



# Partie 4 : Protocoles de liaison de données

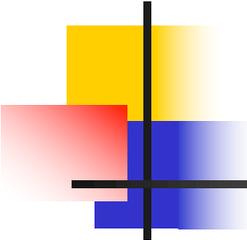
---

Olivier GLÜCK

Université LYON 1 / Département Informatique

Olivier.Gluck@univ-lyon1.fr

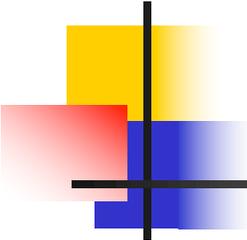
<http://perso.univ-lyon1.fr/olivier.gluck>



# Copyright

---

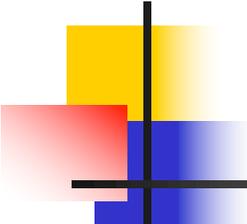
- Copyright © 2024 Olivier Glück; all rights reserved
- Ce support de cours est soumis aux droits d'auteur et n'est donc pas dans le domaine public. Sa reproduction est cependant autorisée à condition de respecter les conditions suivantes :
  - Si ce document est reproduit pour les besoins personnels du reproducteur, toute forme de reproduction (totale ou partielle) est autorisée à la condition de citer l'auteur.
  - Si ce document est reproduit dans le but d'être distribué à des tierces personnes, il devra être reproduit dans son intégralité sans aucune modification. Cette notice de copyright devra donc être présente. De plus, il ne devra pas être vendu.
  - Cependant, dans le seul cas d'un enseignement gratuit, une participation aux frais de reproduction pourra être demandée, mais elle ne pourra être supérieure au prix du papier et de l'encre composant le document.
  - Toute reproduction sortant du cadre précisé ci-dessus est interdite sans accord préalable écrit de l'auteur.



# Remerciements

---

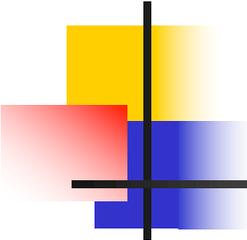
- Certains transparents sont basés sur des supports de cours de :
  - Danièle DROMARD (PARIS 6)
  - Andrzej DUDA (INP Grenoble/ENSIMAG)
  - Shivkumar KALYANARAMAN (RPI/ECSE)
  - Alain MILLE (LYON 1)
  - CongDuc PHAM (LYON 1)
  - Michel RIVEILL (Université de Nice/ESSI)
  - l' Institut National des Télécommunications (INT)
- Des figures sont issues des livres cités en bibliographie



# Bibliographie

---

- « *Réseaux* », 4ième édition, Andrew Tanenbaum, Pearson Education, ISBN 2-7440-7001-7
- « *Réseaux et Télécoms* », Claude Servin, Dunod, ISBN 2-10-007986-7
- « *Analyse structurée des réseaux* », 2ième édition, J. Kurose et K. Ross, Pearson Education, ISBN 2-7440-7000-9
- « *TCP/IP Illustrated Volume 1, The Protocols* », W. R. Stevens, Addison Wesley, ISBN 0-201-63346-9
- « *TCP/IP, Architecture, protocoles, applications* », 4ième édition, D. Comer, Dunod, ISBN 2-10-008181-0
- « *An Engineering Approach to Computer Networking* », Addison-Wesley, ISBN 0-201-63442-6
- Internet...
  - <http://www.guill.net/>
  - <http://www.courseforge.org/courses/>
  - <http://www.commentcamarche.net/ccmdoc/>
  - <http://www.rfc-editor.org/> (documents normatifs dans TCP/IP)



# Plan de la partie 4

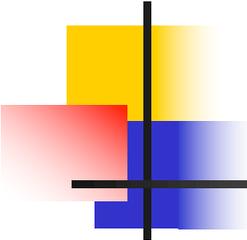
---

- Le protocole HDLC
- Le protocole PPP
- Le protocole SLIP



# Le protocole HDLC

---



# Le protocole HDLC

---

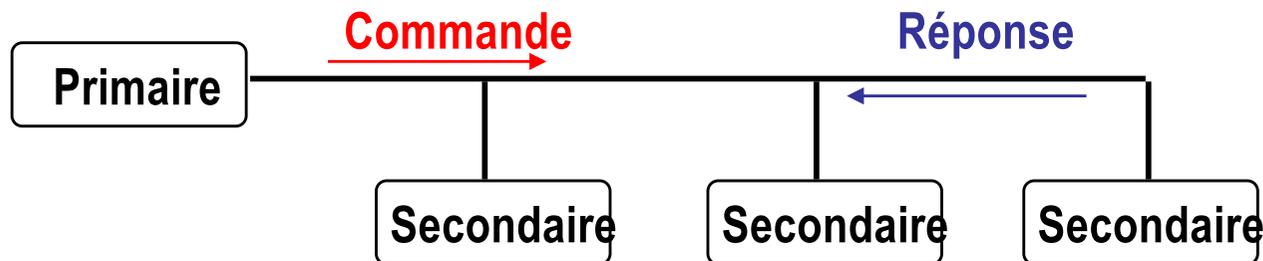
- HDLC - *High Level Data Link Control*
- Protocole de niveau 2 OSI orienté bits
- Premier protocole moderne, normalisé en 1976 par le CCITT (UIT-T)
  - utilisé par exemple dans le GSM entre le mobile et le réseau...
- Nécessite une liaison physique synchrone full-duplex (possibilité sur half-duplex mais avec des restrictions de service...)
- Liaison de données
  - point à point symétrique ou dissymétrique
  - multipoint dissymétrique (polling/selecting)

# Statut des stations

- Système à commande centralisée DISSYMETRIQUE

- Multipoint

**Adresse véhiculée dans les deux sens = celle du secondaire**



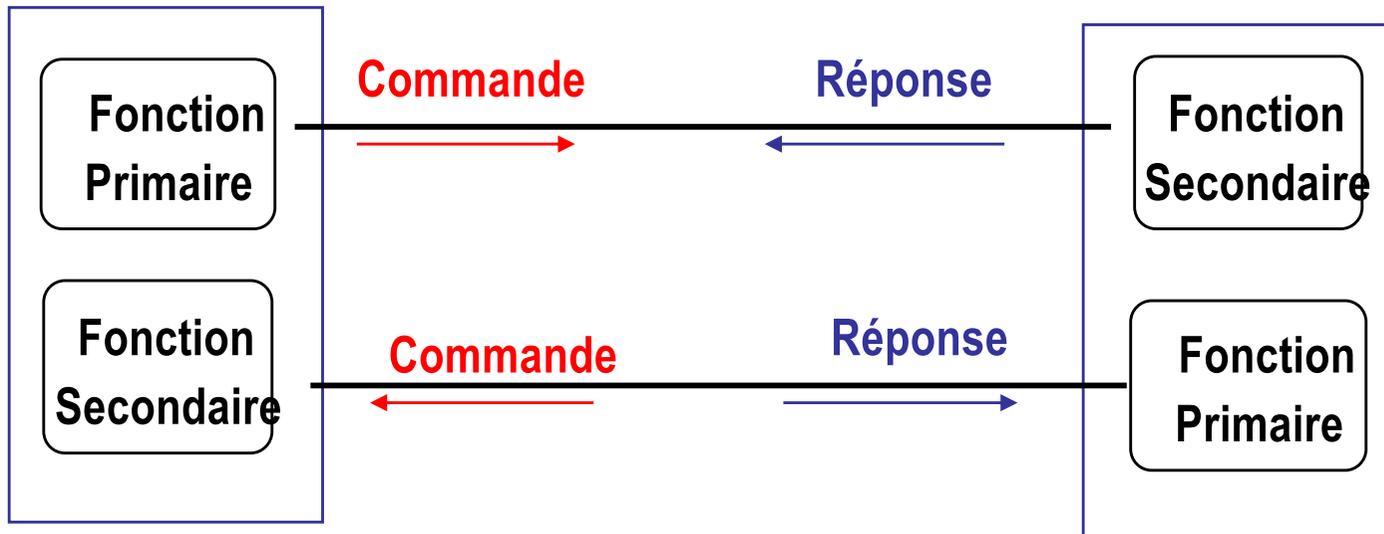
- Point à point

**Pas besoin de véhiculer une adresse !!**

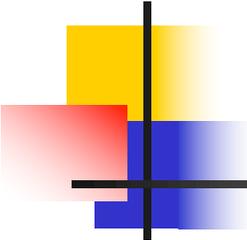


# Statut des stations

- Système à commande centralisée SYMETRIQUE (point à point)



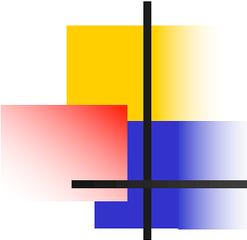
Adresse : FONCTION SECONDAIRE (par ex.)



# Les différentes modes de HDLC

---

- Le mode normal ou synchrone
  - **NRM** - *Normal Response Mode* ou **LAP** - *Link Access Protocol*
  - relation maître/esclave (le primaire invite le secondaire à parler)
  - liaison multipoint dissymétrique
- Le mode asynchrone
  - symétrique
    - **ABM** - *Asynchronous Balanced Mode*
    - chaque extrémité est primaire en émission et secondaire en réception (mode équilibré)
    - liaison point à point uniquement
      - full duplex (**LAP B**alanced - RNIS)
      - half duplex (**LAP X** - télétex)
  - Dissymétrique point à point ou multipoint
    - **ARM** - *Asynchronous Response Mode*
    - Le secondaire peut émettre sans y être autorisé
      - un seul secondaire actif à la fois
      - algorithme de résolution des collisions

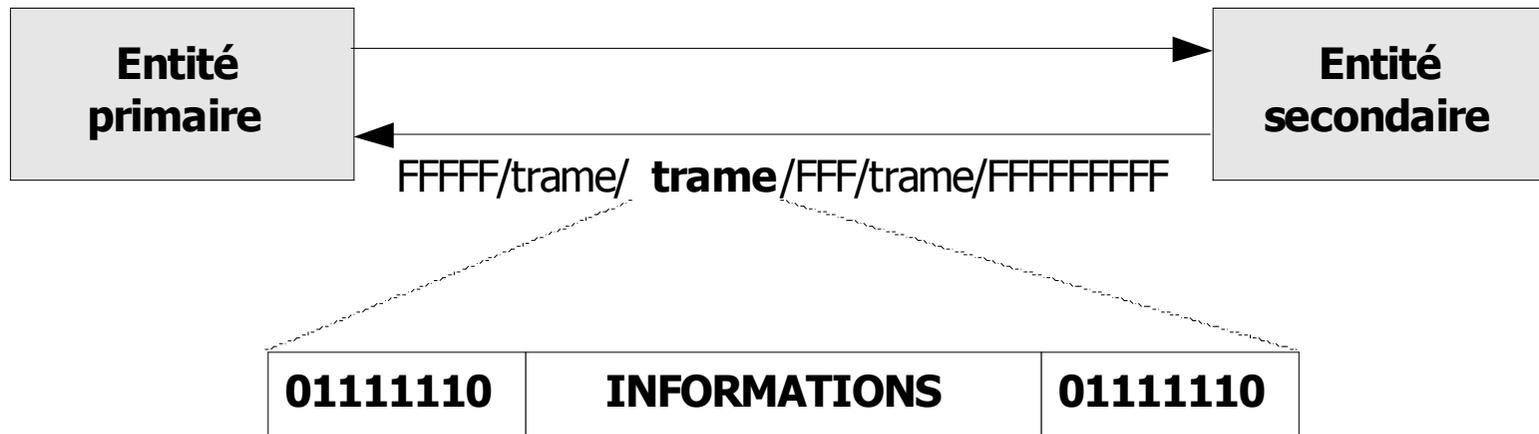


# Le protocole HDLC

---

- L'unité de transfert est la trame (*Frame*)
- Chaque trame est délimitée par un fanion - le seul caractère "spécial" utilisé par le protocole
- En l'absence de données, le fanion est envoyé pour maintenir la synchronisation entre les trames
- Contrôle d'erreur très efficace par CRC avec reprise sur erreur
- Utilise du contrôle de flux, de l'anticipation et du *piggybacking*

# Le protocole HDLC

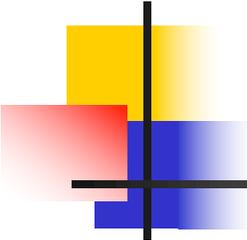


- L'entité est dite primaire si elle initie la communication
- Le fanion est représenté par la séquence "01111110"
- 3 types de trames - signalisation dans la bande
  - trame **I** - trame d'information (échange de données)
  - trame **S** - trame de supervision (supervision de l'échange)
  - trame **U** - trame non numérotées (supervision de la liaison)

# Structure de la trame HDLC

- Le fanion de queue peut faire office de fanion de tête de la trame suivante
- La transparence est assurée par la technique du bit de bourrage
- Un seul champ adresse (mode maître/esclave)
- Champ commande sur 1 ou 2 octets selon que les compteurs Ns/Nr sont stockés sur 3 ou 7 bits
- FCS = CRC relatif à Adresse/Commande/Infos

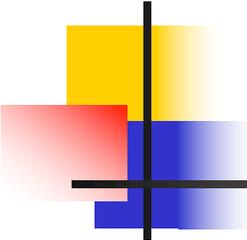
Fanion deb	Adresse	Commande	INFORMATIONS	FCS	Fanion fin
1 octet	1 octet	1 ou 2 octets		2 octets	1 octet



# Le champ adresse

---

- Il stocke l'adresse du secondaire
  - adresse du destinataire pour les trames de commande
  - adresse de l'émetteur pour les trames de réponse
- Si liaison point à point, le champ n' est pas lu !



# Types de trames

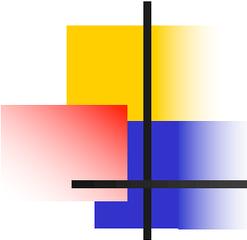
---

- Trames I
  - **Information** ; transfert de la SDU *System Data Unit*
- Trames S
  - Supervision séquencée - pas de champ information :
    - Contrôle de flux : RR, RNR
    - Contrôle d'erreurs : REJ (**Go-Back-N**), SREJ
- Trames U
  - Supervision non séquencée - pas de champ information :
    - **Connexion, Libération**
    - Anomalies, Réinitialisation
    - Test, Identification
    - **Données non séquencées (datagrammes)**

# Le champ "commande"

- **Ns** - compteur des trames **I** émises
- **Nr** - compteur des trames **I** reçues  
Nr contient le numéro de la prochaine trame attendue  
Nr = x acquitte les (x-1) trames précédentes
- Les bits **S** et **U** permettent d'identifier la commande

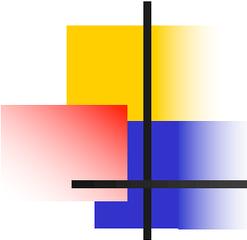
bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
<b>Nr</b>			<b>P/F</b>	<b>Ns</b>			<b>0</b>	trame <b>I</b>
<b>Nr</b>			<b>P/F</b>	<b>S</b>	<b>S</b>	<b>0</b>	<b>1</b>	trame <b>S</b>
<b>U</b>	<b>U</b>	<b>U</b>	<b>P/F</b>	<b>U</b>	<b>U</b>	<b>1</b>	<b>1</b>	trame <b>U</b>



# Le champ "commande"

---

- Le bit **P/F** - bit de contrôle
  - il est appelé P dans une trame de commande, F dans une trame de réponse
  - P (Poll) = demande de réponse explicite ou non émise par une station primaire
  - F (Final) = indication de réponse explicite émise par une station secondaire, suite à une demande explicite émise par un primaire
  - P=1 - sollicite une réponse explicite du secondaire (par ex. car fenêtre d'émission bientôt pleine)
  - Réponse à P=1 par F=1 - le secondaire répond par un acquittement
  - Une station qui reçoit une trame de commande avec le bit P/F=1 doit répondre avec P/F=1



# Le champ "commande"

---

- Trames de supervision

- **RR** - *Receive Ready*

- Prêt à recevoir ou accusé de réception si pas de trame I à envoyer par le récepteur

- **RNR** - *Receive Not Ready*

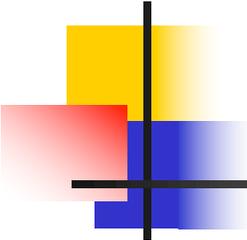
- Non prêt à recevoir ; le récepteur demande à l'émetteur d'arrêter les émissions et acquitte les trames jusqu'à Nr-1

- **REJ** - *Reject*

- Rejet simple ; demande la retransmission **à partir de Nr**

- **SREJ** - *Selective Reject*

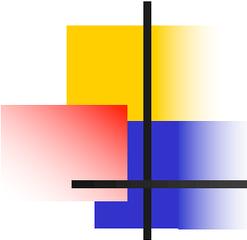
- Rejet sélectif ; demande la retransmission **de Nr**



# Le champ "commande"

---

- Trames non numérotées de commande (primaire vers secondaire)
  - **SNRM** - *Set Normal Response Mode*  
multipoint dissymétrique, maître/esclave (polling/selecting)
  - **SARM** - *Set Asynchronous Response Mode*  
le secondaire peut émettre sans invitation
  - **SABM** - *Set Asynchronous Balanced Mode*  
point à point symétrique, chaque station peut émettre sans autorisation (primaire et secondaire simultanément)
  - **SABME** - *Set Asynchronous Balanced Mode Extended*  
idem SABM mais mode étendu (compteurs sur 7 bits)
  - **DISC** - *Disconnect*  
l'un des terminaux décide de rompre la liaison



# Le champ "commande"

---

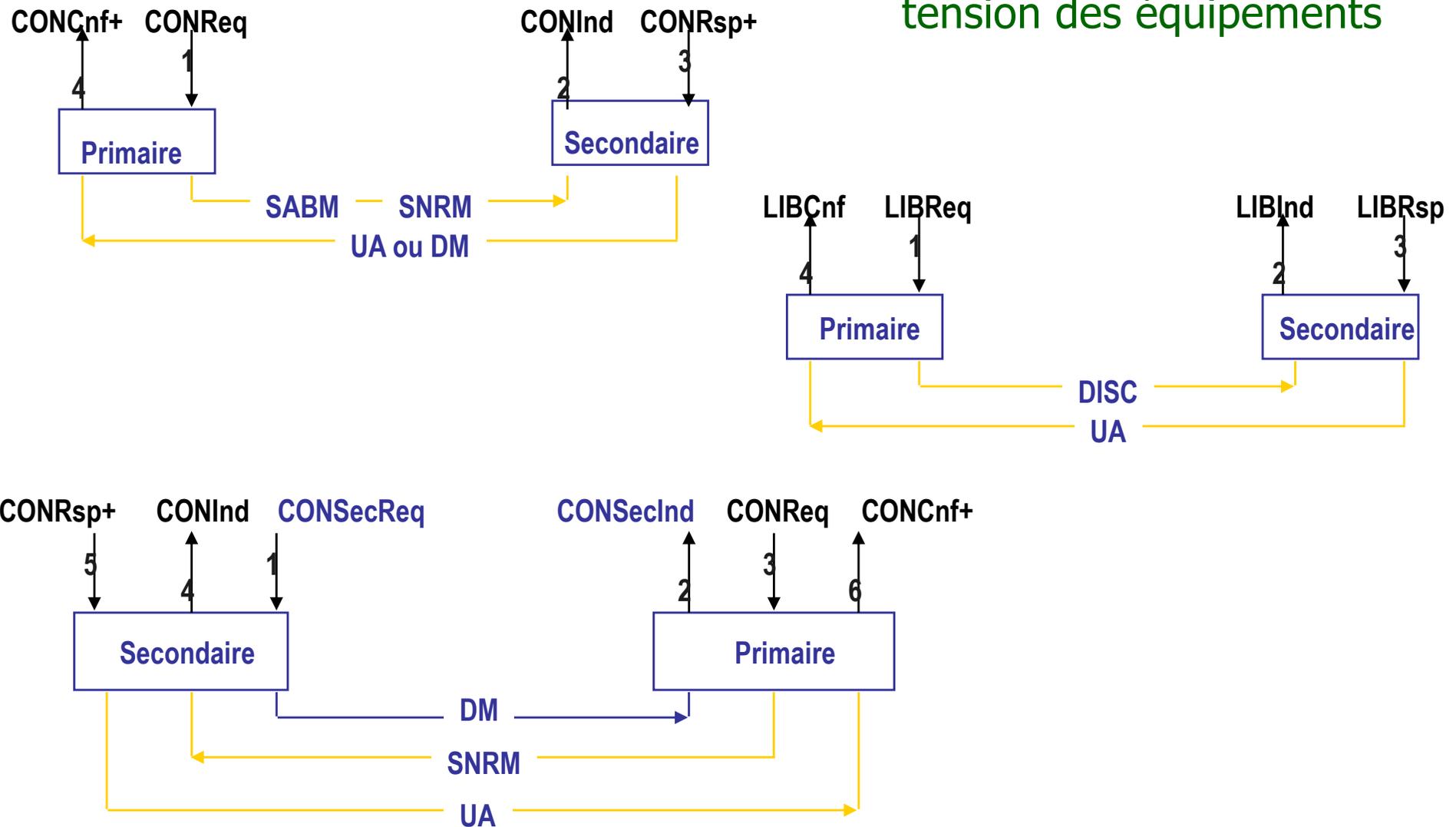
- Trames non numérotées de réponse (du secondaire vers le primaire)
  - **UA** - *Unnumbered Acknowledge*  
acquiescement d'une trame non numérotée
  - **FRMR** - *Frame Reject* ou **CMDR** - *Command Reject*  
une trame ou une commande vient d'être rejetée  
(indication de la raison dans le champ information)
  - **DM** - *Disconnect Mode*  
indique que le terminal est déconnecté
  - **RM** - *Request Disconnect*  
équivalent de DISC pour le secondaire

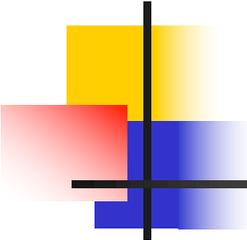
# Récapitulatif des principales commandes

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
<b>Nr</b>			<b>P/F</b>	<b>Ns</b>			<b>0</b>	trame <b>I</b>
<b>Nr</b>			<b>P/F</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>1</b>	<b>RR</b>
<b>Nr</b>			<b>P/F</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>1</b>	<b>RNR</b>
<b>Nr</b>			<b>P/F</b>	<b>1</b>	<b>0</b>	<b>0</b>	<b>1</b>	<b>REJ</b>
<b>Nr</b>			<b>P/F</b>	<b>1</b>	<b>1</b>	<b>0</b>	<b>1</b>	<b>SREJ</b>
<b>0</b>	<b>0</b>	<b>0</b>	<b>P</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>SARM</b>
<b>1</b>	<b>0</b>	<b>0</b>	<b>P</b>	<b>0</b>	<b>0</b>	<b>1</b>	<b>1</b>	<b>SNRM</b>
<b>0</b>	<b>0</b>	<b>1</b>	<b>P</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>SABM</b>
<b>0</b>	<b>1</b>	<b>1</b>	<b>P</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>SABME</b>
<b>0</b>	<b>1</b>	<b>0</b>	<b>P</b>	<b>0</b>	<b>0</b>	<b>1</b>	<b>1</b>	<b>DISC</b>
<b>0</b>	<b>1</b>	<b>1</b>	<b>F</b>	<b>0</b>	<b>0</b>	<b>1</b>	<b>1</b>	<b>UA</b>
<b>1</b>	<b>0</b>	<b>0</b>	<b>F</b>	<b>0</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>CMDR/FRMR</b>
<b>0</b>	<b>0</b>	<b>0</b>	<b>F</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>DM</b>

# Connexion et libération

Connexion à la mise sous tension des équipements

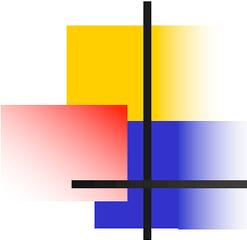




# Règles de reprise

---

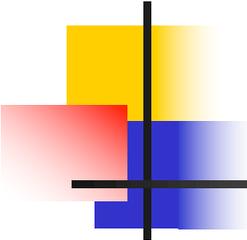
- Débordement (contrôle de flux)
  - la station qui ne peut plus recevoir de trames I émet un **RNR** dont le Nr indique la première trame non acceptée
  - elle émet une trame **RR** quand elle est de nouveau prête à recevoir de nouvelles trames
- Erreur de transmission
  - toute trame dont l'analyse du FCS indique une erreur de transmission est ignorée
  - aucune autre action n'est entreprise



# Règles de reprise

---

- Erreur de numéro de séquence Ns (version 1)
  - la trame dont le Ns n'est pas celui attendu est ignorée ainsi que les suivantes
  - la station recevant cette trame émet une trame **REJ** dont le numéro Nr indique la trame attendue
- Erreur de numéro de séquence Ns (version 2)
  - la trame dont le Ns n'est pas celui attendu est conservée
  - la station recevant cette trame émet une trame **SREJ** pour demander la retransmission de la trame Nr
- Rejet de commande
  - à la réception d'une trame CMDR/FRMR, la liaison est considérée comme rompue



# Règles de reprise

---

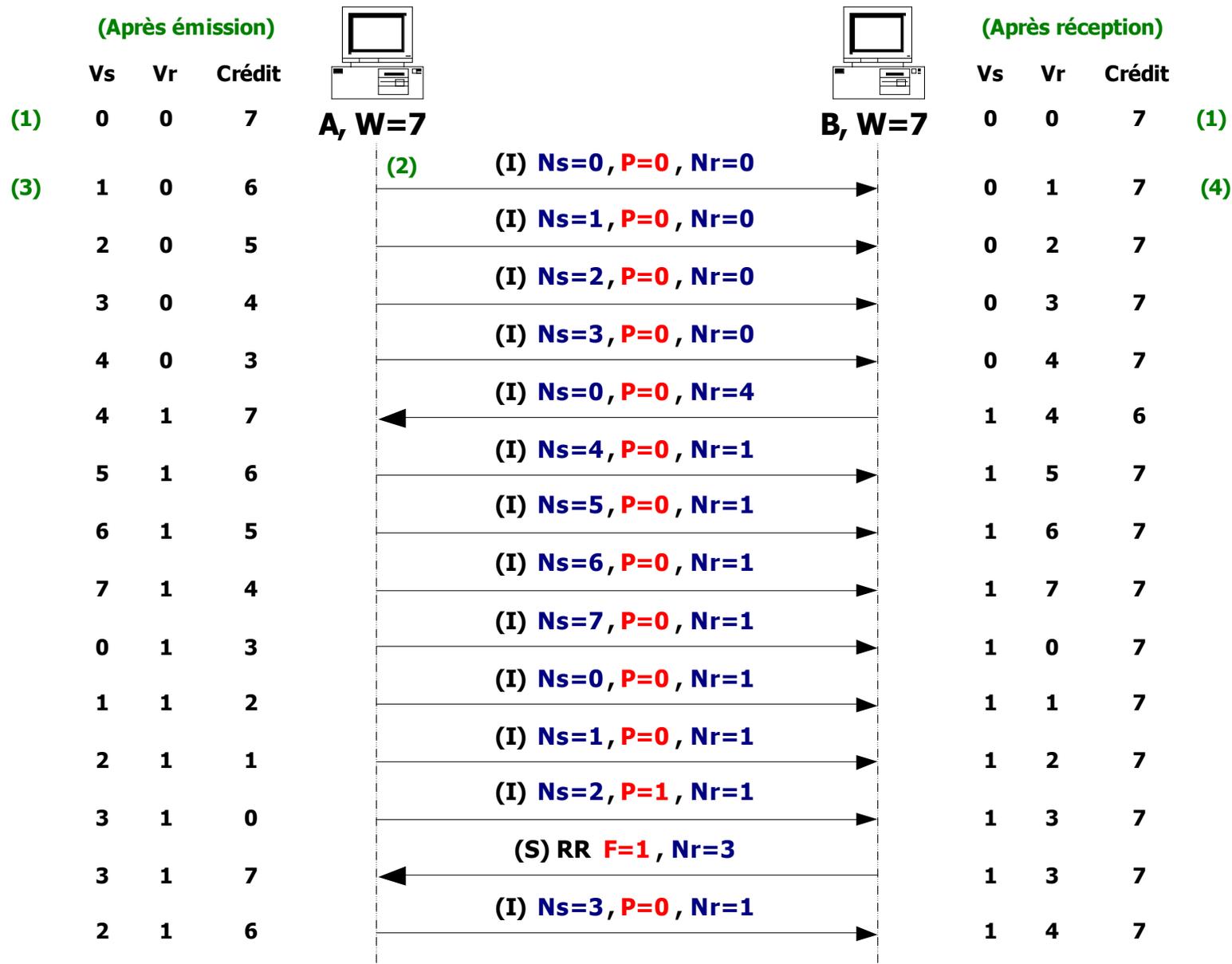
- Deux temporisateurs

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

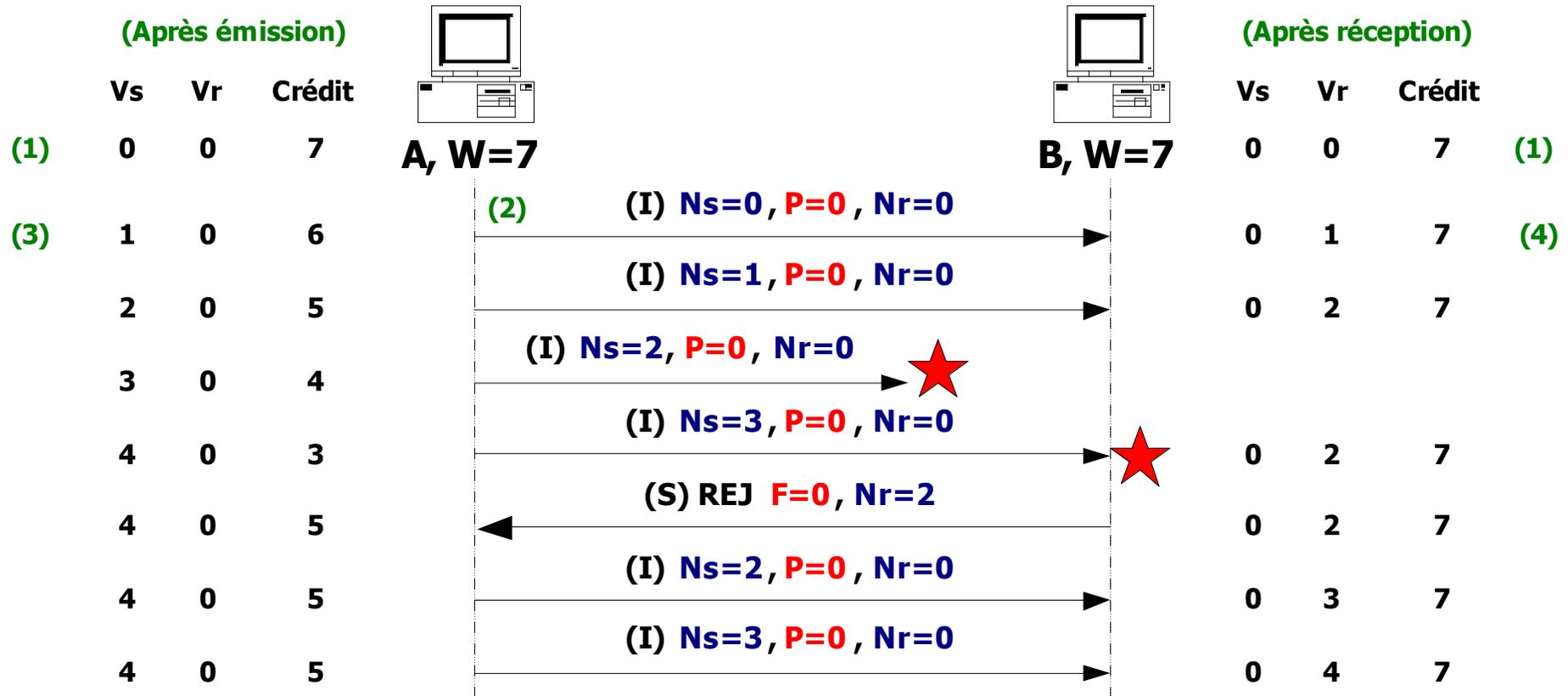
- Reprise sur time-out

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

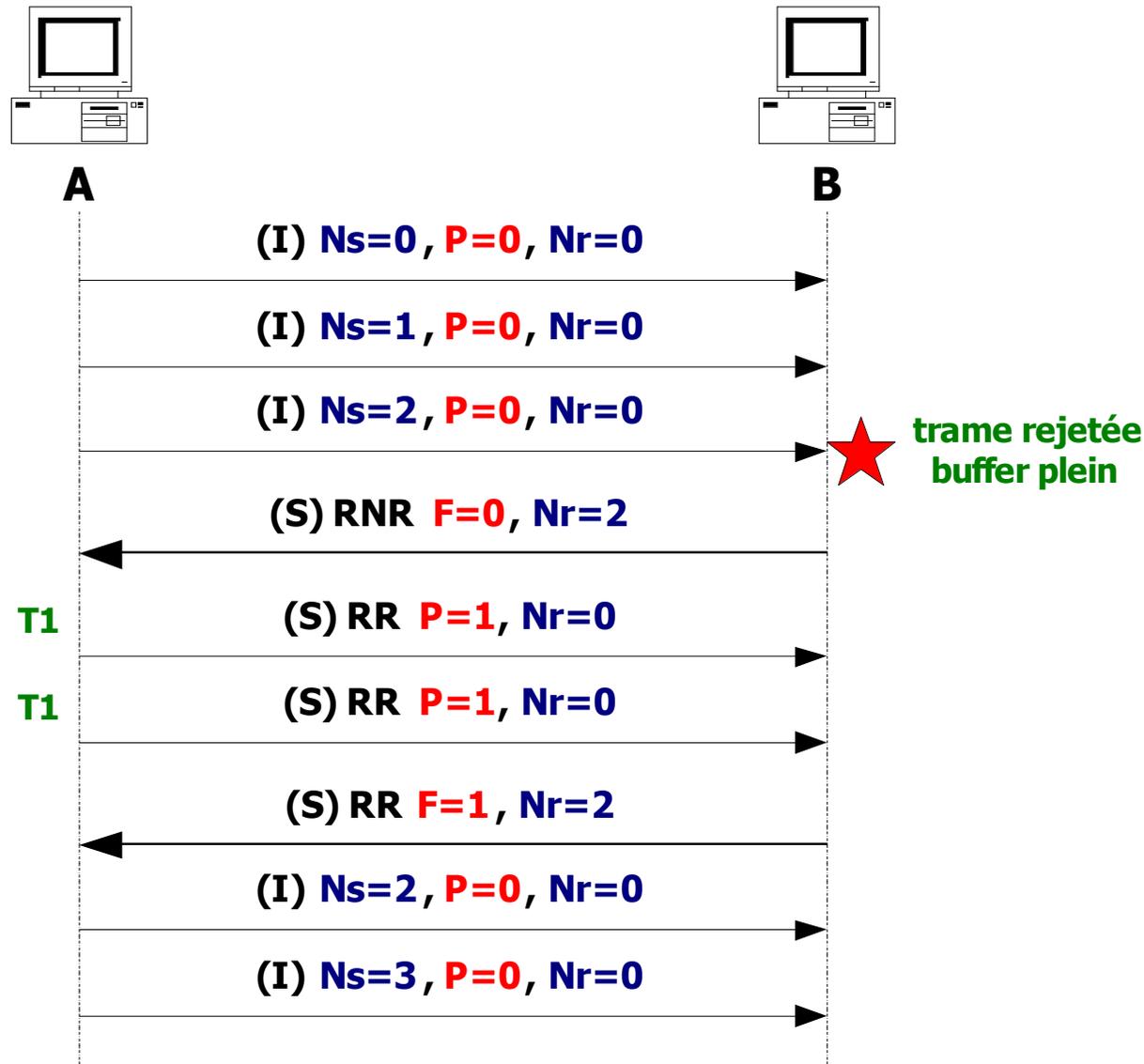
# Exemple d'échange de données



# Exemple de gestion d'erreurs



# Exemple de gestion du contrôle de flux



# Gestion sautante de la fenêtre

0	1	2	3	4	5	6	7	0	1	2	3	4	
---	---	---	---	---	---	---	---	---	---	---	---	---	--

0	1	2	3	4	5	6	7	0	1	2	3	4	
---	---	---	---	---	---	---	---	---	---	---	---	---	--

0	1	2	3	4	5	6	7	0	1	2	3	4	
---	---	---	---	---	---	---	---	---	---	---	---	---	--

0	1	2	3	4	5	6	7	0	1	2	3	4	
---	---	---	---	---	---	---	---	---	---	---	---	---	--

## ■ EXEMPLE $W=3$

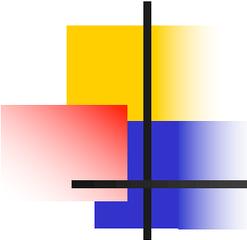
- on peut émettre 0, 1, 2
- on reçoit trame RR demandant 3
- on peut émettre 3, 4, 5
- on reçoit trame RR demandant 6
- on peut émettre 6, 7, 0
- on reçoit trame RR demandant 1
- on peut émettre 1, 2, 3
- etc ...



## Cours 4 (suite)

---

Les protocoles SLIP et PPP

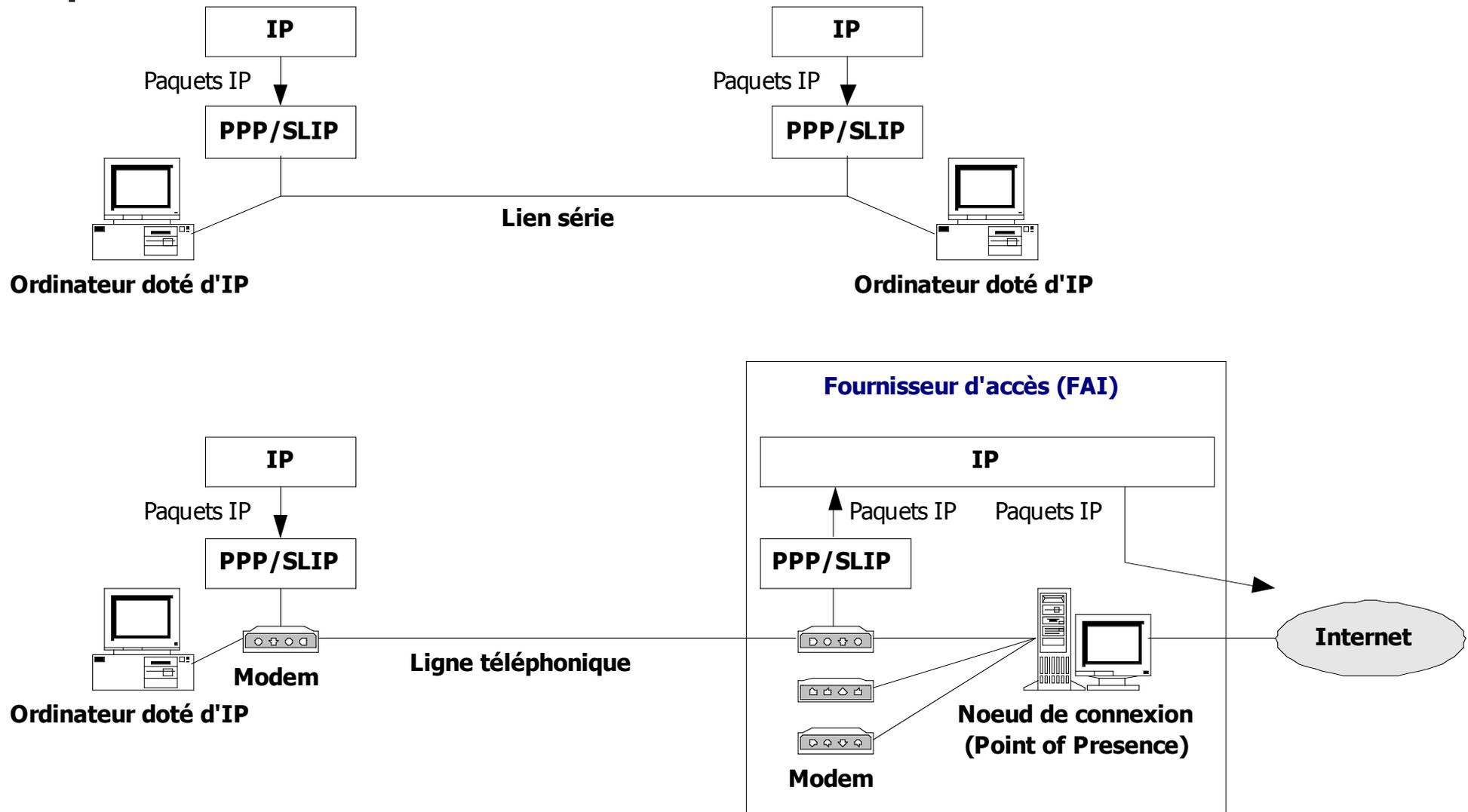


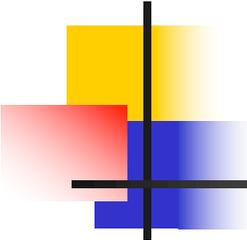
# Les protocoles SLIP et PPP (1)

---

- SLIP et PPP sont des protocoles de niveau liaison pour les réseaux Internet
  - SLIP : *Serial Line Internet Protocol*
  - PPP : *Point to Point Protocol*
- SLIP et PPP ne sont pas normalisés
  - SLIP : RFC 1055
  - PPP : RFC 1661
- Ils sont utilisés pour assurer des connexions IP sur des liaisons série
  - des connexions à Internet par modem
  - des connexions point à point entre deux ordinateurs directement connectés l'un à l'autre (i.e. pas besoin de carte *Ethernet*)

# Les protocoles SLIP et PPP (2)

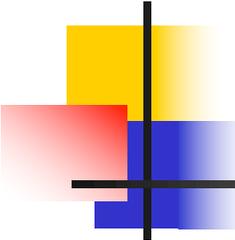




# Le protocole SLIP (1)

---

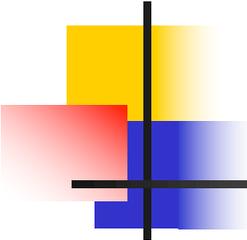
- Protocole asynchrone orienté caractère
- Fonctionnalités restreintes
  - pas de négociation à l'ouverture de la connexion (adresses IP, informations DNS, mode de SLIP utilisé fixés manuellement avant le démarrage de SLIP)
  - ne fonctionne qu'avec IPv4 (IPv6 et X.25 ne sont pas pris en charge)
  - pour des adresses fixes connues à l'avance
  - pas de détection ou correction d'erreurs (ces fonctions doivent être fournies par les couches supérieures)
  - trop de versions existent, il n'y a donc pas de standard de l'Internet



## Le protocole SLIP (2)

---

- Délimitation des paquets IP par insertion du caractère de contrôle END avant et après chaque paquet
- Transparence assurée par la technique du *character stuffing*
  - 3 caractères spéciaux supplémentaires (ESC, ESC\_END et ESC\_ESC)
  - côté émission :
    - si le caractère END apparaît dans les données, SLIP le remplace par la séquence ESC - ESC\_END
    - si le caractère ESC apparaît dans les données, SLIP le remplace par la séquence ESC - ESC\_ESC

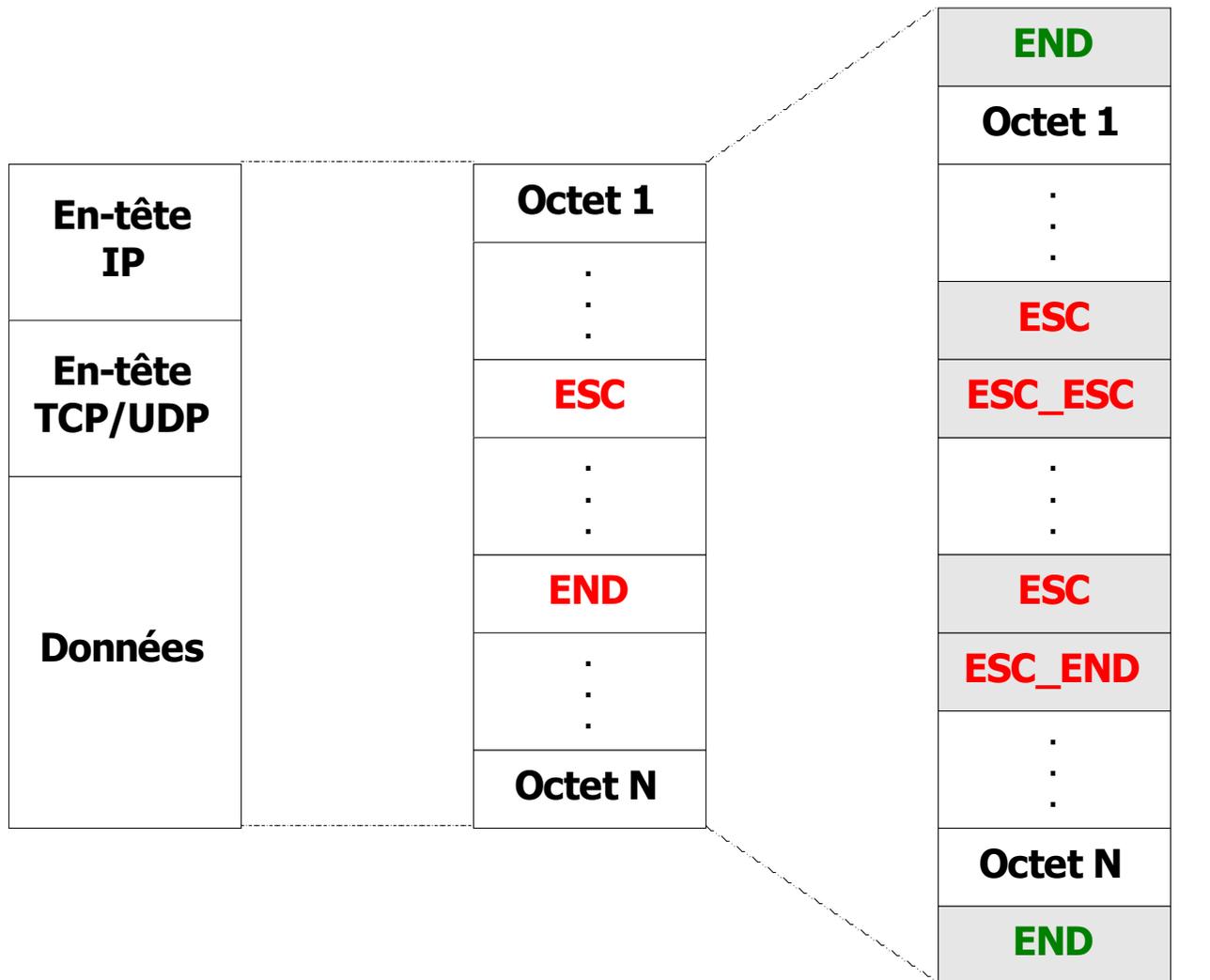


# Le protocole SLIP (3)

---

- Transparence assurée par la technique du *character stuffing*
  - côté réception :
    - si le caractère ESC apparaît, le caractère suivant doit être ESC\_END ou ESC\_ESC ; dans ce cas, le caractère ESC est supprimé et le caractère suivant est remplacé par ESC ou END
    - si le caractère END apparaît, il s'agit de la fin du paquet IP ; celui-ci est alors délivré à la couche IP supérieure

# Le protocole SLIP (4)



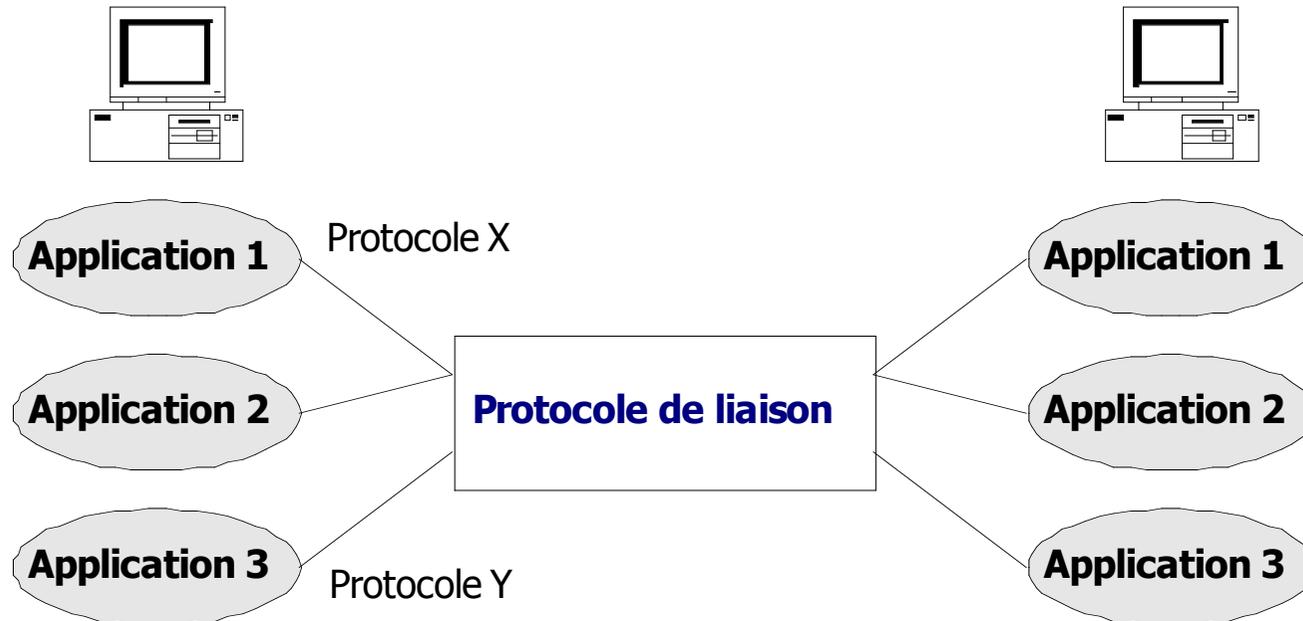
**Paquet IP**

**Paquet IP sous forme  
d'une séquence d'octets**

**Flux d'octets SLIP**

# Le protocole PPP (1)

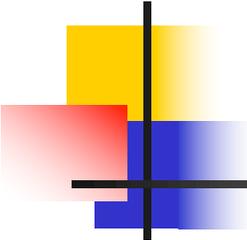
- PPP, contrairement à HDLC par exemple, est multi-protocoles
  - un champ spécifique dans la trame PPP permet de distinguer le protocole utilisé par les données transportées (le protocole X ou Y est encapsulé dans la trame PPP)



# Le protocole PPP (2)

- Utilisé sur des liaisons point à point synchrones ou asynchrones
- Format de la trame
  - délimitation des trames par le fanion 0x7E
  - champ adresse inutile (point à point) -> 0xFF
  - champ contrôle identique à celui de HDLC
  - champ protocole sur 2 octets identifie le protocole de niveau supérieur (IP, IPX, IPv6, ...)
  - champ FCS identique à celui de HDLC

<b>Fanion 0x7E</b>	<b>Adresse 0xFF</b>	<b>Commande</b>	<b>Protocole</b>	<b>Données</b>	<b>FCS</b>	<b>Fanion 0x7E</b>
------------------------	-------------------------	-----------------	------------------	----------------	------------	------------------------

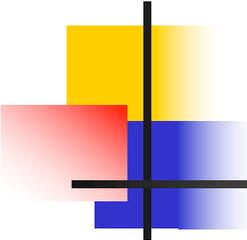


# Le protocole PPP (3)

---

## ■ Transparence

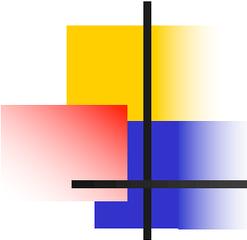
- technique du bit de bourrage (insertion d'un 0 tous les 5 bits à 1) sur liaison synchrone
- sur une liaison asynchrone
  - le fanion dans les données est remplacé par la séquence ESC, ESC\_FLAG
  - le caractère d'échappement est remplacé par la séquence ESC, ESC\_ESC
  - pour éviter que les modems ne considèrent comme une commande les caractères ASCII dont la valeur est inférieure à 32, des caractères de transparence ESC\_Commande sont ajoutés (les caractères dont la transparence doit être assurée sont négociés à la connexion)



# Le protocole PPP (4)

---

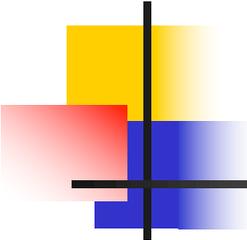
- PPP utilise un ensemble de sous protocoles pour la négociation de paramètres et la sécurisation des échanges
  - LCP (*Link Control Protocol*) - négociation des paramètres de l'échange
  - PAP (*PPP Authentication Protocol*) - authentification de l'utilisateur par mot de passe avant le transfert de données
  - CHAP (*Challenge Authentication Protocol*) - authentification sécurisée de l'utilisateur (cryptage)
  - NCP (*Network Control Protocol*) - négociation des paramètres réseau : affectation d'adresses IP, compression d'en-tête



# Comparaisons SLIP et PPP (1)

---

- SLIP beaucoup plus ancien que PPP
- SLIP n'est pas un standard de l'Internet (beaucoup de versions différentes de SLIP) alors que PPP est un standard de l'IETF
- SLIP ne fonctionne que sur des liaisons asynchrones
- Les fonctionnalités de PPP sont plus nombreuses (détection d'erreurs, authentification de l'émetteur, attribution d'adresses IP)



# Comparaisons SLIP et PPP (2)

---

- Compatibilité
  - SLIP est uniquement compatible IPv4
  - PPP est compatible IPv4, IPv6, X.25, ...
- Compression des données utiles des paquets IP
  - permet d'améliorer l'exploitation des liaisons lentes par modem
  - SLIP : compression des en-têtes des paquets IP uniquement (mode CSLIP)
  - PPP : compression possible des données utiles des paquets IP