétés de transparence** des données utilisateur.

  - Chaque trame **commence et se termine par la chaîne 01111110** (Terminologie Drapeau, fanion ou flag)

  - A l'émission **quand on détecte une donnée avec une de suite de 5 bits 1** consécutifs on insère un 0, **en réception le bit 0 suivant 5 bit 1** est automatiquement enlevé.

  - Un **fanion** délimiteur 01111110 est donc **toujours considéré comme tel**.

Protocoles à trames de bits : la structure des trames (2)

- **2) Le champ adresse** (A)

  Permet de traiter les voies multipoint (adresse d'une station secondaire).

  Pour les voies point à point (on distingue les commandes des réponses).

- **3) Le champ contrôle (C)** contient :
  - Le **type** de la trame.
  - Le **numéro de séquence**
  - Les **acquittements**.
  - Le **bit de commande réponse.**

- **4) Le champ données (I)**

  Il peut être arbitrairement long (à traiter en liaison avec le CRC dont l'efficacité décroît en raison de la probabilité d'erreurs en rafale).

- **5) Le champ détection d'erreur (FCS "Field Check sequence") .**

  Généré par le polynôme du CCITT: X16 + X12 + X5 + 1

  En LAPB une variante est appliquée pour détecter les pertes de fanions.

Protocoles à trames de bits : les trois catégories de trames

- **1. Information :Trame I ("Information")**

  Transportent des informations significatives

- **2. Supervision : Trame S ("Supervision")**

  Utilisées pour superviser les échanges de trames I.

  Exemples : Envoyer un acquittement explicite

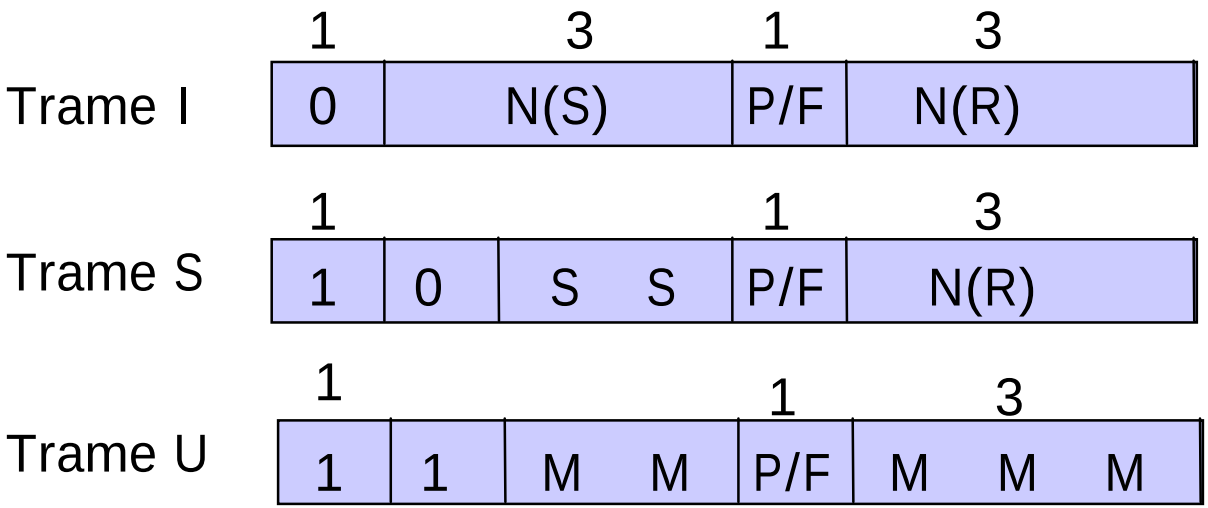
  Demander une suspension temporaire

- **3. Gestion : Trame U ("Unnumbered")**

  Assurent les fonctions nécessaires avant et après l'échange des données.

  Exemples : connexion, déconnexion d'une station, traitements d'erreurs de protocole.

Protocoles à trames de bits : les trois catégories de trames



- **N(S)** : numéro de séquence en émission
- **N(R)** : numéro de séquence de la prochaine trame non encore reçue.
- **S** : type de la fonction de supervision
- **M** : type de la trame non numérotée

Le Bit P/F : Scrutation/Fin d'émission Commande/Réponse ("Poll/final")

- **Première Signification en mode normal de réponse : Invitation à émettre ou fin d'émission** (un primaire gère un groupe de secondaires):

  - Dans les trames de commande le bit à 1 **noté P** signifie "invitation pour la station adressée à émettre (**polling**)".

  - Dans les trames de réponse en provenance de stations secondaires ce bit à 1 **noté F** signifie **fin** de transmission.

- **Seconde Signification** en mode asynchrone équilibré **: Commande Réponse.**

  - La station A recevant une trame avec le bit P l'interprète comme une **commande** émise par le primaire distant B.

  Exemple type: commande d'acquittement immédiat à destination du secondaire local.

  - A doit répondre immédiatement à la commande par **une trame de réponse avec le bit F positionné** car le site distant B a armé un délai de garde pour retransmettre.

Problème du bit P/F en mode symétrique

- **Les deux stations possèdent** :

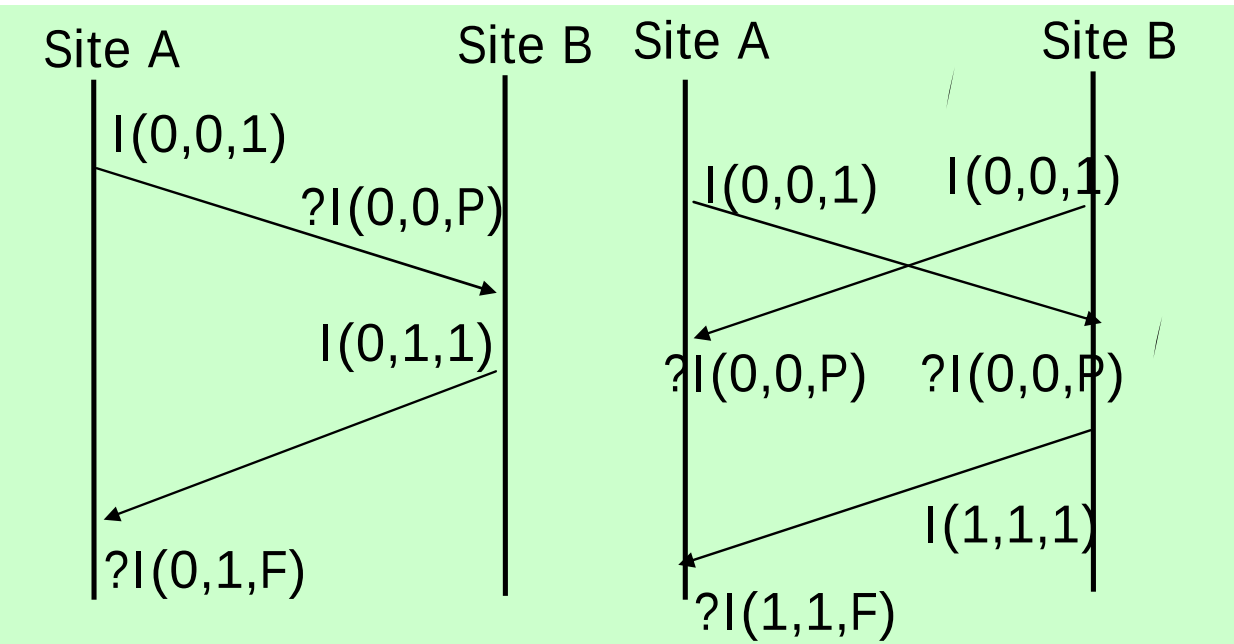
- les fonctions primaires (émettent des trames de commande avec bit P).
- les fonctions secondaires (émettent des trames de réponses avec bit F).

Mais un seul bit pour deux significations

- Une station A **recevant un bit P/F** ?

1- **C'est un bit P** que B a émis par ses fonctions primaires.

2- **C'est un bit F** en réponse à un bit P que A avait envoyé avant.



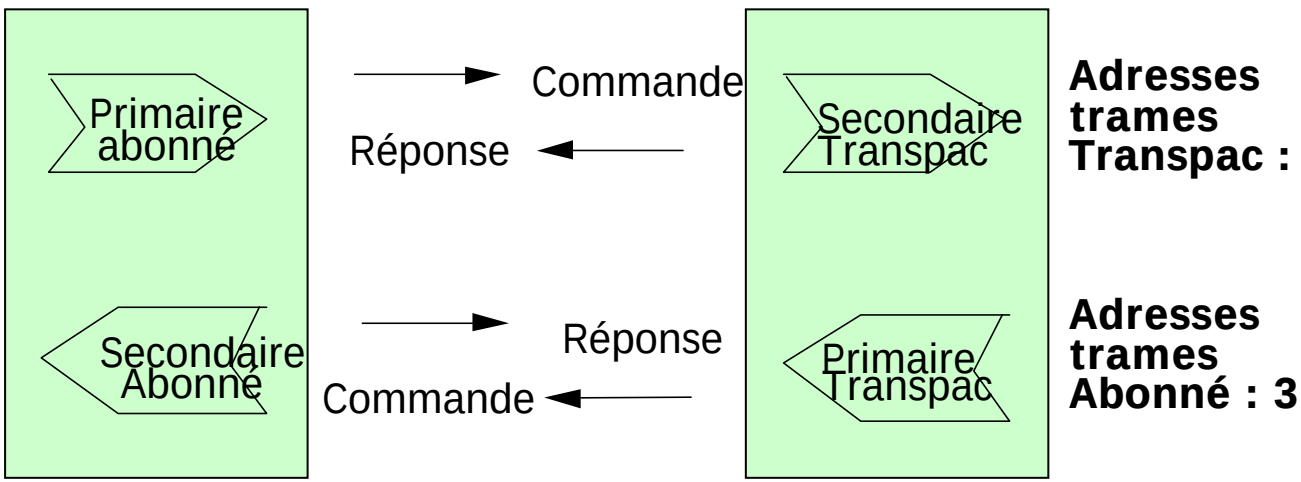
Solution bit P/F en mode symétrique : disposer de deux bits

- 1) En LAPB on ne dispose pas de deux bits dans l'octet de contrôle.

  On conserve la structure du champ contrôle => **On utilise l'adresse**.

  - Si une station agit comme primaire elle place l'adresse de la station éloignée (cas d'une initiative d'émission, bit P)

  - Si une station agit en secondaire elle place dans les trames son adresse (cas d'une réponse bit F).



- 2) Dans les protocoles LAPD, Relais de trame on a **défini** le format des trames pour faire place à un bit de plus C/R (commande/réponse).

Les quatre types de trames de supervision (1)

- **Type 0 - RR Prêt à recevoir ("Receiver Ready")**

  - Station prête à recevoir des trames I.

  - Permet d'accuser réception des trames dont le numéro de séquence est inférieur ou égal à N(R)-1.

  => Trame utilisée lorsqu'il n'y a pas suffisamment de trafic dans un sens pour transporter les acquittements.

- **Type 1 - REJ Rejet – ("Reject")**

  - Acquittement négatif (protocole 4).

  - Le champ N(R) désigne la première trame en séquence qui n'a pas été reçue.

  =>L'émetteur doit réémettre toutes les trames depuis N(R).

  - Toutes les trames jusqu'à N(R)-1 sont acquittées.

Les quatre types de trames de supervision (2)

- **Type 2 RNR- Non prêt à recevoir "Receiver not ready"**

  - Indique une incapacité temporaire à accepter les trames d'informations suivantes (en cas de saturation des tampons par exemple).

  - Mécanisme de contrôle de flux plus fort que celui de la fenêtre (un état d'exception) qui finit avec RR ou REJ.

- **Type 3 SR - Demande de retransmission sélective "Selective Reject"**

  - Demande de retransmission de la seule trame dont le numéro est contenu dans N(R)

    Non applicable au LAPB et SDLC.

    Applicable à HDLC et ADCCP.

Trames non numérotées (type U) : Ouverture de connexion

- **Distinction des modes d'échange** en ouverture:

  . Mode normal de réponse **NRM**

  . Mode symétrique asynchrone **ABM**

- **Distinction de deux formats de trames :**

  . **Format standard** : Champ commande sur 8 bits (n° séquence 3 bits).

  . **Format étendu** : Champ commande sur 16 bits (n° séquence 7 bits).

- **Demande à une station éloignée de se mettre en ligne dans un mode d'échange**

  . SNRM : Mise en mode normal de réponse standard ("Set Normal Response Mode")

  . SNRME: Mise en mode normal de réponse étendu ("Set Normal Response Mode Extended")

  . SABM : Mise en mode asynchrone symétrique standard ("Set Asynchronous Balanced Mode")

  . SABME : Mise en mode asynchrone symétrique étendu ("Set Asynchronous Balanced Mode Extended")

Trames non numérotées (type U) : Autres trames de gestion de connexion

- **DISC** : Déconnexion ("Disconnect")

  . Demande de fin du mode opérationnel (par exemple dans le cas où une station suspend son fonctionnement pour une opération de maintenance).

- **UA** : Réponse d'accusé de réception non numéroté ("Unnumbered Acknowledge")

  . Acceptation par laquelle une station notifie son accord à une commande non numérotée (commande U)

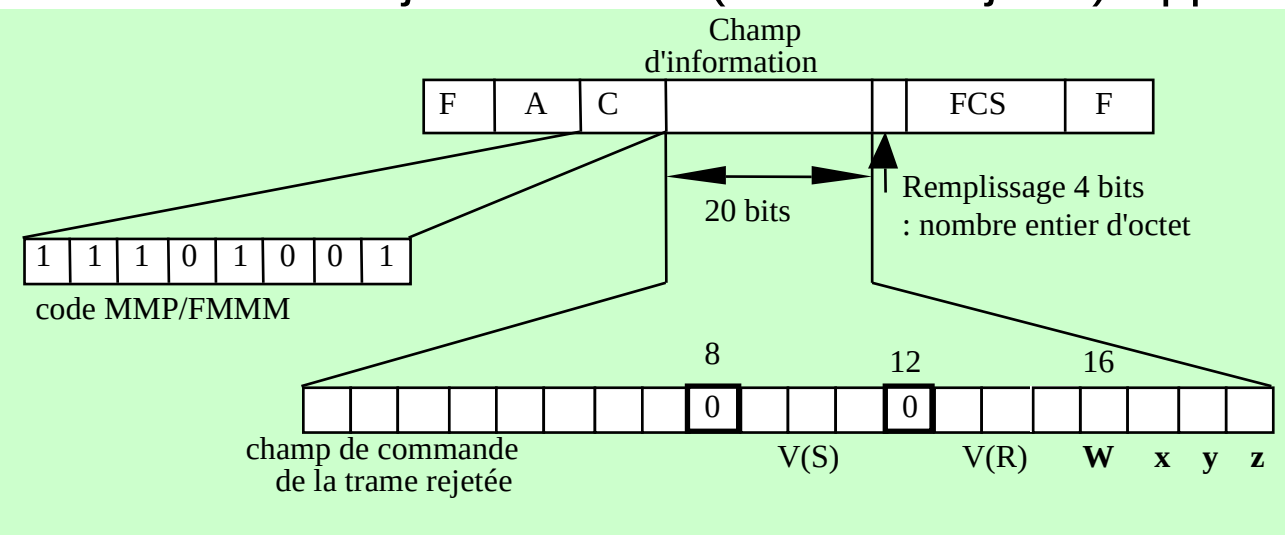
  . Il n'y a pas d'autre champ utilisé

- **DM** : Réponse mode déconnecté ("Disconnect Mode")

  . Indique que la station n'est pas en ligne et ne peut pas accepter de trame.

Trames non numérotées : Trames de rapport d'erreurs protocolaires

- **Trames qui indiquent qu'une condition d'erreur ne peut être corrigée** par retransmission car la trame est incorrecte sémantiquement (erreur de protocole).
- **FRMR**: Rejet de trame ("FRaMe Reject") Applicable au LAPB.



V(S) : Numéro d'émission en cours à la station secondaire.

V(R) : Numéro de réception en cours à la station secondaire.

bit W - Commande invalide : Exemple : une trame de supervision de type 3 demande de répétition sélective n'est pas admise en LAPB.
bit x : Présence indue d'un champ d'information (les trames S et U n'ont pas de champ d'information en général sauf FRMR et CMDR).
bit y : Champ d'information de la trame reçue trop grand pour une station secondaire (ou trop petit). Exemple : Trame de moins de 32 bits interdite en HDLC.
bit z : Numéro de séquence incorrect - accusé de réception d'une trame non émise.

Trames non numérotées (type U) : Autres trames

- **UI** : Trame de gestion d'information non numérotées ("Unnumbered Information")

  Pour les protocoles sans connexion, un seul type de trame UI (applicable à PPP).

  Pour échanger des informations protocolaires.

  Exemple : obtention dynamique d'une adresse (LAPD)

- **XID** : Trame d'identification ("eXchange IDentifier")

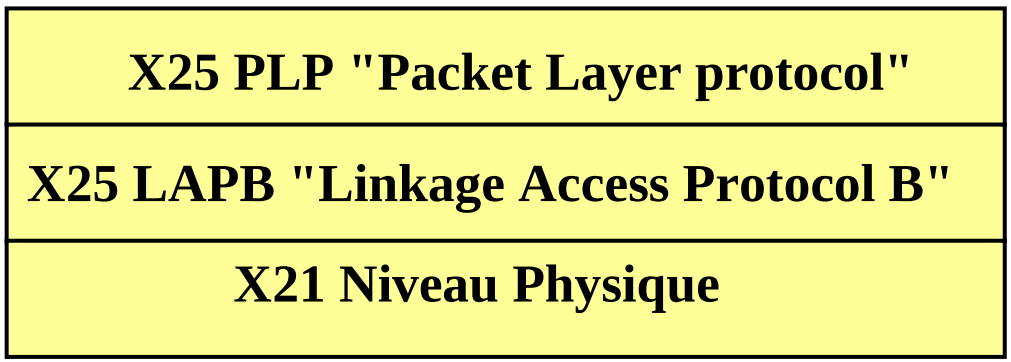
  Pour échanger **les identificateurs** de stations.

  Cette trame peut comporter un champ d'information géré par le niveau supérieur

  (Applicable au LAPD)

Protocoles à trames de bits : Le protocole LAPB

- **Rappel de la situation du protocole** (niveau liaison dans les réseaux publics "Transpac").



- **Résumé des trames utilisées en LAPB**

SABM : Ouverture de connexion.
UA : Acquittement non numéroté.
DISC : Fermeture de connexion.
DM : Indication de mode déconnecté.
FRMR : Erreur de protocole.
I : Trame d'information.
RR : Acquittement explicite.
RNR : Suspension temporaire.
REJ : Rejet d'une suite de trames I.

Protocoles à trames de bits : Le protocole LAPD

- **Rappel de la situation du protocole :** niveau liaison pour le canal D dans le réseau numérique à intégration de services (RNIS)

X25 PLP	**Q930-I451 ProtocoleD**
Q921-I441 LAPD	
I431 "Interfaces S/T"	

- **Principes généraux du protocole LAPD**

- En mode connecté

Echange de trames numérotées I en mode asynchrone équilibré étendu (SABME).

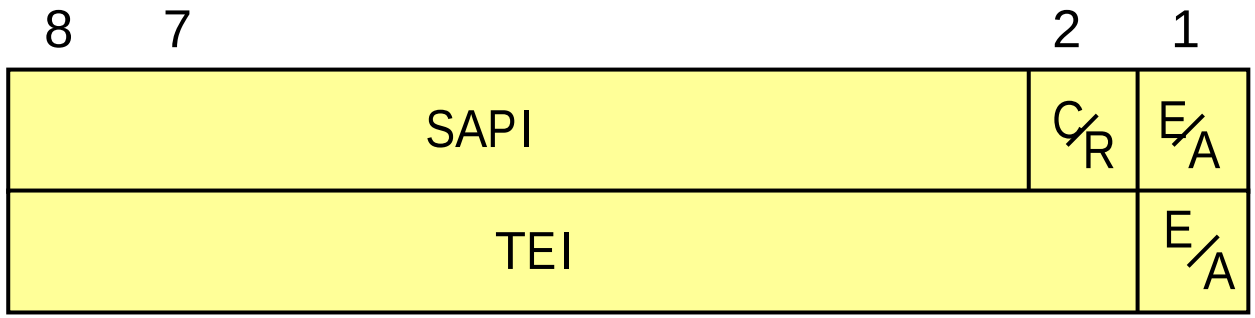
Amélioration des possibilités d'adressage: affectation dynamique d'adresse (XID).

En cas de violation de protocole il y a réinitialisation complète sans notification de cause (pas de FRMR).

- En mode non connecté

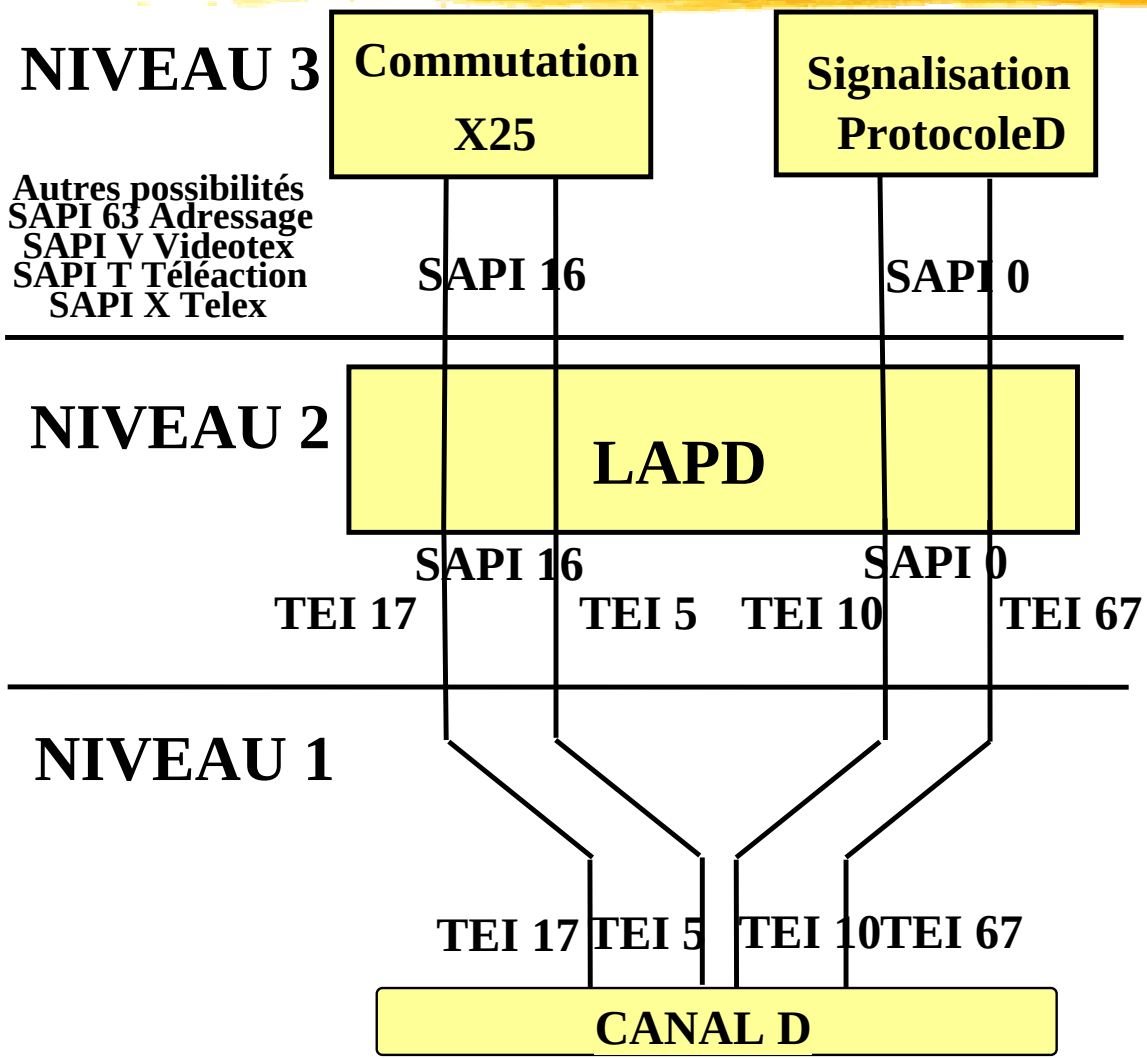
Echange de trames d'informations non numérotées **UI** sans correction d'erreur ni contrôle de flux.

Protocole LAPD : Format des adresses



- **E/A :** Bits d'extension d'adressage.
- **C/R :** Bit distinguant les commandes des réponses.
- **SAPI : 'Service Access Point Identifier'** Permet le multiplexage de différents flux de réseau sur une liaison.
- **TEI : 'Terminal End-Point Identifier'** Adresse de l'appareil physique (une appareil peut utiliser plusieurs adresses).

Protocoles à trames de bits : Adressage en LAPD



LAPD : Gestion des identificateurs de terminaux

- **a) Affectation statique**
  - Générée par le fabricant de 0 à 63.
  - Problème en cas de configurations conflictuelles.
- **b) Affectation dynamique**
  - Existence d'une fonction de la couche liaison permettant de demander un TEI à la mise sous tension d'un terminal.
  - TEI non déjà affecté entre 64 et 126.
  - Utilisation de trames UI typées de SAPI 63 TEI 127 pour le protocole d'attribution d'adresse.
    - Demande d'identité
    - Identité refusée
    - Vérification d'identité
    - Réponse à la vérification
    - Suppression d'identité.
    - Demande de vérification.

LAPD : Résumé des trames utilisées

- **SABME** : Ouverture de connexion mode étendu.
- **UA** : Acquittement non numéroté.
- **DISC** : Fermeture de connexion.
- **DM** : Indication de mode déconnecté.
- **XID** : Echange d'identificateur.
- **UI** : Information non numérotée.
- **I** : Trame d'information.
- **RR** : Acquittement explicite.
- **RNR** : Suspension temporaire.
- **REJ** : Rejet d'une suite de trames information.

II.2
Protocole PPP
("Point to Point Protocol")

1 Généralités
2 La transmission de données
3 La configuration de liaison
4 La configuration de réseau
5 La compression d'entêtes

Protocole PPP
("Point to Point Protocol")

II.2.1
Généralités

A) Historique
B) Objectifs généraux
C) Architectures
D) Organisation du protocole

A) Historique des protocoles de liaison point à point avec IP

- **IP** (1980) : défini pour l'interconnexion de **réseaux**.
- **Besoin d'une adaptation** des paquets IP aux différents réseaux visés: IP sur réseaux locaux (Ethernet, ...)
  - IP sur réseaux longues distance (X25)
- Besoin d'un **protocole de liaison point à point** pour acheminer les paquets IP sur voies séries.
  - **Solutions propriétaires** très simples (début 1980).
  - **Normalisation minimum: SLIP** (1984,RFC en 1988)
  - Création d'un groupe de travail IETF **pour une solution complète : PPP** (RFC 1134 nov1989)
- Améliorations successives de PPP : RFC complémentaires.
  - Version en cours **RFC 1661** (juillet 1994)

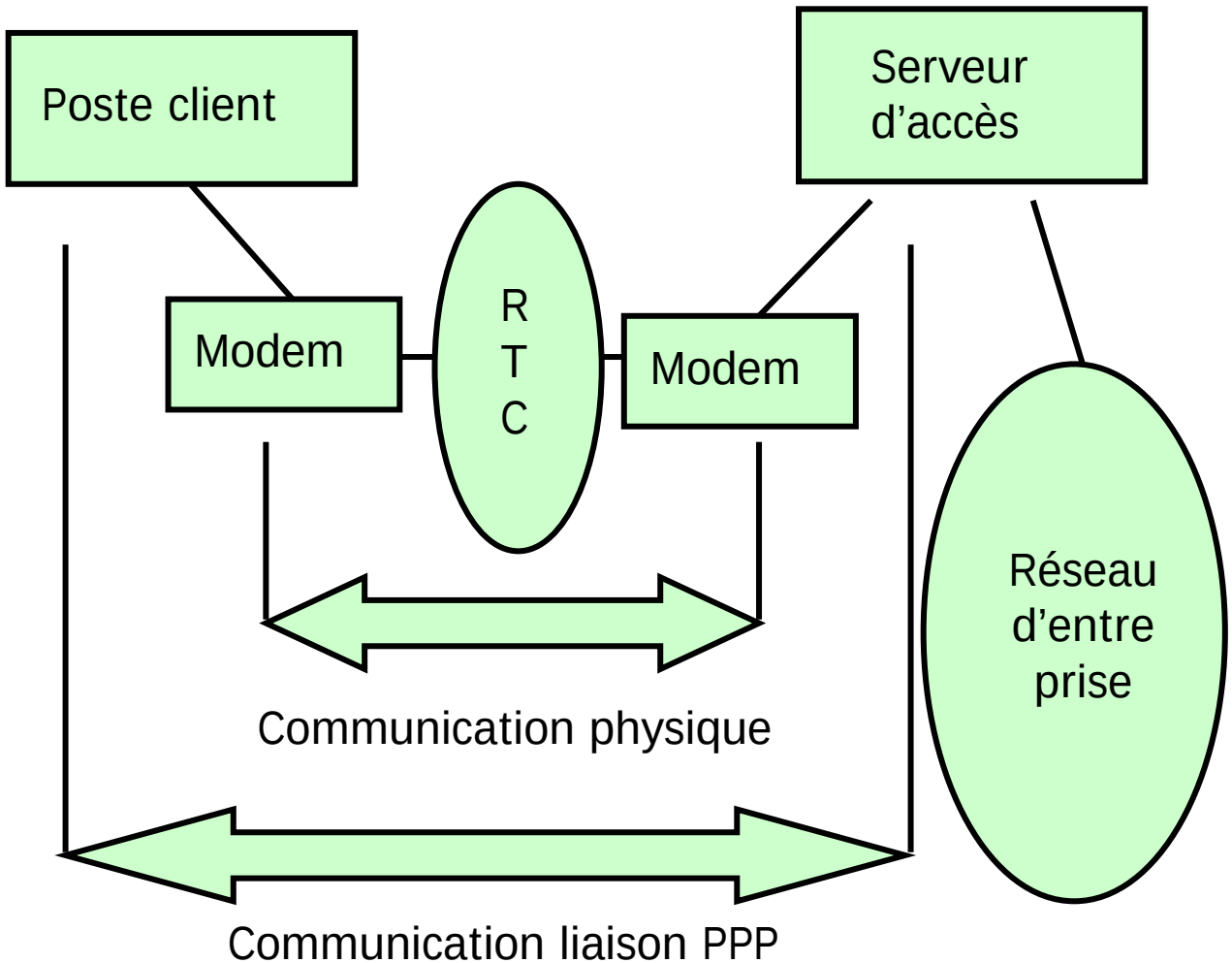
A) Protocole SLIP (RFC 1055 juin 1988) ("Serial Line Internet Protocol")

- **Solutions propriétaires 3COM, SUN** 1984 (R. Adams)
  - SLIP Implanté en 1984 sur BSD. RFC en 1988.
- **Solutions adoptées dans SLIP:**
  - **Délimitation en transparence caractère**
    - Définition d'un caractère de fin de trame "END" (0xC0) et d'une transparence: si END apparaît dans les données séquence d'échappement 0xDB, 0xDC.
    - Si 0xDB apparaît dans les données => émission 0xDB, 0xDB
  - **Uniquement prévu pour transporter de l'IP**
  - **Amélioration** : compression des entêtes (RFC1144 Van Jacobson) => CSLIP ('Compressed SLIP').
- **SLIP: un protocole qui a été quand même très utilisé.**

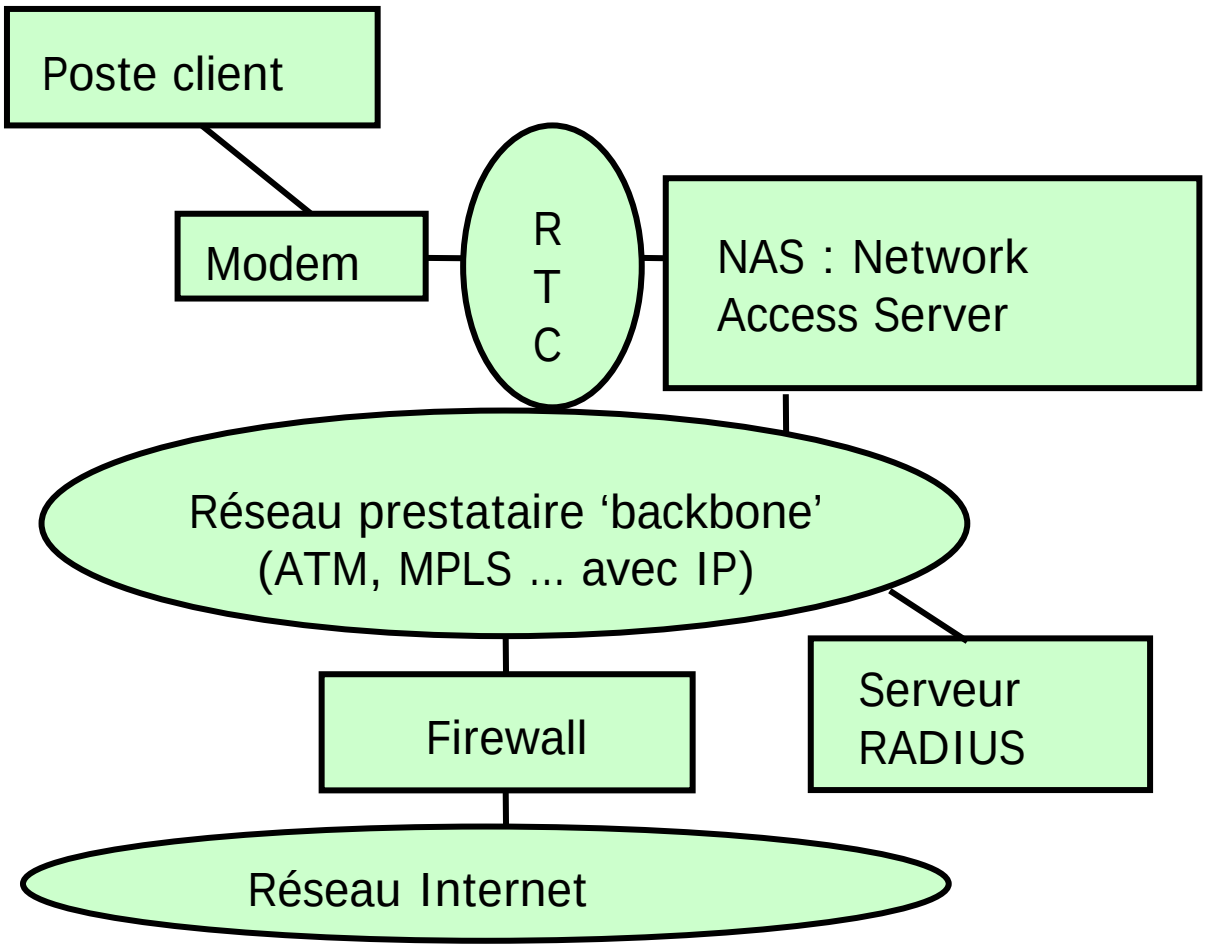
B) Objectifs généraux de PPP

- **Pallier les insuffisances** de SLIP
  - **Pas de** multiplexage, de traitement d'erreurs, d'affectation d'adresse IP, d'authentification d'accès=> Protocole presque vide
  - **La RFC n'a jamais été approuvée définitivement.**
- **PPP : une solution universelle** pour la **communication** au niveau **liaison en point à point.**
  - **Grande variété d'appareils visés :** hôtes, routeurs, points d'accès
  - G**rande variété de protocoles** de niveau 3 (multiplexage de flots d'origines très diverses Internet, Appletalk, IPX, ...)
  - **Grande variété de voies de communication** tous débits.
    - Liaisons spécialisées séries synchrones, asynchrones (ADSL...).
    - Architectures de réseaux pouvant être utilisées comme des voies de communication point à point: RTC, RNIS, X25, FR, ATM, réseaux locaux.
  - **Implantation de solutions** pour de très nombreux problèmes.

C) Architectures de réseaux avec PPP : Solution de base



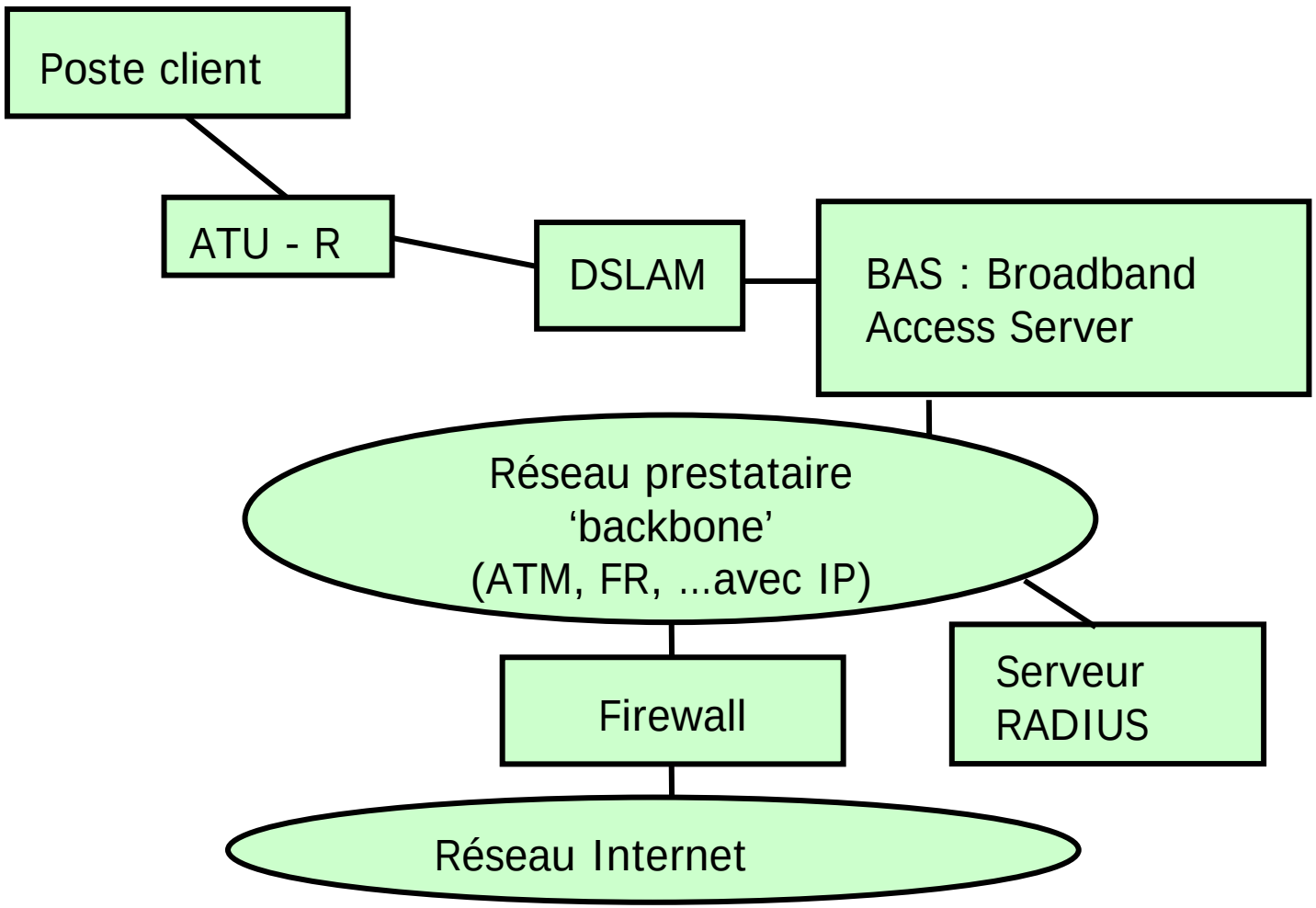
C) Architecture de fournisseur d'accès Internet (FAI, ' ISP ')



C) Notion de NAS : ‘ Network Access Server ’

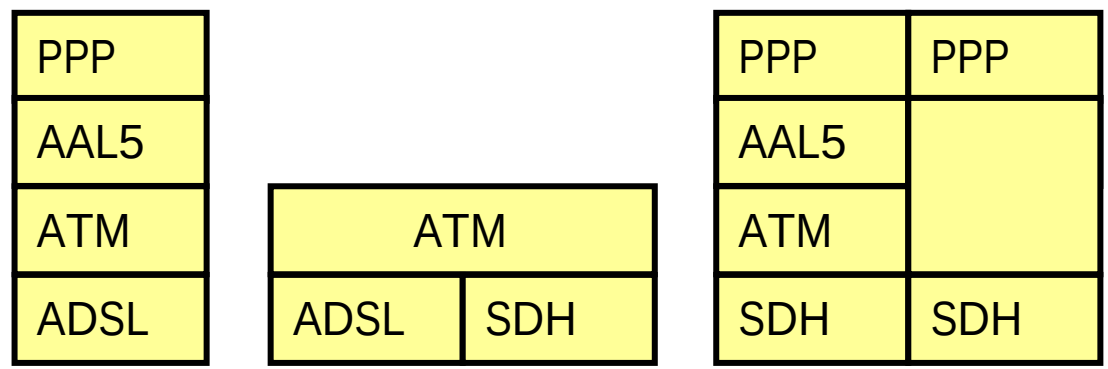
- **NAS (‘Network Access Server’) :** Systèmes dédiés supportant de nombreux types d’accès physiques séries (modems V90, canaux B, T2 , V35, Ethernet, ...).
- **NAS : relais de négociation en PPP** pour l’accès Internet : l’adresse IP, type de compression, ...
- **NAS :** installés sur tout le territoire (dans les autocoms).
- **Utilisation d’un serveur centralisé** pour l’authentification et la comptabilité (RADIUS).
- **NAS :** achemine ensuite les données en PPP/IP sur tous les types de média voulus (ATM, Ethernet, SDH, FR) avec le poste serveur.

C) Architecture de fournisseur d'accès avec ADSL



C) Notion de DSLAM et de BAS

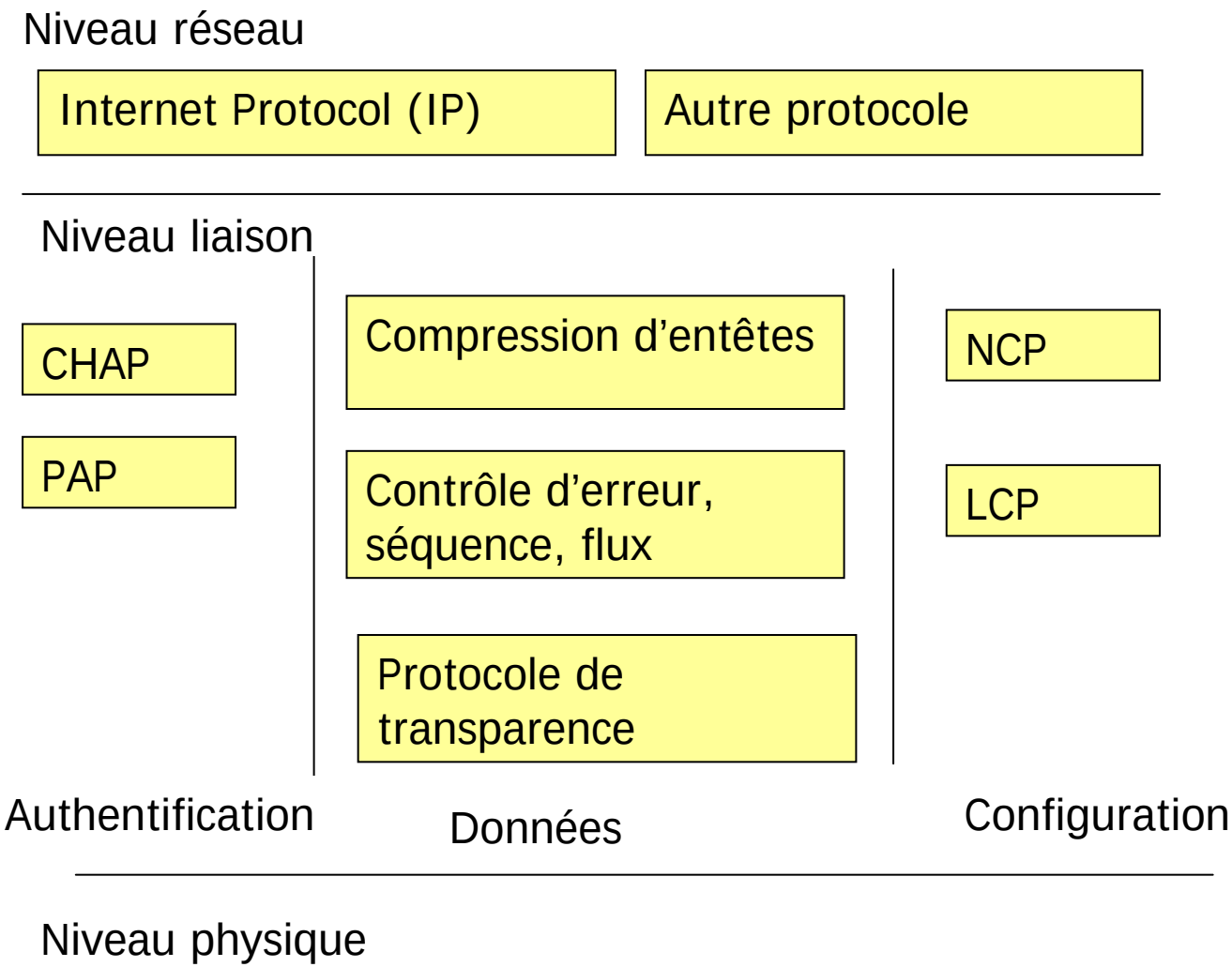
- **ATU-R ‘A**DSL **T**ransceiver **U**nit- **R**emote terminal end’ : Le modem (coté usager)
- **DSLAM ‘D**igital **S**ubscriber **L**ine **A**ccess **M**ultiplexer**’** : un multiplexeur de voies ADSL (avec modem coté prestataire).
- **BAS** ‘**B**roadband **A**ccess **S**erver’ : le NAS pour voies ADSL.



D) Organisation générale de PPP : les grandes parties

- **Protocole de transmission de données :**
  - Un protocole pour communiquer (encapsuler des datagrammes provenant de plusieurs protocoles de niveau réseau ...).
- **Protocole de contrôle de liaison** :
  - LCP : 'Link Control Protocol'
  - Un protocole pour établir, configurer, tester une connexion de liaison.
- **Protocoles de contrôle réseau** :
  - NCPs : 'Network Control Protocols'
  - Une famille de protocoles pour établir, configurer des paramètres pour les protocoles de niveau réseau.
- **Protocoles d'authentification** :
  - Une famille de protocoles pour contrôler l'accès au réseau.

D) Organisation générale de PPP : la suite des protocoles PPP



Protocole PPP
("Point to Point Protocol")

II.2.2
La transmission des données

A) Mécanismes de transparence

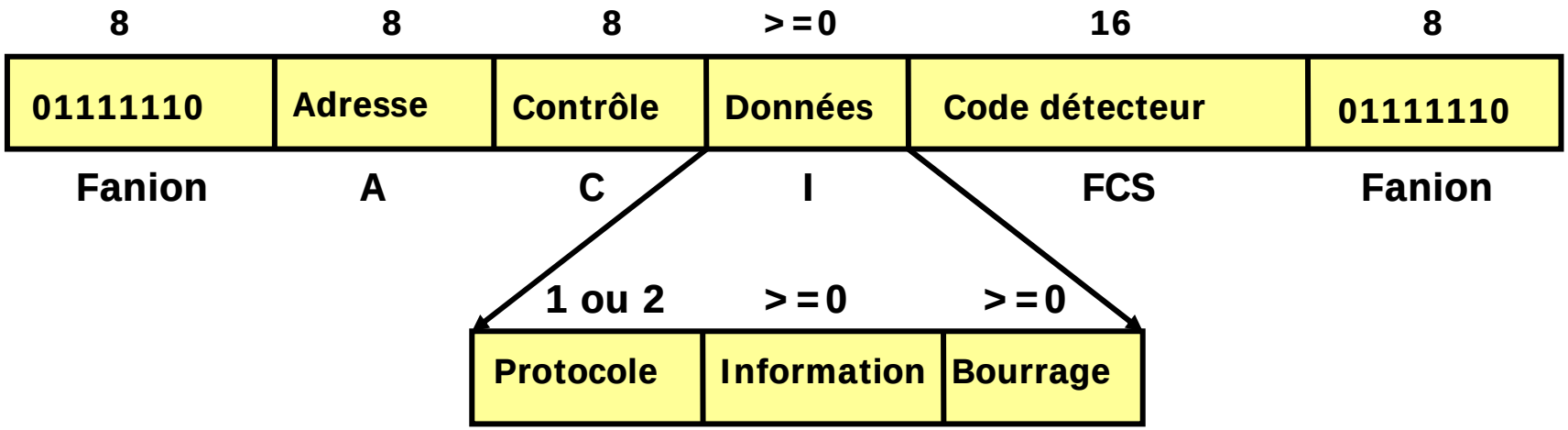
B) Contrôle d'erreur, de flux, de séquence.

C) Multiplexage (encapsulation multi-protocole)

Introduction : organisation générale de la transmission des données

- **Protocole de transparence**
  - Doit fonctionner avec les principales voies de communications existantes (synchrones à trame de bits, asynchrones avec format caractères) => Deux sortes de transparence.
- **Protocole de contrôle d'erreur, de flux, de séquence**
  - Très voisin de LAPB.
- **Multiplexage (encapsulation multi protocole)**
  - PPP permet le multiplexage de différents flots provenant de différents niveaux réseaux (codés sur 2 octets).
- **Compression des entêtes** (Van Jacobson RFC 1144).
  - Compression des entêtes PPP : Recherche d'un surcoût minimum.
  - Ex: PPP utilise 8 octets pour le tramage pouvant être réduits à 2 ou 4 octets lorsque des mécanismes de compression sont utilisés.
  - Compression des entêtes IP, TCP.

Introduction : Format de la trame PPP



- **Reprend le format** de la trame HDLC.
- **Ajoute une possibilité** d'encapsulation multi-protocole.

A) Délimitation (mécanismes de transparence) : Généralités

- **Adaptation à la voie physique (RFC1662 'PPP in HDLC Framing')**
- **Deux types de voies** => deux méthodes différentes de transparence:
  - **Voies synchrones au niveau bit** ('bit synchronous')

    => Transparence binaire HDLC.

    Comme dans les protocoles à trames de bits on rajoute un 0 après tout séquence de 5 bits à 1 **('bit stuffing').**

  - **Voies asynchrones par octet** ou **synchrone au niveau octet**

    => Transparence caractère PPP.

Transparence caractère : Le mode par défaut

- **Délimiteur de trame**
  - Comme en HDLC fanion 01111110 en hexa 0x7F.
- **Caractère d'échappement ('escape')**
  - Caractère 01111101 en hexa 0x7d.
- **Caractères de contrôle** : Ils dépendent de la voie utilisée.
  - Les caractères de contrôle soumis au mécanisme de transparence sont définis par une table de bits **ACCM** ('**Async Control Character Map'**).
  - Si le bit ACCM est à 1 le caractère associé est remplacé par une séquence d'échappement de deux caractères:
    - Le caractère escape.
    - Le caractère de contrôle en ou exclusif avec 0x20.

Transparence caractère (suite)

- Les caractères compris entre 0 et 31 (0x00 et 0x20) **sont en général réservés au pilotage des modems**.
- **Si on veut les utiliser au niveau liaison,** il faut les mettre dans la table **ACCM**.
- **Exemples d'application de la transparence PPP:**
  - 0x05 est codé 0x7d, 0x25. (Contrôle modem code 5)
  - 0x7e est codé 0x7d, 0x5e. (Flag Sequence)
  - 0x7d est codé 0x7d, 0x5d. (Control Escape)
  - 0x03 est codé 0x7d, 0x23. (ETX)
  - 0x11 est codé 0x7d, 0x31. (XON)
  - 0x13 est codé 0x7d, 0x33. (XOFF)

B) Contrôle d'erreur, de flux, de séquence : Choix PPP deux solutions

- **1 Solution standard (par défaut) :**

- Transmission **non fiable**,
- Pas de contrôle d 'erreur, de flux, de séquence.

- **2 Solution fiable (en option) :**

- Transmission **fiable** en mode connecté
- Contrôle d'erreur, de flux, de séquence

Contrôle d'erreur, de flux, de séquence : Transmission non fiable

- **RFC 1662 'PPP in HDLC Framing' : Mode par défaut.**
- **Protocole de liaison sans connexion.**
- Pas de de contrôle **d'erreur**, de **séquence**, de **flux**.
- Les trames incorrectes (FCS faux) sont **détruites.**
- **Une seule trame UI 'Unnumbered Information'**

Fanion 0x7E	Adresse 0xFF	Contrôle 0x03	Données	FCS	Fanion 0x7E

- **Fanion :** un octet (0x7e) pour la synchro trame. Un seul fanion entre deux trames.
- **Adresse :** un octet (0xff), adresse diffusion en multipoint ('All-Stations address**').**
- **Champ Contrôle :** un octet (0x03), type 'Unnumbered Information' (UI) avec bit Poll/Final (P/F) bit à zéro.
- **Champ code polynomial :** Frame Check Sequence (FCS) deux octets. Possibilité de négocier un code sur 32-bit (quatre octets).

Contrôle d'erreur, de flux, de séquence : Transmission fiable

- **Transmission fiable :** 'Numbered Mode ' RFC 1663 PPP Reliable Transmission
- **Mode négociable** en début de transmission :
  - Si l'on considère que la liaison n'est pas assez **fiable.**
  - Si l'on veut éviter des **problèmes** avec la **compression**.
- **Protocole défini en fait** par la norme **ISO 7776** (Description of the X.25 **LAPB-Compatible** DTE Data Link Procedure).

- **Remarque :** Possibilité d'utiliser des tailles de fenêtre de 1 à 127, modes d'ouverture de connexion SABM (1 à 7) ou SABME (1 à 127).

C) Multiplexage et zone de données

- **Format de la zone données de la trame**
  - Type de protocole réseau transporté
  - Charge utile (paquet réseau, LCP, ...)
  - Bourrage
- **Type de protocole réseau transporté**
  - **Valeur sur deux octets** définie par le RFC1340 définissant les codes des protocoles autorisés (IANA-ICANN).
  - **Quelques exemples de valeurs possibles**

    0x0021 IP 0x002B IPX

    Ox8021 IPCP 0xC021 LCP

    0x002D TCP/IP avec compression

    0x002F TCP/IP Sans compression
  - **Codage du type de protocole** sur deux octets ramené à un octet par négociation si nécessaire.

Zone données : taille maximum et bourrage

- **Zone des données usager :** longueur maximum en PPP
  - Terminologie liaison : **MRU** 'Maximum Receive Unit'
  - **Valeur négociable** à l'établissement de la liaison : valeur par défaut = 1500 octets.
  - **Fragmentation** par le niveau réseau.
- **Bourrage**
  - Si le médium de transmission utilise un **format fixe** (taille de paquets, de cellule ATM, ...)
  - et que cette taille obligatoire de la trame ne correspond pas à la taille de l'information à transporter.

Protocole PPP
("Point to Point Protocol")

II.2.3
La configuration de liaison
(LCP ‘ Link Configuration Protocol ’

A) Gestion de connexion avec LCP

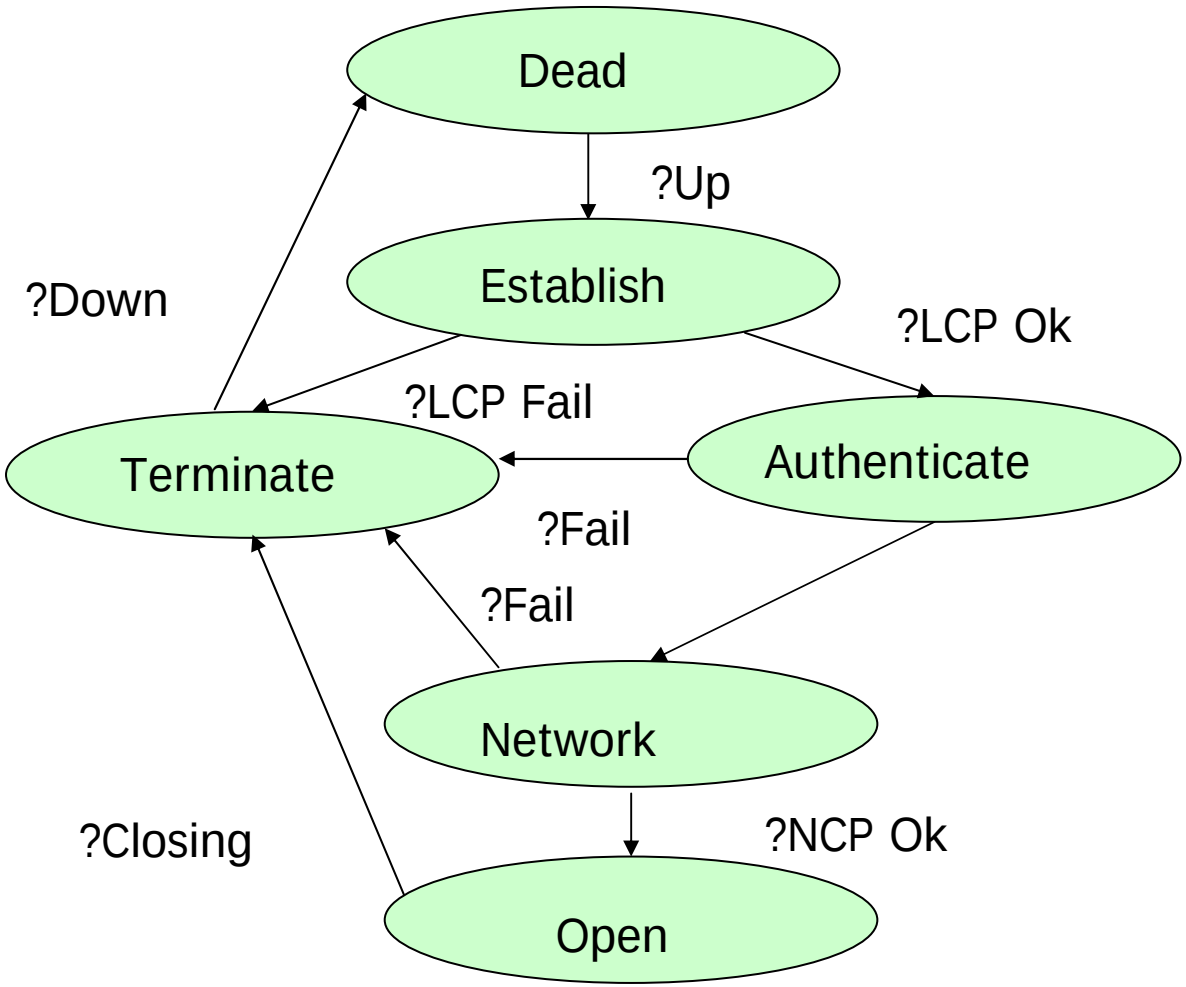
B) Le protocole LCP

C) Les options de configuration LCP.

A) LCP et la gestion de connexion

- **LCP permet** d'ouvrir, de fermer la liaison PPP : **gestion de connexion.**
- **LCP permet de négocier** des paramètres de fonctionnement à l'ouverture (puis de renégocier),
  - Existence de **paramètres par défaut** automatiquement échangés au début entre pairs sans intervention opérateur.
  - **Configuration possible par l'opérateur** à l'ouverture**.**
  - **Exemples de négociation**
    - Définir le format d'encapsulation (négociation de la compression)
    - Définir la taille maximale des trames (taille MRU).
- **LCP permet de détecter** certaines conditions d'erreur (liaison en fonctionnement correct, en panne, en boucle)

Automate LCP



Établissement d'une connexion PPP : Commentaire des différentes phases (1)

- **Liaison non opérationnelle** (‘Link Dead’)
  - Le niveau physique n’est pas prêt.
  - Un événement externe (détection de porteuse, démarrage opérateur, ...) permet de passer à l’état prêt.
- **Liaison en cours d’établissement** (‘Link Establishment phase’).
  - Le LCP (‘Link Control Protocol’) établit les paramètres de liaison.
- **Authentification** (‘Link Authentication Phase’)
  - L’authentification (si elle est demandée) prend place aussitôt que possible après établissement des paramètres de liaison.
  - Si l’authentification échoue on termine.

Établissement d'une connexion PPP : commentaire des différentes phases (2)

- **Négociation des paramètres de réseau** (‘Network-Layer Protocol Phase’).
  - Chaque niveau réseau (comme IP, IPX, ou AppleTalk) configure ses propres paramètres (‘Network Control Protocol’).
- **Ouvert** (‘Open Phase’)
  - Après avoir atteint cet état PPP peut transporter les paquets de données.
- **Terminé** (‘Link Termination Phase’)
  - PPP termine dans différents cas: perte de porteuse, mauvaise qualité, mauvaise authentification, expiration d’un délai d’inactivité, fermeture décidée par l’opérateur.
  - LCP échange des paquets de terminaison.
  - Informe le niveau réseau de la fermeture.

B) Le protocole LCP

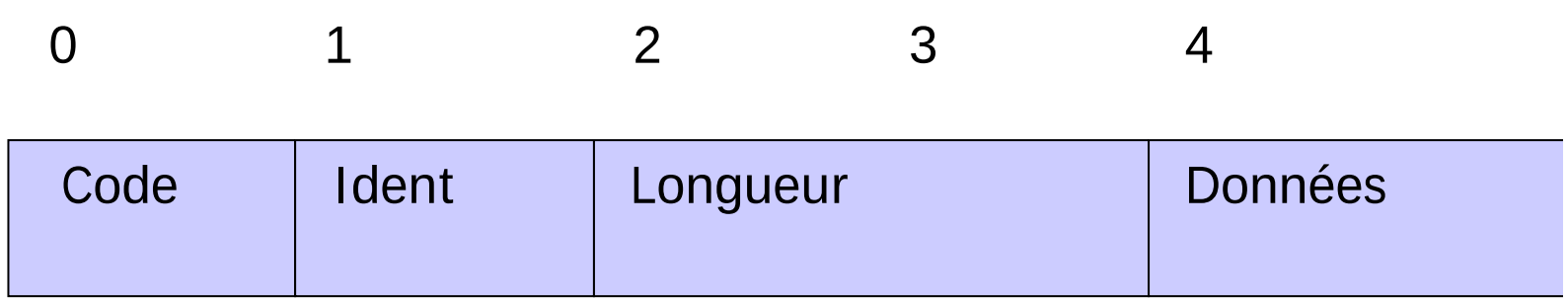
Protocole qui permet principalement la négociation des options de configuration d'une liaison PPP.

- Existence d'une configuration **par défaut**.
- LCP permet de **modifier ces options**.
- Chaque **extrémité propose ses options.**

Principales notions

- Définition des messages LCP et principes de la négociation.
- Définition des **options (attributs) négociables.**
- Les messages LCP sont encapsulés dans **la zone information d'une trame PPP (type de protocole C021).**

Protocole LCP : Format de la trame LCP



- **Représentation uniquement de la charge utile LCP.**
- **Code (‘ code’)** : sur un octet le type LCP.
- **Identificateur (‘ Identifier’)** : sur un octet il permet d’associer les requêtes et les réponses.
- **Longueur (‘Length’) : s**ur deux octets, la longueur inclut le code, l‘identificateur et la donnée.
- **Données (‘Data’)** : la zone données est vide ou son format est défini par le code type : contenu essentiel les valeurs négociées.

Protocole LCP : Liste des types LCP (codes)

Code	Désignation du paquet
1	Configure-Request
2	Configure-Ack
3	Configure-Nak
4	Configure-Reject
5	Terminate-Request
6	Terminate-Ack
7	Code-Reject
8	* Protocol-Reject
9	* Echo-Request
10	* Echo-Reply
11	* Discard-Request
12	* RESERVED

Codes valables pour IPCP et LCP

*** Codes valables pour LCP seulement**

Protocole LCP : Description détaillée de types LCP (1)

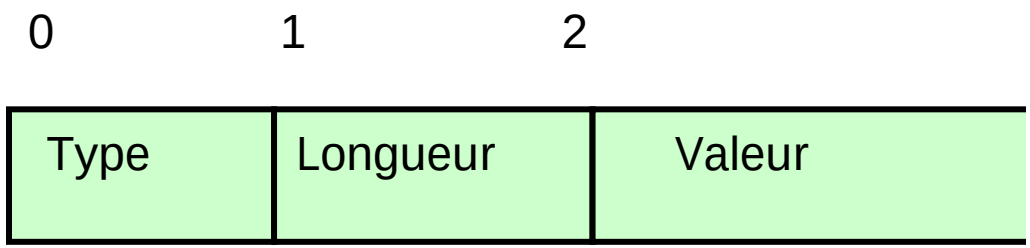
- **Configure-Request**
  - Pour ouvrir une connexion : le paquet Configure-Request contient toutes les options que l'on modifie par rapport aux valeurs par défaut
- **Configure-Ack**
  - Si toutes les options de configuration sont reconnues et acceptées réponse : configure-Ack.
- **Configure-Nak**
  - Si toutes les options de configuration sont reconnues mais certaines ne sont pas acceptables: réponse Configure-Nak.
  - Le champ données contient les valeurs de configuration non acceptées
- **Configure-Reject**
  - Si certaines options de configuration ne sont pas reconnues ou ne sont pas acceptables dans le cadre d'une négociation prévue par l'administrateur réseau alors la réponse est un 'configure-Reject'.

Protocole LCP : Description détaillée de types LCP (2)

- **Terminate-Request et Terminate-Ack**
  - Pour fermer une connexion LCP.
- **Code-Reject :** Type LCP inconnu (champ code).
- **Protocol-Reject :** Type de protocole inconnu (champ proto).
- **Echo-Request et Echo-Reply**
  - Permet de tester une liaison PPP : échange d'un nombre magique sur 4 octets caractéristique de l'émetteur si une valeur a été négociée (sinon 0). Le nombre magique doit être celui du site distant. Si c'est celui du site local il y a une boucle.
- **Discard-Request**
  - Outil de test de liaison : une émission simple, local vers distant, avec destruction immédiate du paquet.
  - Utilisation : Déverminage, test de performance, ...
  - Contient un nombre magique s'il a été négocié (sinon 0).

C) Valeurs négociables : Options de configuration

- **Données négociables :** codées type, longueur, valeur



- **Différents types de données négociables**

0	RESERVED	7	Protocol-Field-Compression
1	Maximum-Receive-Unit	8	Addre-and-Contr-Field-comp
2	Async-Control-Character-Map	9	FCS Alternatives
3	Authentication-Protocol	10	Padding protocol
4	Quality-Protocol	11	Numbered mode
5	Magic-Number		etc .....
6	Reserved		

Description de quelques options (1)

- **Maximum-Receive-Unit (MRU)**
  - La valeur de la taille maximum par défaut est de 1500 octets. Une implantation doit toujours être capable de recevoir cette taille.
  - Par cette négociation on peut indiquer au site distant que l'on peut recevoir des paquets de plus grande taille ou que l'on demande la transmission de paquets de plus petite taille.
- **Async-Control-Character-Map**
  - Permet de redéfinir la tables des codes caractères qui seront soumis à la transparence caractères.
  - La table est sur 32 bits (4 octets) et concerne les codes caractères de 0x0000 à 0X0020.
- **Authentication-Protocol**
  - Permet de définir le protocole d'authentification utilisé :
  - PAP (code protocole 0xC023, CHAP (code 0xC223).

Description de quelques options (2)

- **Quality-Protocol**
  - Pour négocier le type de protocole de gestion de la qualité de la liaison (valeur sur 2 octets). Principal choix: LQR ‘Link Quality Report’ 0xC025
- **Magic-Number**
  - Pour detecter les liaisons qui bouclent (‘looped-back links’).
  - Chaque coté tire aléatoirement un nombre aléatoire sur 4 octets (le nombre magique qui doit changer à chaque nouvelle ouverture.
- **Protocol-Field-Compression (PFC)**
  - Pour demander la compression de la zone protocole (de 2 à 1 octet).
  - Le FCS est alors calculé sur la trame compressée (pas sur la trame originale non compressée).
- **Address-and-Control-Field-Compression (ACFC)**
  - Pour demander la compression des zones adresses et contrôle dans les trames PPP. Le FCS est calculé sur la trame compressée.

Protocole PPP ("Point to Point Protocol")

II.2.4

La configuration de réseau

(NCP ‘ Network Configuration Protocol ’

Introduction NCP

- **NCP permet la négociation d'options** de configuration du **niveau 3 réseau** dans le cadre du niveau 2 liaison.
- **Fonctions nécessairement dépendantes** du protocole réseau utilisé => Existence de protocoles **spécifiques** NCP par type de réseau.
- **Exemples de RFC NCP existantes :**

  RFC 1332 IPCP pour IP

  RFC 2023 IPV6CP pour IPV6

  RFC 1552 IPXCP pour IPX

  RFC 1378 ATCP pour AppleTalk

  RFC 1377 0SINLCP pour OSI

  RFC 2097 NBFCP pour NetBeui ..... Etc

Le protocole IPCP ‘IP Configuration Protocol’

- **Protocole qui utilise les messages de LCP** (types 1 à 7) pour négocier des options relatives à IP:
- **Exemple 1) Type d’option 2: IP-Compression-Protocol**
  - Une façon de négocier la compression des entêtes IP : théoriquement un code sur deux octets permet de sélectionner la compression souhaitée => Pratiquement deux possibilités : **002d** Compression TCP/IP Van Jacobson ou par défaut pas de compression.
- **Exemple 2) Type d’option 3 : IP-Address**
  - L’émetteur peut dans un Configure-Request demander d’utiliser une adresse IP s’il en connaît une (0 sinon).
  - Le site distant peut accepter l’adresse proposée ou rejeter par un NAK en retournant une autre adresse.

=> Deux zones adresses IP (proposée et affectée) pour la négociation,

- **Affectation d’adresses plus utilisée** => DHCP

Protocole PPP ("Point to Point Protocol")

Conclusion

Raisons du succès du protocole PPP

- **Hégémonie du protocole IP**

=> PPP est obligatoirement le niveau liaison le **plus répandu.**

- **Mais aussi beaucoup de qualités**
  - Protocole de convergence adapté à tout
    - **pour toutes les voies point à point et tous les réseaux .**
    - **pour toutes les architectures de réseau et tous les protocoles**.
  - **Rassemble** la plupart des concepts et solutions..
  - **Enrichit** les fonctions habituellement dévolues au niveau liaison par des fonctions d'administration et de sécurité (LCP, NCP, authentification).
  - **En constante amélioration.**

Conclusion

Évolution des protocoles de liaison industriels

- **Historiquement : grande variété de propositions** qui diffèrent souvent peu : des options jugées nécessaires dans le domaine visé.

  => Trop grande hétérogénéité.

- **Une orientation des protocoles les plus anciens vers des solutions assez riches** en termes de contrôle d'erreur, de flux, de séquence

  => Solutions jugées plutôt coûteuses justifiées par les taux d'erreurs.

- **Changement d'orientation** avec Internet :
  - Redistribution des fonctions en s'orientant vers un découpage ou le rôle de contrôle d'erreur, de flux, de séquence est allégé si nécessaire.
  - Les fonctions d'administration sont renforcées.

Bibliographie : Cours de liaison point à point

- A. S. Tannenbaum ‘Computer Networks’ Prentice Hall.
- W.R. Stevens "TCIP/IP Illustrated, The protocols" , Addison Wesley.
- L. Toutain ‘Réseaux locaux et Internet’ Hermés.
- Sites web et RFC.