
Réseaux I
Protocole de liaison de données

Nicolas Baudru
mél : nicolas.baudru@esil.univmed.fr
page web : nicolas.baudru.perso.esil.univmed.fr

Lors de la transmission des données sur un support de transmission, des erreurs peuvent se produire : le circuit de données n'est pas sûr en général.

Le protocole de liaison de données supervise et définit un ensemble de règles pour assurer la fiabilité des échanges sur un circuit de données.

Règles pour la spécification des unités de données échangées, appelées **trames** :

- ▶ délimitation des différentes trames
- ▶ contrôle de la validité des trames
- ▶ autres informations de commande

Règles pour la gestion du dialogue entre extrémités :

- ▶ procédure de reprise sur erreur
- ▶ contrôle du flux de données entre les extrémités
- ▶ gestion des acquittements des trames

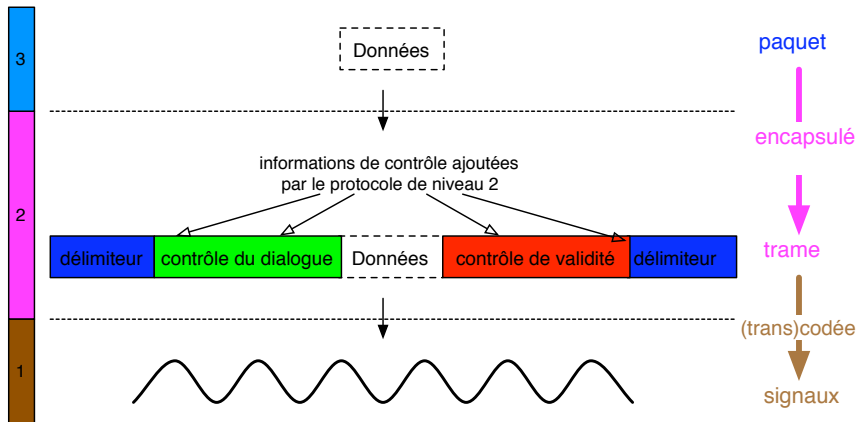
- 1 Spécification des trames
- 2 Gestion du dialogue entre extrémités
- 3 Conclusion et mise en oeuvre

- 1 Spécification des trames
- 2 Gestion du dialogue entre extrémités
- 3 Conclusion et mise en oeuvre

La trame (données de couche 2)

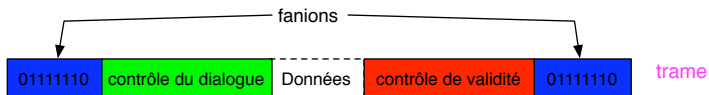
Trame = paquet (données provenant de la couche 3) + infos de contrôle.
On dit que le paquet est **encapsulé** dans une trame.

couches OSI



Délimitation de trames par fanion

Une séquence de bits spécifiques, appelée **fanion**, est ajoutée au début et à la fin des trames pour les délimiter.



Afin d'interpréter correctement la trame, un mécanisme de **transparence** doit être mis en oeuvre par le protocole sur l'ensemble de la trame hors fanions. On peut utiliser par exemple la méthode du **bit stuffing**.

Exemple : Données du paquet : 0110 1111 1100 1110 1001.

Fanion utilisé par le protocole : 0111 1110.

Sans mécanisme de transparence la trame envoyée serait mal interprétée :

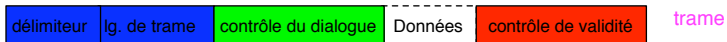
0111 1110 0110 1111 1100 1110 1001 0111 1110.

Avec mécanisme de transparence, il n'y a plus d'ambiguïté :

0111 1110 0110 1111 10100 1110 1001 0111 1110.

Délimitation de trames par transmission de la longueur des données

Un champs en entête indique la longueur des données utiles de la trame (en octet, mot ou double mot). Ce champs est généralement précédé d'un délimiteur de trame.



Ce mécanisme induit une taille de trame maximale. De plus les données utiles doivent avoir une longueur multiple d'un octet, d'un mot ou d'un double mot suivant le cas.

Exemple : Données du paquet : 0110 1111 1110 1110 1001 0010.

Taille du champs lg : 4 bits

Nombre d'octets de la trame : 3 soit 0011 (en binaire)

Fanion : 0101 0101

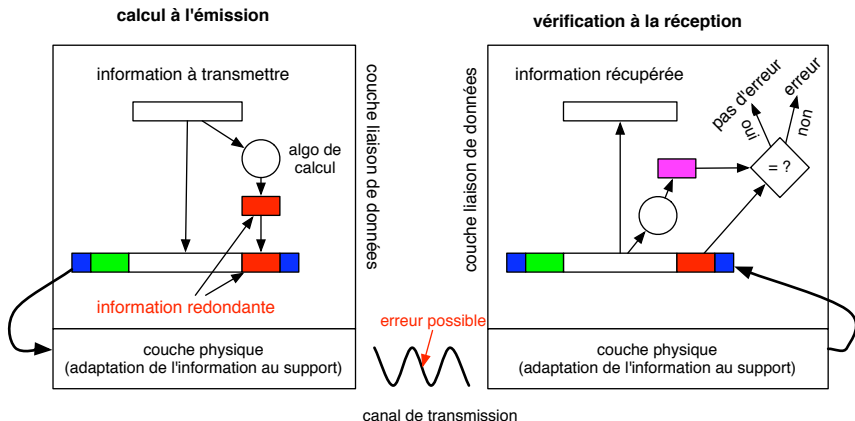
Trame envoyée : 0101 0101 0011 0110 1111 1100 1110 1001 0010.

Les supports de transmission n'étant pas fiables, il faut mettre en oeuvre des mécanismes pour vérifier la validité des trames reçues. Pour cela, on ajoute aux trames transmises une certaine **redondance**. Plusieurs mécanismes de détection d'erreur existent :

- ▶ détection par echo
- ▶ détection par répétition
- ▶ détection d'erreur par clé calculée
- ▶ détection et correction d'erreur par code correcteur

Détection d'erreurs par clé

On ajoute aux trames transmises une redondance qui est une information de contrôle calculée par un algorithme spécifié dans le protocole à partir des données utiles.



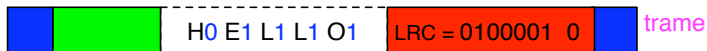
Exemple d'algorithmes de détection d'erreurs : VRC et LRC

Vertical Redundancy Check (VRC) : on effectue un calcul de parité pour chaque caractère. Permet de déterminer si un nombre impair d'erreur s'est produit lors de la transmission d'un caractère.

Longitudinal Redundancy Check (LRC) : on effectue un calcul de parité sur les bits de même rang. S'utilise souvent avec VRC pour renforcer le code.

Exemple : VRC + LRC

lettre	lettre codée	VRC
H	0001001	0
E	1010001	1
L	0011001	1
L	0011001	1
O	1111001	1
LRC	0100001	0



Exemple d'algorithmes de détection d'erreurs : CRC

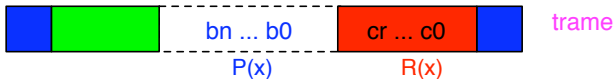
Cyclic Redundancy Check (CRC) : appelé aussi **contrôle polynomial**, il est très utilisé dans les protocoles modernes car il permet de détecter des erreurs sur plusieurs bits.

Cette méthode utilise les polynômes suivants :

- ▶ **Polynôme associé à une information $P(x)$** : soit $b_n \dots b_0$ la suite de bits correspondant à l'information à transmettre. Alors $P(x) = b_n x^n \dots b_0 x^0$.
- ▶ **Polynôme générateur $Q(x)$** : polynôme caractérisant le contrôle.
- ▶ **Polynôme reste $R(x)$** : correspond à l'information redondante ajoutée en fin de trame. Si r est le degré de $Q(x)$ alors $R(x)$ est le reste de la division euclidienne de $x^r \times P(x)$ par $Q(x)$:

$$R(x) = (x^r \times P(x)) \bmod Q(x) = c_{r-1} x^{r-1} \dots c_0 x^0$$

L'information redondante correspondante à $R(x)$ est $c_{r-1} \dots c_0$.



Exemple d'algorithmes de détection d'erreurs : CRC (suite)

Exemple : Soit 1000001110000100 l'information à transmettre. Alors le polynôme correspondant est :

$$P(x) = x^{15} + x^9 + x^8 + x^7 + x^2$$

Soit le polynôme de contrôle de degrés 12 suivant :

$$Q(x) = x^{12} + x^{11} + x^3 + x^2 + x + 1$$

La division de $x^{12} \times P(x)$ par $Q(x)$ donne le polynôme reste :

$$R(x) = (x^{12} \times P(x)) \bmod Q(x) = x^{11} + x^9 + x^8 + x^7 + x^6 + x^4 + 1$$

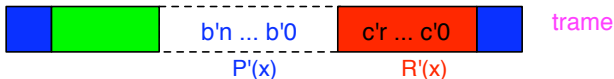
L'information redondante à ajouter en fin de trame est donc : 101111010001.

La trame envoyée contient donc : 1000001110000100 101111010001.



Exemple d'algorithmes de détection d'erreurs : CRC (fin)

Le protocole de liaison de données du coté destinataire reçoit donc une trame du type :



Pour vérifier la validité des données, il suffit d'effectuer le quotient suivant :

$$(x^r \times P'(x) + R'(x))/Q(x)$$

Si le reste est nul alors on suppose qu'il n'y a pas eu d'erreur de transmission. Si le reste n'est pas nul, alors il y a eu une erreur de transmission et il faut demander la réémission de la trame.

- 1 Spécification des trames
- 2 Gestion du dialogue entre extrémités
- 3 Conclusion et mise en oeuvre

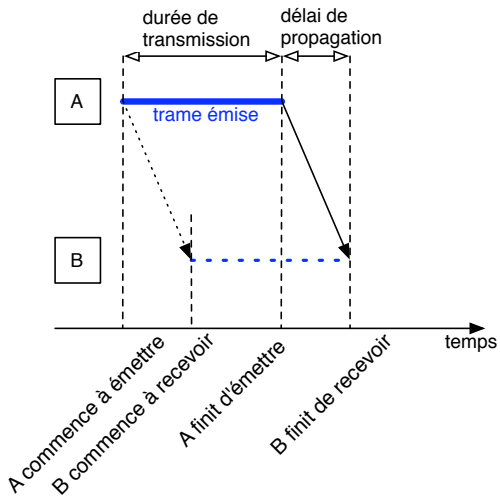
Modes d'exploitation d'une liaison de données

Le mode d'exploitation d'une liaison de données peut être :

- ▶ **simplex** : l'échange de données se fait dans un seul sens ;
- ▶ **half-duplex** : l'échange de données se fait **alternativement** dans les deux sens. Ce mode d'exploitation nécessite souvent des règles supplémentaires d'accès au support pour éviter la contention. Ex. : Ethernet (IEEE 802.3) ;
- ▶ **full-duplex** : l'échange de données se fait dans les deux sens **simultanément**. Ex. : ppp.

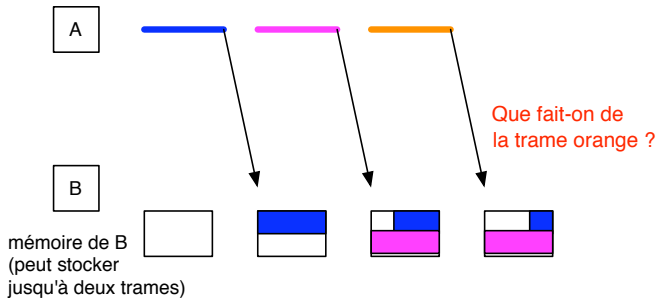
Remarque : le mode d'exploitation d'une liaison de données peut différer de celui du circuit de données. Par exemple, un circuit de données permettant des communications en half-duplex peut être exploité seulement en simplex par le protocole de liaison.

Représentation des échanges de données



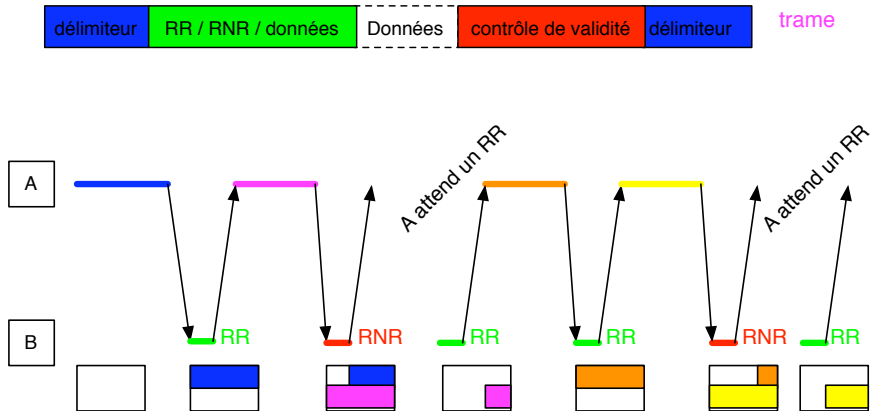
Contrôle de flux — définition

Contrôle de flux : mécanisme de contrôle du rythme d'envoi des informations vers le récepteur.



Contrôle de flux — trames de supervision

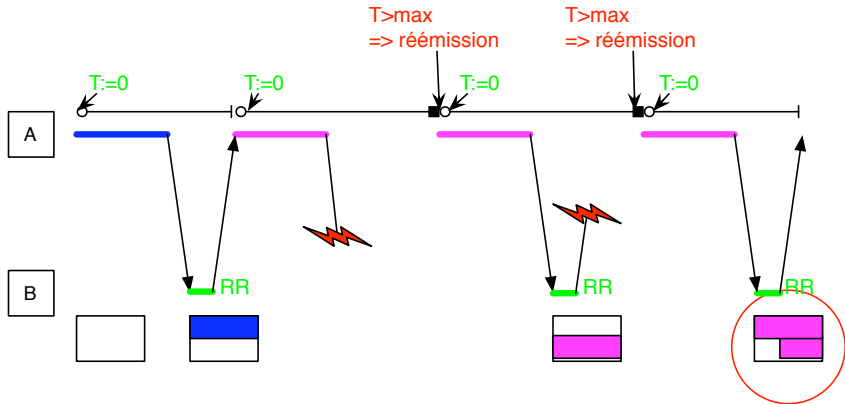
Trames de supervision : **RR** (Receiver Ready) et **RNR** (Receiver Not Ready).



Exemple sur un canal exempt d'erreur

Gestion des acquittements

Les trames de supervision peuvent aussi servir d'acquittement afin de gérer les erreurs de transmission. Cela nécessite l'utilisation d'un **temporisateur**.



La même trame est reçue deux fois !

Numérotation des trames d'information

Pour éviter qu'une même trame soit reçue deux fois, le protocole de communication numérote chaque trame d'information.



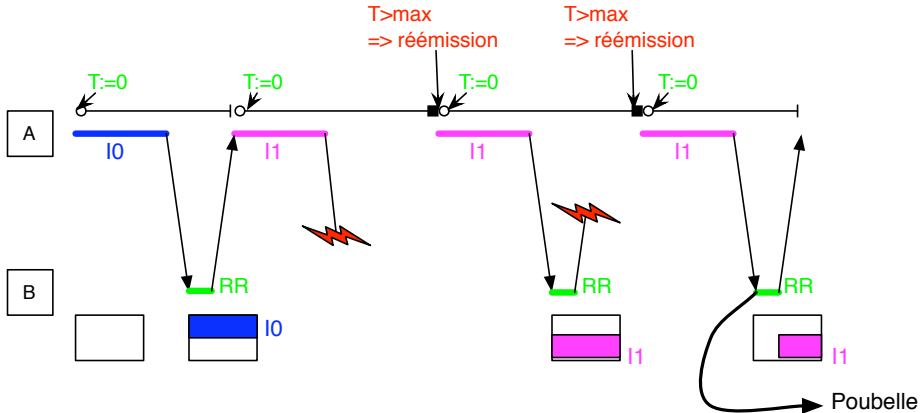
Deux trames avec des numéros différents seront considérées différentes !

Mise en oeuvre :

- ▶ Soit m le nombre de bits dans la trame utilisé pour la numérotation. Alors il ne peut y avoir plus de 2^m numéros de trames différents.
 \implies il faut numérotter modulo 2^m .
- ▶ L'émetteur possède un compteur E (modulo 2^m). Si le compteur vaut n alors la prochaine trame envoyée portera le numéro n . Puis le compteur est incrémenté de 1 (mod 2^m).
- ▶ Le récepteur possède un compteur R (modulo 2^m). Si le compteur vaut n alors la prochaine trame reçue doit porter le numéro n . Puis R est incrémenté de 1 (mod 2^m).

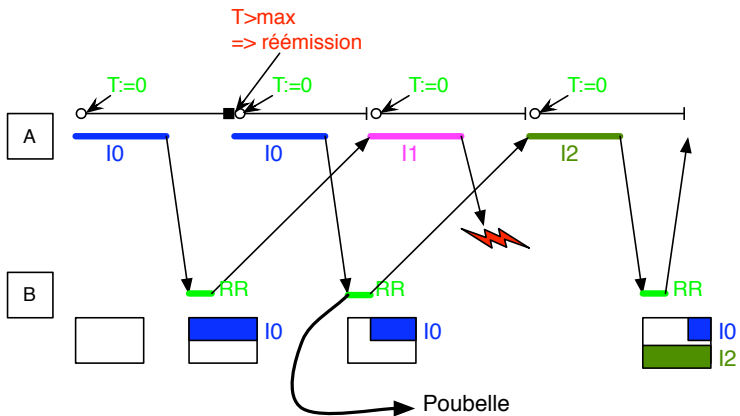
Que manque-t-il dans cet algorithme ?

Numérotation des trames d'information



Le choix de la temporisation a-t-elle une influence sur la gestion du dialogue ?

Numérotation des trames d'information : problème de temporisation



Comment régler le problème ?

Notion de fenêtre

Dans les exemples précédents, l'émetteur doit attendre de recevoir un RR avant de pouvoir réémettre. Afin d'augmenter l'efficacité du dialogue, on utilise la notion d'**anticipation** :

on autorise maintenant l'émetteur à envoyer plusieurs trames successivement sans avoir reçu de RR.

Si la numérotation des trames se fait modulo k , le nombre de trames r pouvant être émises sans RR doit être inférieur à $k - 1$.

Pourquoi ?

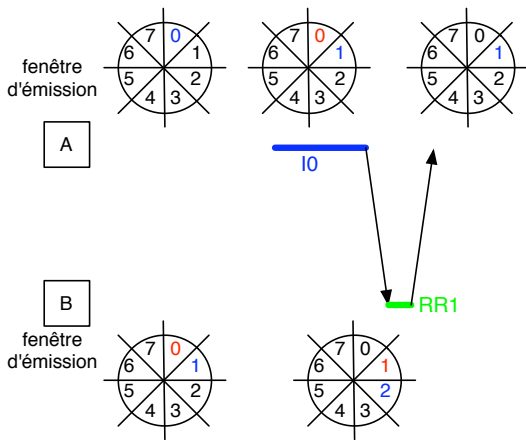
Ce nombre r est appelé **fenêtre d'anticipation**.

Puisque plusieurs trames peuvent être émises successivement, il faut en cas d'erreur que la trame de supervision RR explicite quelles trames ont été correctement reçues. \implies il faut donc numéroter les trames RR.

Exemple : Si l'émetteur a émis les trames I3, I4, I5 et I6 puis qu'il reçoit la trame RR5, alors les trames I3 et I4 ont bien été reçues.

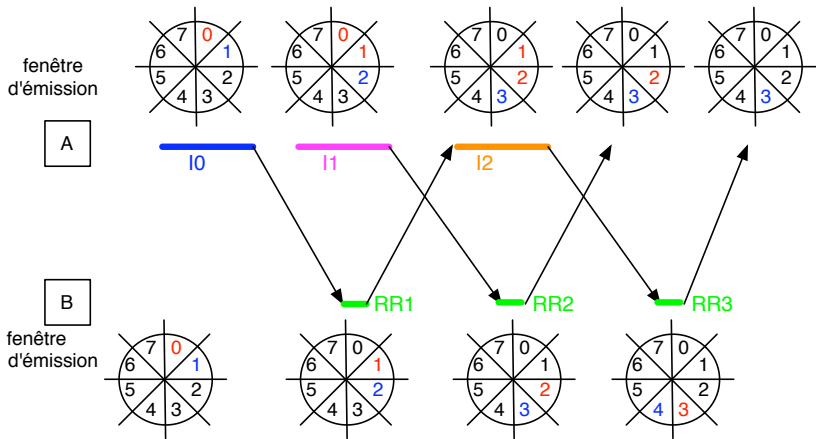
Notion de fenêtre — Un premier exemple

Avec une fenêtre d'anticipation égale à 1 sur un circuit sans erreur :



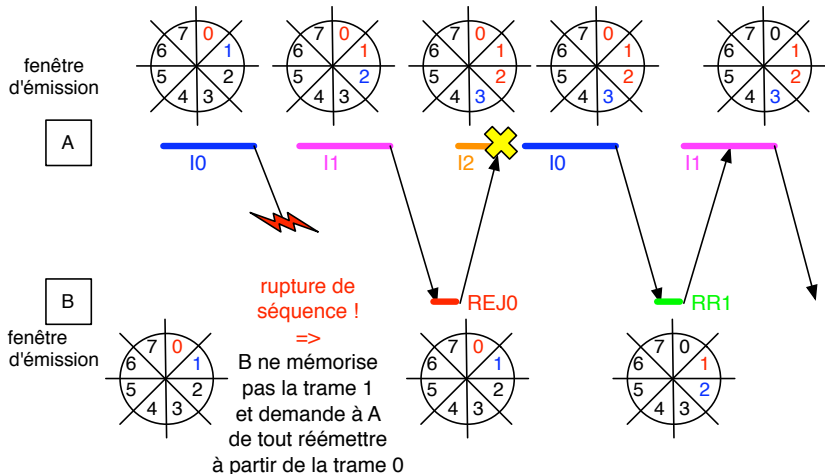
Notion de fenêtre — un autre exemple

Avec une fenêtre d'anticipation égale à 2 sur un circuit sans erreur :



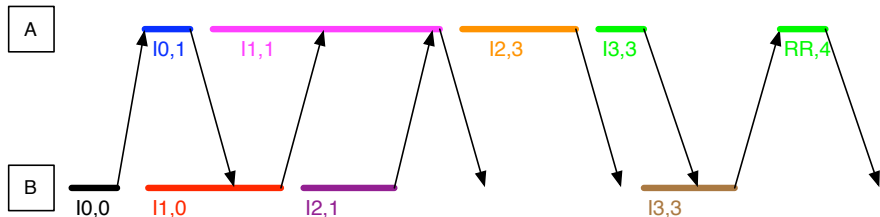
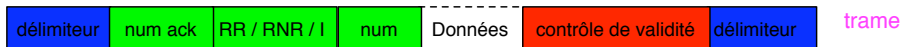
Protocole Go-Back-N

Avec une fenêtre d'anticipation égale à 3 sur un circuit avec erreur :



Piggy-Backing

Supposons que l'échange soit bidirectionnel. Dans ce protocole, les trames d'information vont aussi jouer le rôle de trames de supervision. Pour cela, il suffit d'ajouter un champs supplémentaire acquittant les trames émises dans le sens opposé :



Avec une fenêtre d'anticipation égale à 3.

- 1 Spécification des trames
- 2 Gestion du dialogue entre extrémités
- 3 Conclusion et mise en oeuvre**

Services offerts par un protocole de liaison de données

Nous avons vu qu'un protocole de liaison de données peut offrir plusieurs services au protocole de niveau supérieur :

- ▶ service sans acquittement, ni connexion, ni contrôle de flux
- ▶ service avec acquittement, mais sans connexion ni contrôle de flux
- ▶ service avec acquittement, connexion et contrôle de flux

Ne manque-t-il pas quelque chose pour que le protocole de liaison de données soit complet ?

HDLC (High level Data Link Control)

Protocole point à point normalisé par l'ITU. Il offre un service avec acquittement, connexion et contrôle de flux.

- ▶ fonctionne en duplex intégral sur un liaison point à point
- ▶ transmission synchrone
- ▶ met en oeuvre le bit stuffing
- ▶ utilise le plus souvent le Go-back-N avec un mécanisme de contrôle de flux
- ▶ transmission par bit de poids faible d'abord pour chaque champs

HDLC (High level Data Link Control)

Structure de la trames HDLC :

- ▶ **Fanion** : marque le début et la fin de la trame de longueur quelconque.
- ▶ **Adresse** : identifie une des extrémités de la liaison.
- ▶ **Contrôle** : décrit le type de la trame (RR/RNR/REJ,I,U).
- ▶ **Données** : données provenant du protocole supérieur (peut être vide).
- ▶ **FCS** : Frame Control Sequence. C'est un contrôle polynomial avec comme polynôme générateur $x^{16} + x^{12} + x^5 + 1$.



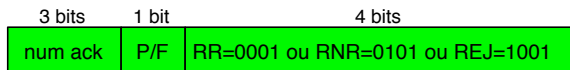
HDLC — le champs contrôle

Trois types de trame HDLC :

- ▶ I : trame d'information



- ▶ S : trame de supervision RR/RNR/REJ. Le champs de données est vide.



- ▶ U : trame non numérotée. Sert entre autres pour l'initialisation et la libération de la liaison. Le champs de données est vide.

