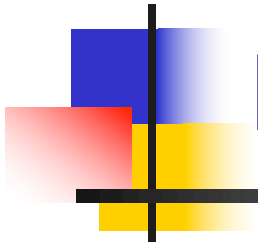
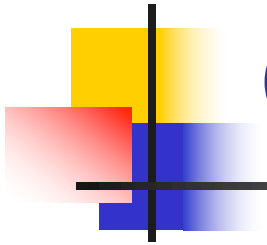
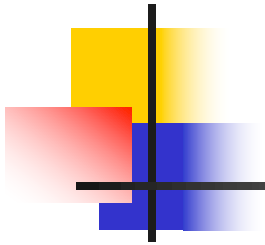


Partie 3 : Notions de protocoles

Olivier GLÜCK
Université LYON 1 / Département Informatique
Olivier.Gluck@univ-lyon1.fr
http://perso.univ-lyon1.fr/olivier.gluck

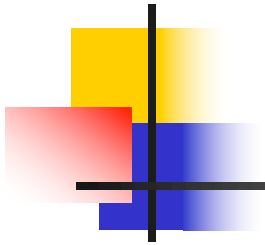


Copyright



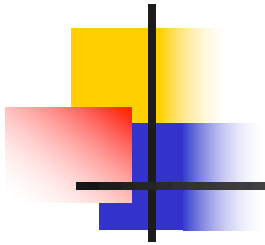
Remerciements

- 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



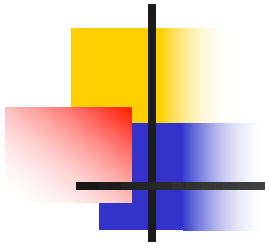
Bibliographie

- « *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



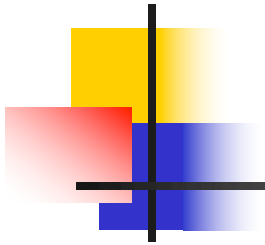
Bibliographie

- Internet…
  - http://www.guill.net/
  - http://www.courseforge.org/courses/
  - http://www.commentcamarche.net/ccmdoc/
  - http://www.protocols.com/
  - http://dir.yahoo.com/Computers_and_Internet/
  - http://www.rfc-editor.org/ (documents normatifs dans TCP/IP)



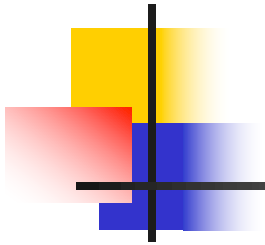
Plan de la partie 3

- La délimitation des données
  - Notion de fanion
  - Notion de transparence
- Le contrôle d'intégrité
  - Notion d'erreur
  - Détection d'erreur par clé calculée
  - Les codes autocorrecteurs
- Le contrôle de l'échange
  - Du mode Send & Wait aux protocoles à anticipation
  - Contrôle de flux
- La signalisation



Rappel : un protocole

- Un ensemble de conventions préétablies pour réaliser un échange (fiable) de données entre deux entités
- Il définit le format des en-têtes et les règles d'échange
  - syntaxe et sémantique des messages…
- En particulier :
  - délimitation des blocs de données échangés
  - contrôle de l'intégrité des données reçues
  - organisation et contrôle de l'échange
  - éventuellement, contrôle de la liaison



Rappel : rôle de la liaison de données

- Transfert de données fiable entre deux équipements de liaison
  - Taux d'erreurs résiduel négligeable (détection et contrôle des erreurs de la couche physique)
  - Sans perte (contrôle de flux)
  - Sans duplication
  - Maintien des trames en séquence (dans l'ordre !)
- Service fourni au réseau
  - Etablir, maintenir et libérer les connexions de liaison de données entre entités de réseau
- Service bi-point et multipoint
  - En multipoint : la LD gère l'accès au support (CSMA/CD)

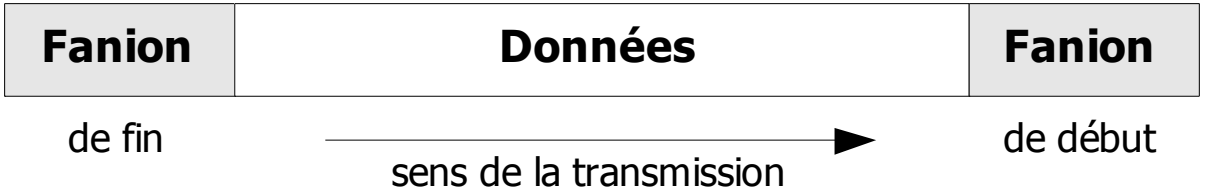
La délimitation des données

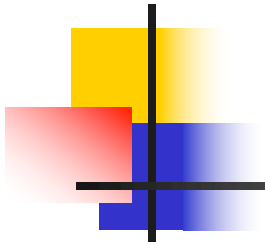
Notion de fanion

Notion de transparence

Notion de fanion

- Lors d'une transmission de données, il faut pouvoir repérer le début et la fin de la séquence des données transmises
  - bit de "start" et bit de "stop" en transmission asynchrone
  - fanion en transmission synchrone
    - un caractère spécial
    - ou une séquence de bits particulière

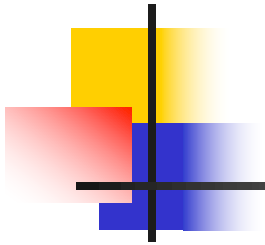




Notion de fanion

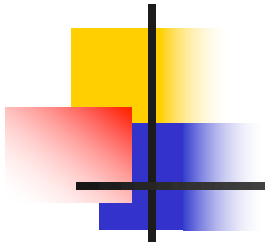
- 3 fonctions essentielles
  - délimite les données
  - permet de maintenir la synchronisation de l'horloge de réception (émis en l'absence de données à émettre)
  - permet au récepteur de se caler correctement sur une frontière d'octets (synchronisation caractère)

    -> reconnaissance des caractères
- Question
  - Qu'est ce que la définition d'un caractère spécial pose comme problème ?



Notion de transparence

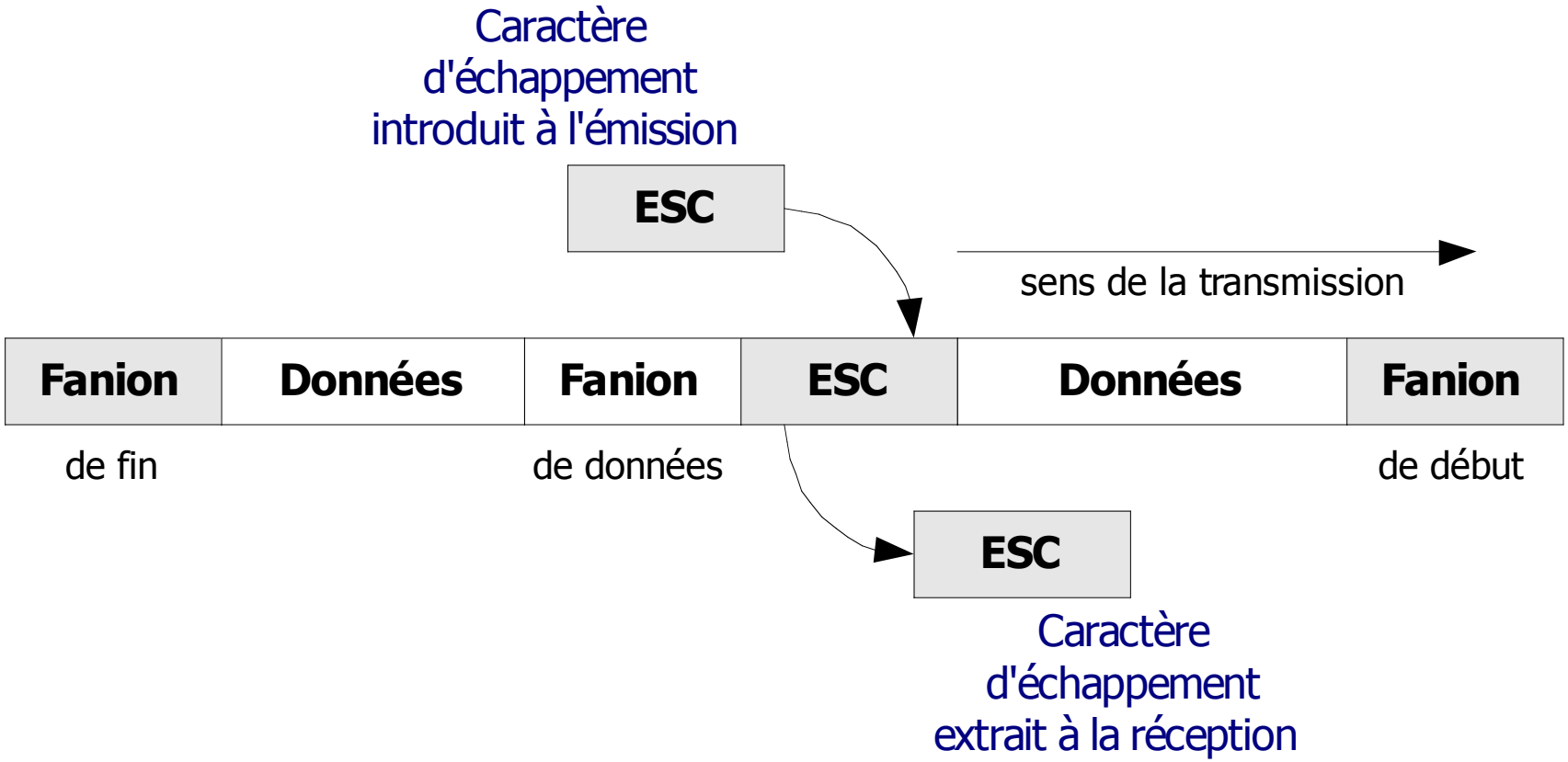
- Les caractères "spéciaux" comme le fanion ne sont pas délivrés aux couches supérieures : ils sont interprétés pour les besoins du protocole
- Les caractères "spéciaux" doivent pouvoir être transmis en tant que données et donc délivrés en tant que tel
  - -> mécanismes de transparence
  - -> définition d'un autre caractère spécial : le **caractère d'échappement**



Notion de transparence

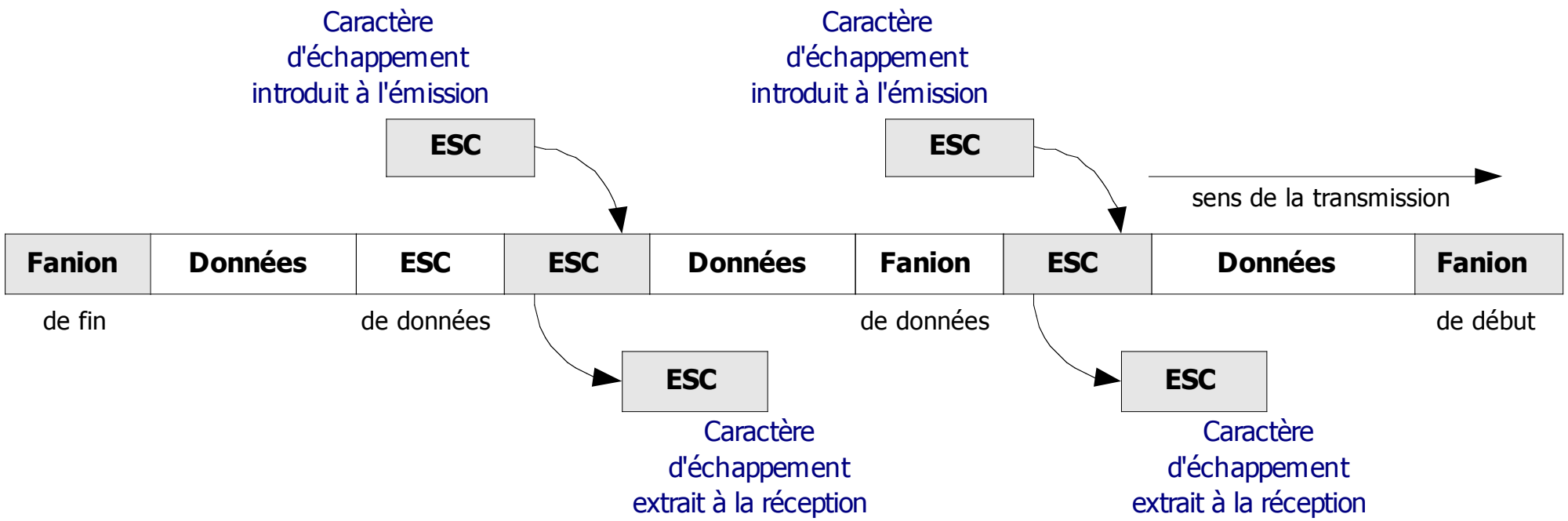
- Caractère d'échappement : le caractère suivant n'est pas interprété
- Fonctionnement
  - Côté émission : insertion du caractère d'échappement devant le caractère à protéger
  - Côté réception : l'automate examine chaque caractère pour découvrir le fanion de fin ; s'il rencontre le caractère d'échappement, il l'élimine et n'interprète pas le caractère suivant -> il le délivre au système

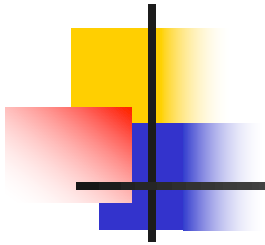
Notion de transparence



- Et si on veut transmettre le caractère d'échappement en tant que données ?

Notion de transparence

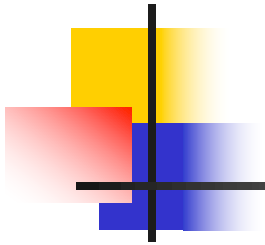




Protocoles orientés caractères/bits

- Protocoles orientés caractères
  - trame=nb entier de caractères délimités par des caractères de commande
  - tous les caractères "de commande", dédiés au contrôle de l'échange, sont représentés par un caractère spécial qui doit être systématiquement précédé d'un caractère d'échappement
- Protocoles orientés bits
  - les informations de contrôle sont dans un champ particulier de la trame → il faut assurer la transparence pour le fanion uniquement
  - seul le fanion est un "caractère" spécial
  - la transparence binaire est assurée par l'insertion d'un "0" tous les 5 bits à "1"
  - le fanion est représenté par "01111110" ; c'est la seule séquence pouvant contenir plus de 5 bits à "1" consécutifs -> **technique du bit de bourrage**

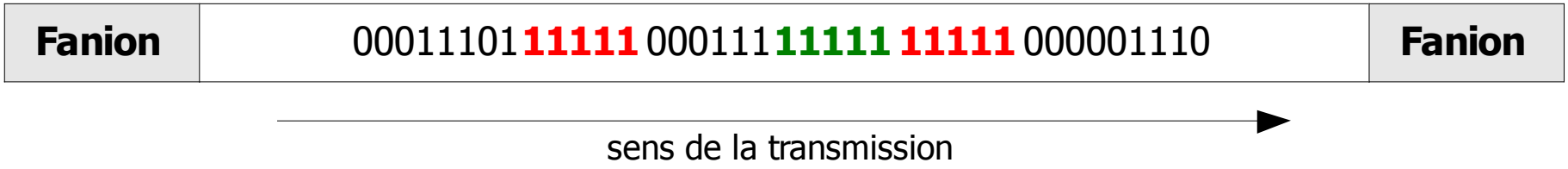
La technique du bit de bourrage



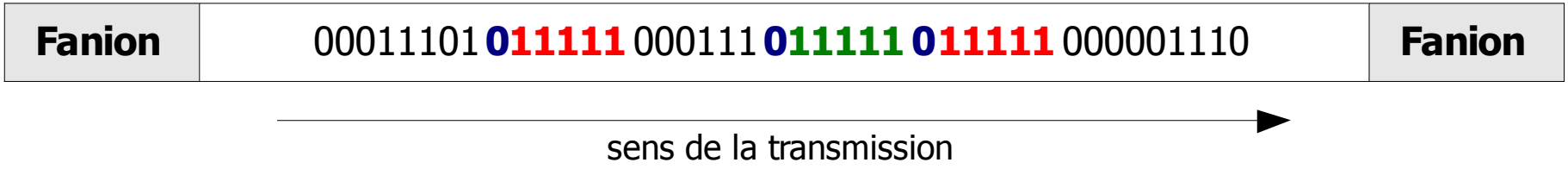
- Seul le fanion (01111110) peut contenir plus de 5 bits consécutifs à "1"
- Côté émission : si 5 bits consécutifs sont à "1", l'automate insère un "0"
- Côté réception : si 5 bits consécutifs sont à 1, l'automate regarde le bit suivant :
  - s'il est à "1", il s'agit du fanion
  - s'il est à "0", le "0" est enlevé de la séquence (il a été introduit à l'émission)
- Permet la resynchronisation des horloges en interdisant les longues séquences de bits à 1

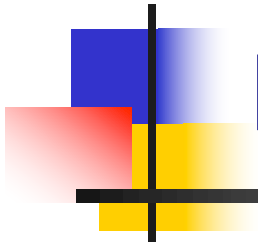
La technique du bit de bourrage

Séquence originale



Séquence transmise



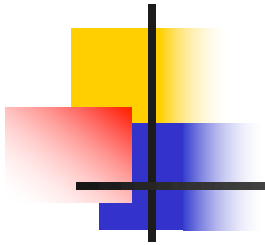


Le contrôle d'intégrité

Notion d'erreur

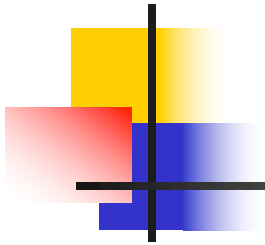
Détection d'erreur par clé calculée

Codes autocorrecteurs



Le contrôle d'intégrité

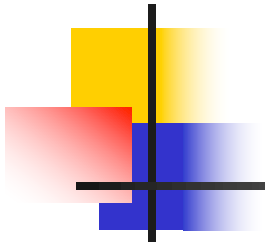
- Plusieurs facteurs peuvent modifier le contenu des données
  - facteurs d'origine humaine
    - problème de sécurité des données
    - transmission de mots de passe chiffrés...
  - facteurs d'origine physique : des bits sont erronés
    - on parle de contrôle d'erreur
    - erreurs dues à un phénomène physique
      - rayonnements électromagnétiques
      - distorsions
      - bruit
      - perte de la synchronisation des horloges (fibre)



Le taux d'erreur binaire (BER)

- BER = *Bit Error Rate*
- Teb = Nb bits erronés / Nb bits transmis
- Exemple
  - L'émetteur transmet la suite 0001110101101010
  - Le récepteur reçoit la suite 0001100101111011
  - Teb = 3/16 = 0,1875
- En pratique
  - RTC : Teb=10^{-4}
  - Réseaux locaux : Teb=10^{-9}
  - Les erreurs se produisent généralement par rafale

Le taux d'erreur binaire (BER)



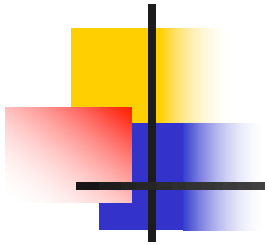
- Teb représente la probabilité de recevoir un bit erroné
- La probabilité de recevoir correctement un bloc de N bits est alors :

  $p = (1\text{-}Teb)\ldots(1\text{-}Teb) = (1\text{-}Teb)^N$

- La probabilité de recevoir un bloc erroné est alors :

  $p = 1 - (1\text{-}Teb)^N$

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

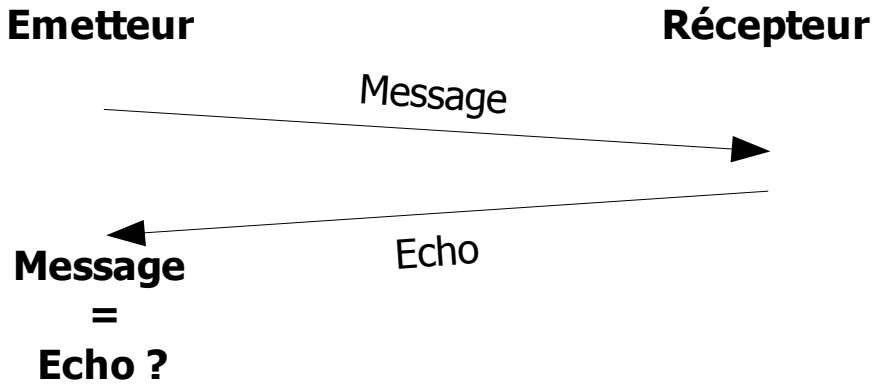


La détection d'erreur

- But : vérifier la validité des données reçues chez le destinataire
- Idée : ajouter une certaine redondance dans l'information transmise
- 4 techniques
  - la détection par écho
  - la détection par répétition
  - la détection d'erreur par clé calculée
  - la détection et correction d'erreur par code

La détection par écho

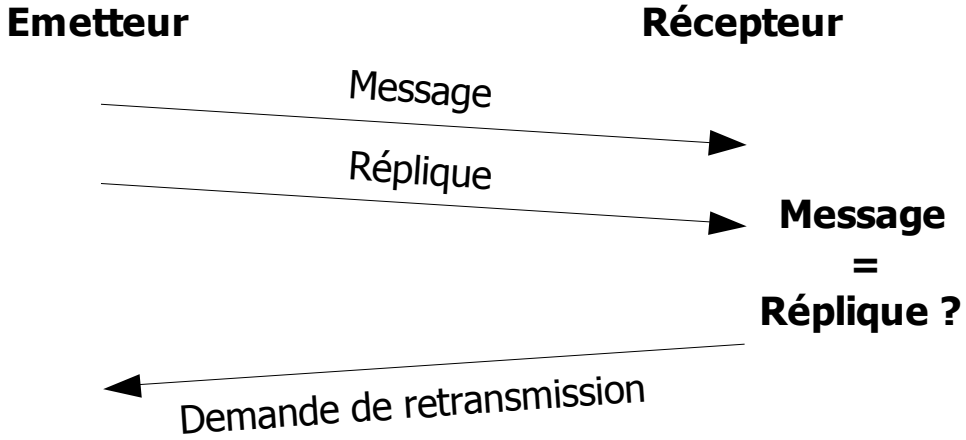
- Le récepteur renvoie chaque message reçu (écho)
- L'émetteur compare l'écho au message initial et le renvoie si les deux messages sont différents



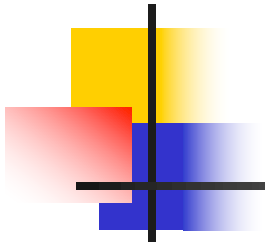
- Utilisée par terminaux asynchrones (telnet, minitel, …)
- Problèmes
  - redondance totale
  - l'écho peut lui-même être erroné

La détection par répétition

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



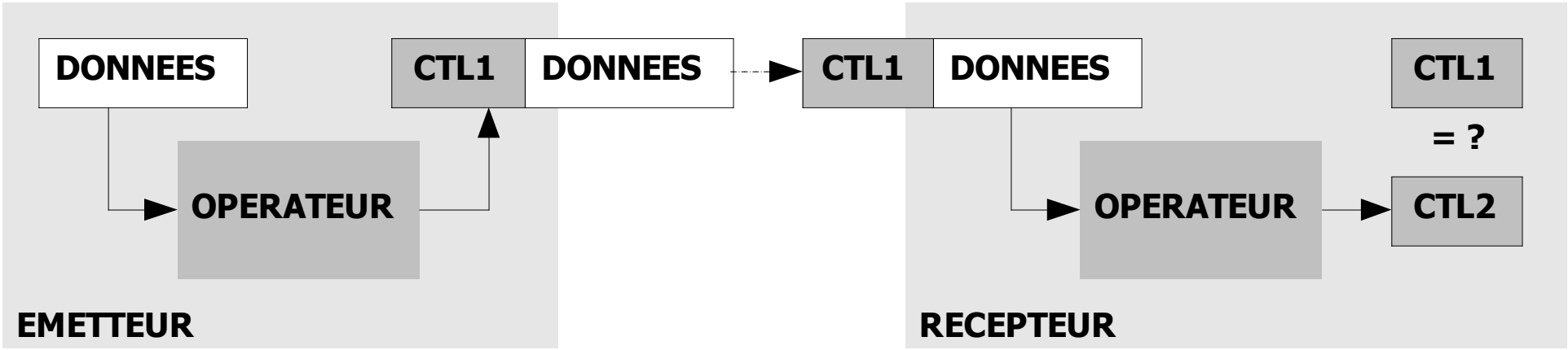
- Utilisée dans les milieux sécurisés très perturbés (applications temps réel)
- Problèmes
  - redondance totale
  - la réplique peut être erronée
  - contrôle sur le récepteur



La détection d'erreur par clé calculée

- L'émetteur ajoute au message une information supplémentaire (clé) calculée à partir du message d'origine
- Le récepteur recalcule la clé selon la même méthode à partir des informations reçues et compare à la clé reçue
- Le récepteur ignore les données si les clés sont différentes et peut demander la retransmission (reprise sur erreur)

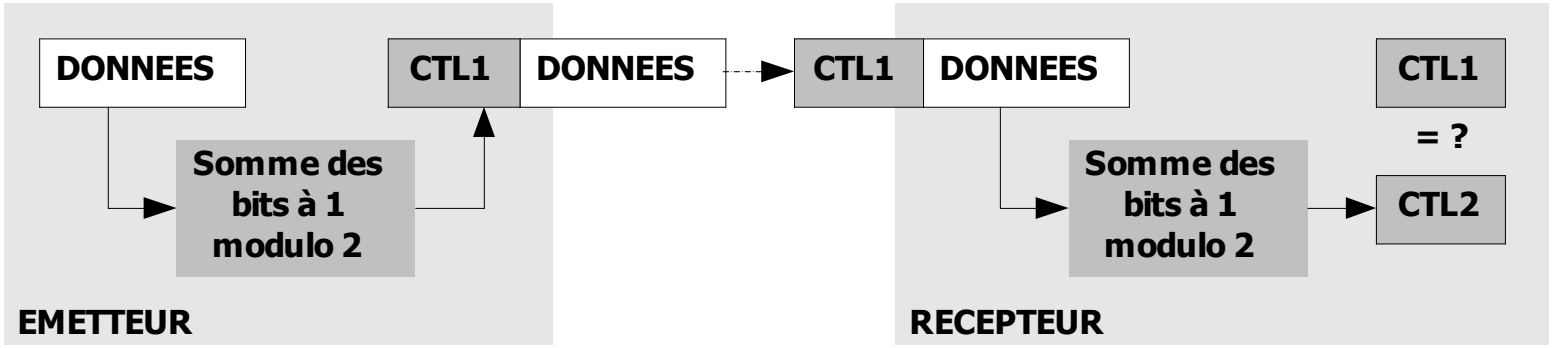
La détection par clé calculée



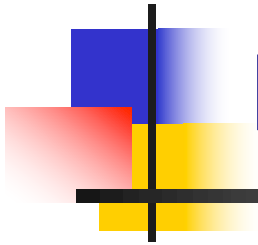
- La clé est parfois appelée
  - CRC : *Cyclic Redundancy Check*
  - FCS : *Frame Check Sequence*
- La clé peut elle-même être corrompue

La détection par clé calculée

- Un exemple : la technique du bit de parité



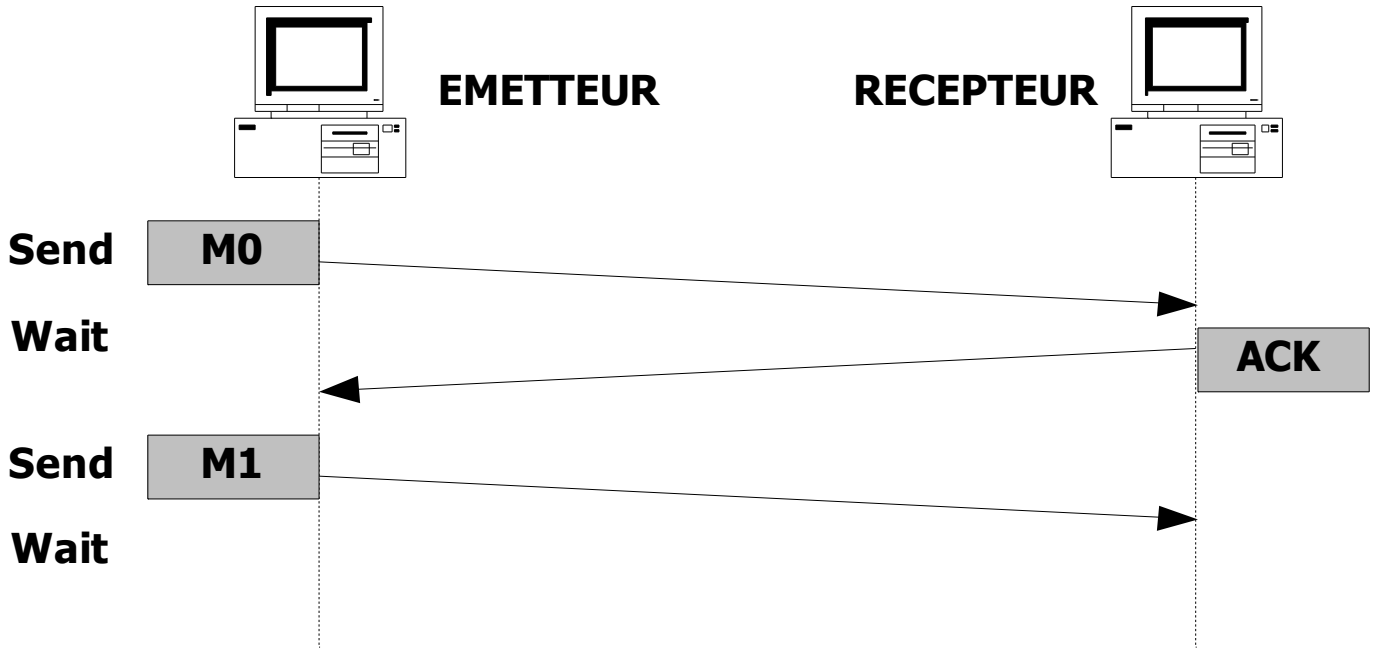
  - Exemple : S en ASCII est représenté par 1010011 -> bit de parité = 0
  - Simple mais Redondance faible
  - Ne permet de détecter que les erreurs portant sur un nombre impair de bits
  - Utilisé pour la transmission des caractères ASCII



Le contrôle de l'échange

Les mécanismes de base

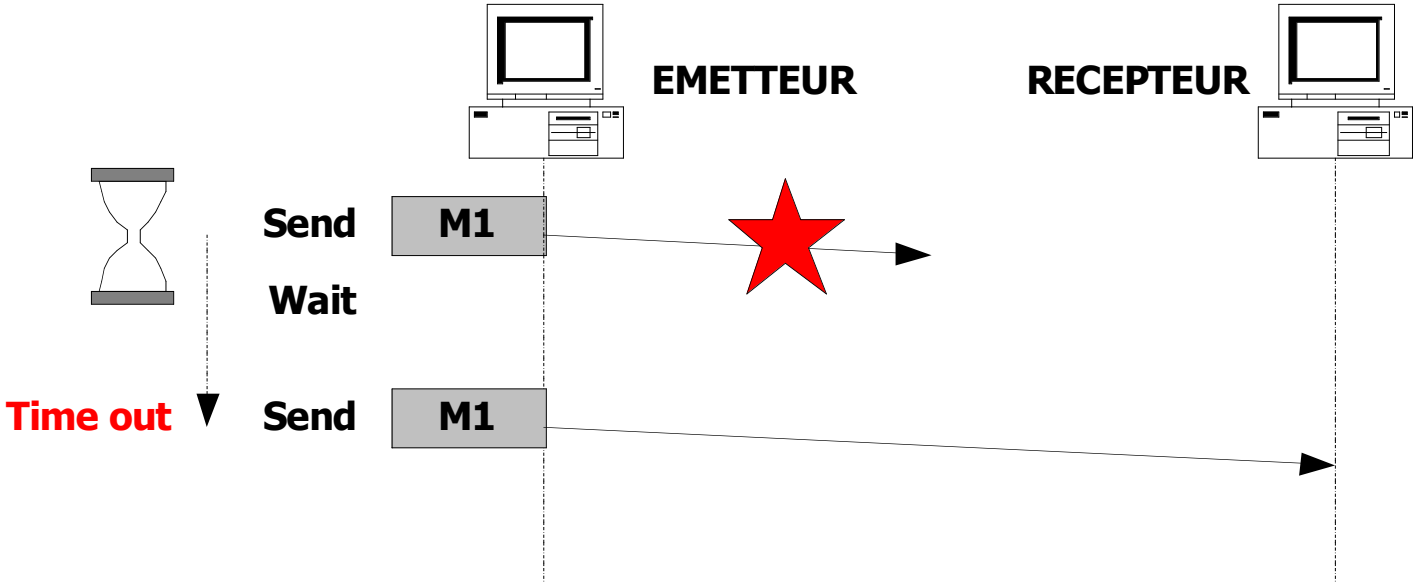
- Le mode *Send & Wait*



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

Les mécanismes de base

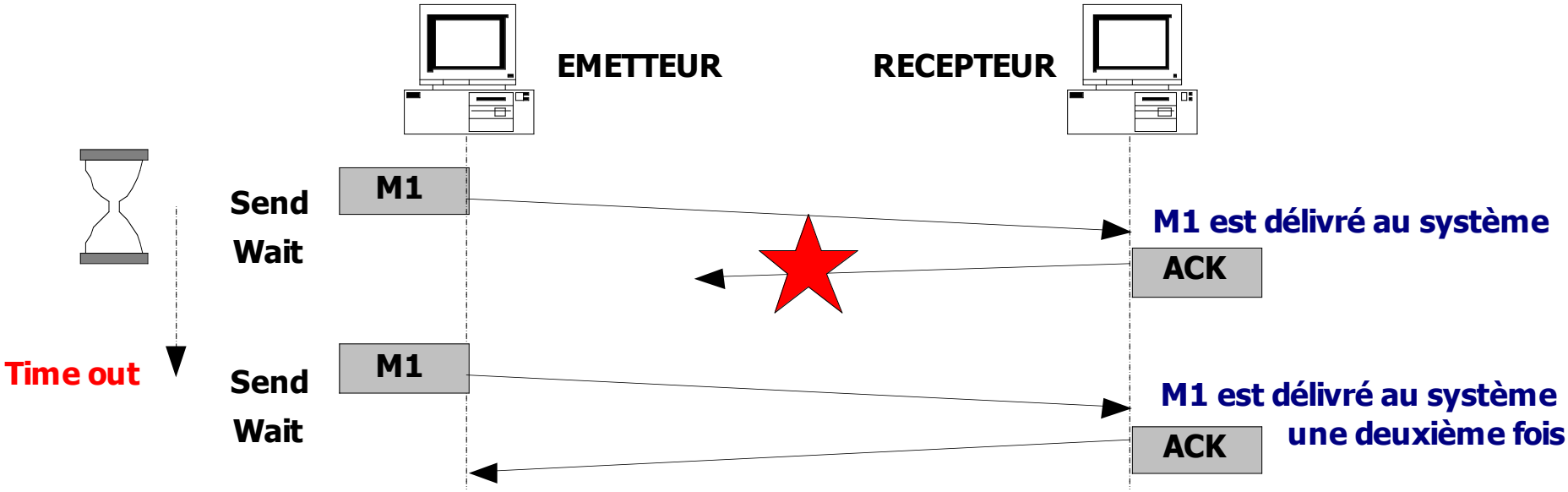
- La reprise sur temporisation



- Time out = compteur
- Problème : que se passe t-il si l'ACK est perdu ?

Les mécanismes de base

- Perte de l'ACK



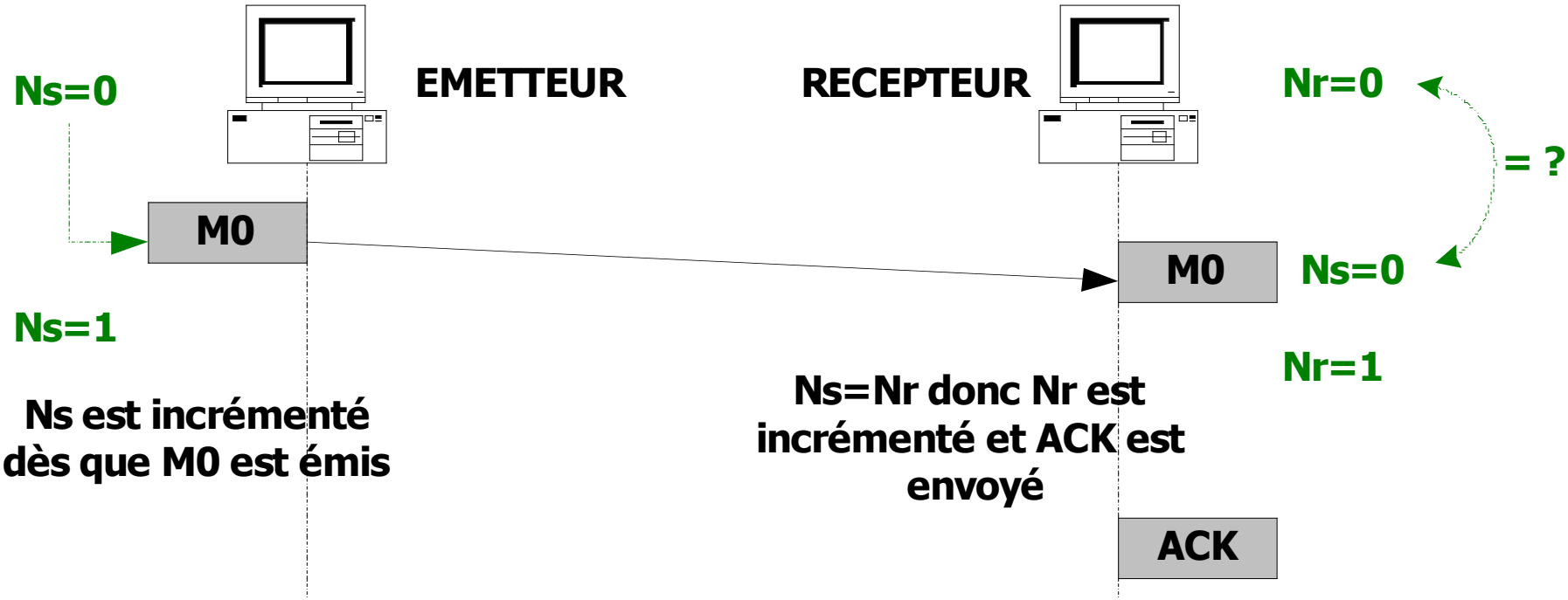
- Remarque : le timer doit être bien réglé (compromis). Si trop grand ? Si trop petit ?
- Solution ? Un ACK d'ACK avant de délivrer M1 ?

Les mécanismes de base

- Numérotation des messages émis
  - On utilise 2 compteurs (**Ns** en émission, **Nr** en réception)
  - Ns et Nr sont initialisés à zéro
  - Ns contient le numéro du prochain message à émettre
  - Nr contient le numéro du prochain bloc à recevoir
  - Ns est transmis de l'émetteur vers le récepteur
  - Un message n'est délivré côté récepteur que si le Ns reçu est égal au Nr local
  - Si Ns < Nr, le message a déjà été reçu, le récepteur le "jette" et l'acquitte de nouveau
  - Attend t-on pour envoyer Mi+1 que Mi soit acquitté ?
  - Ns > Nr est-il possible ?

Les mécanismes de base

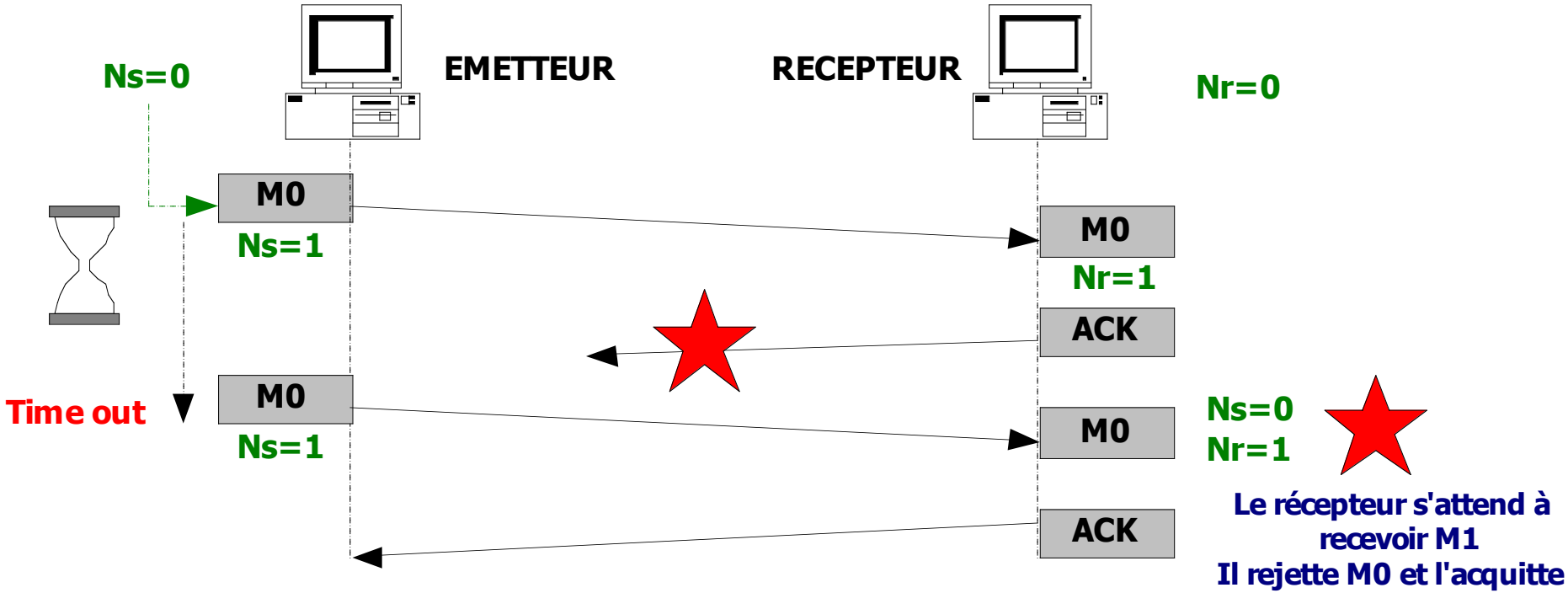
- Numérotation des messages émis



Evite la duplication et permet le contrôle de séquencement des données reçues

Les mécanismes de base

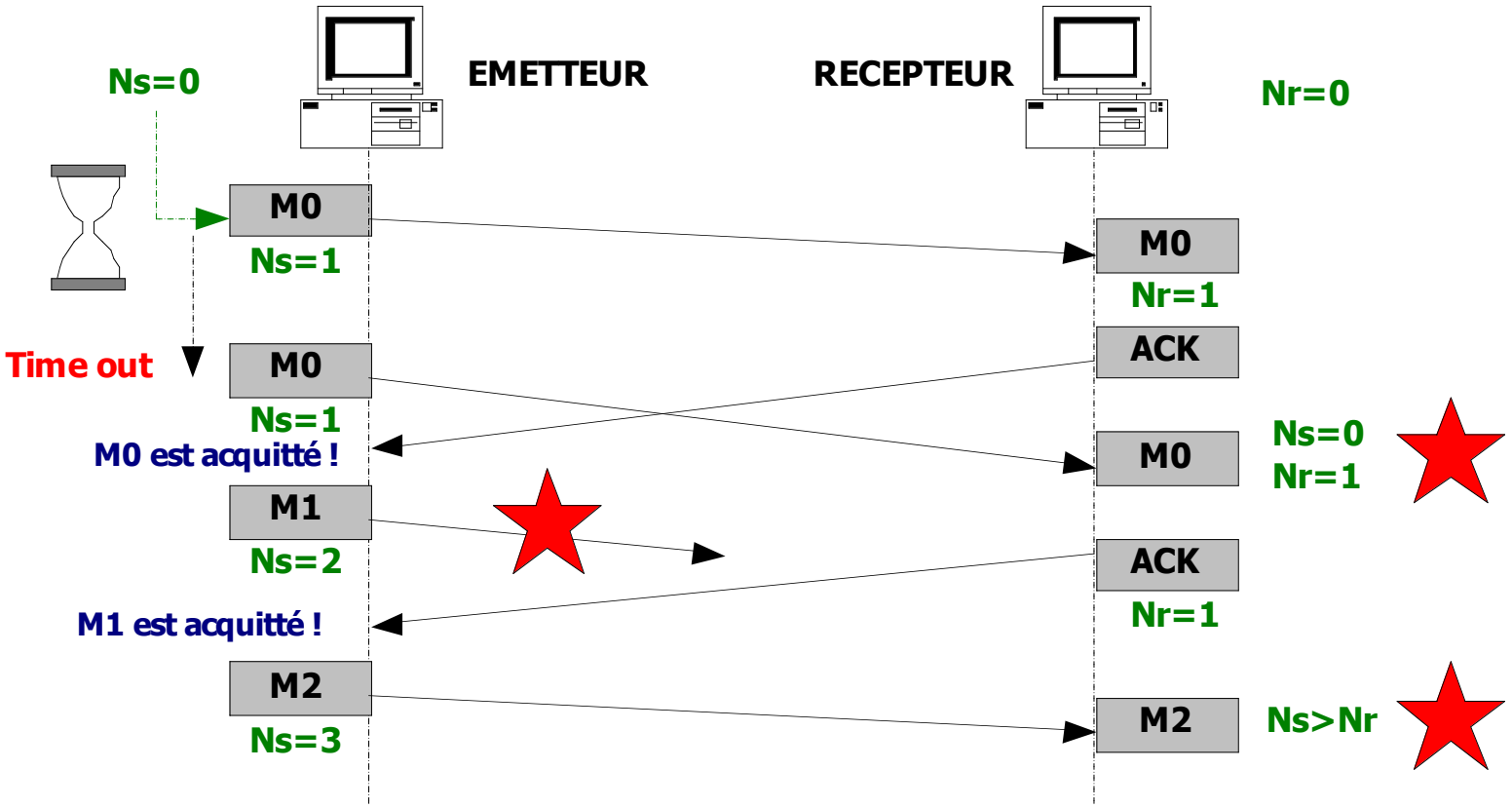
- Perte de l'acquittement



Le deuxième M0 reçu est rejeté

Les mécanismes de base

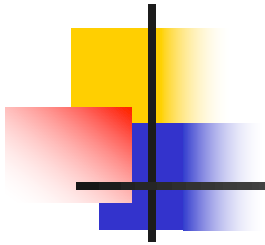
- Délai d'acquittement trop important



M1 n'a jamais été reçu et pourtant il est acquitté
-> il faudrait numéroter aussi les acquittements !

Les mécanismes de base

- Attend t-on pour envoyer Mi+1 que Mi soit acquitté ?
  - Mode *Send&Wait :*
    - on attend (pas adapté pour RTT grand)
    - il est quand même nécessaire de numéroter les acquittements
  - Si on n'attend pas, il faut pouvoir
    - stocker les messages non acquittés sur l'émetteur
    - numéroter les acquittements
- Ns > Nr est-il possible ?
  - Possible dans le cas du slide précédant (même en Send&Wait avec non numérotation des ack)
  - Possible si on envoie Mi+1 alors que Mi n'a pas été reçu (et donc pas acquitté)
  - Les messages n'arrivent alors pas dans le bon ordre sur le récepteur
  - -> soit on refuse les messages tels que Ns>Nr
  - -> soit on stocke les messages désordonnés sur le récepteur

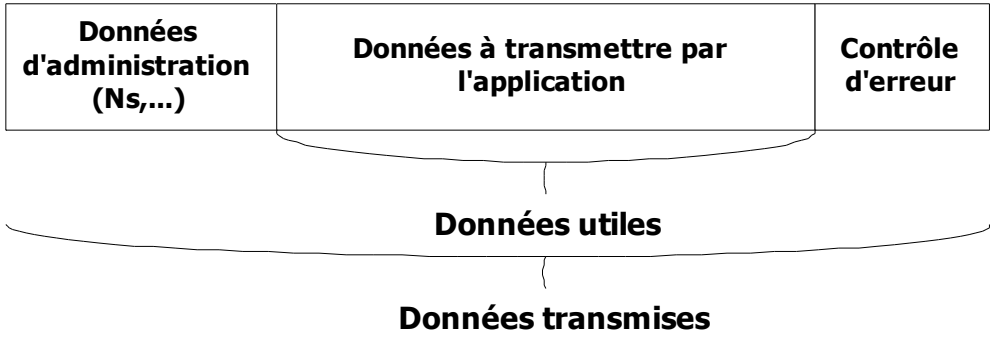


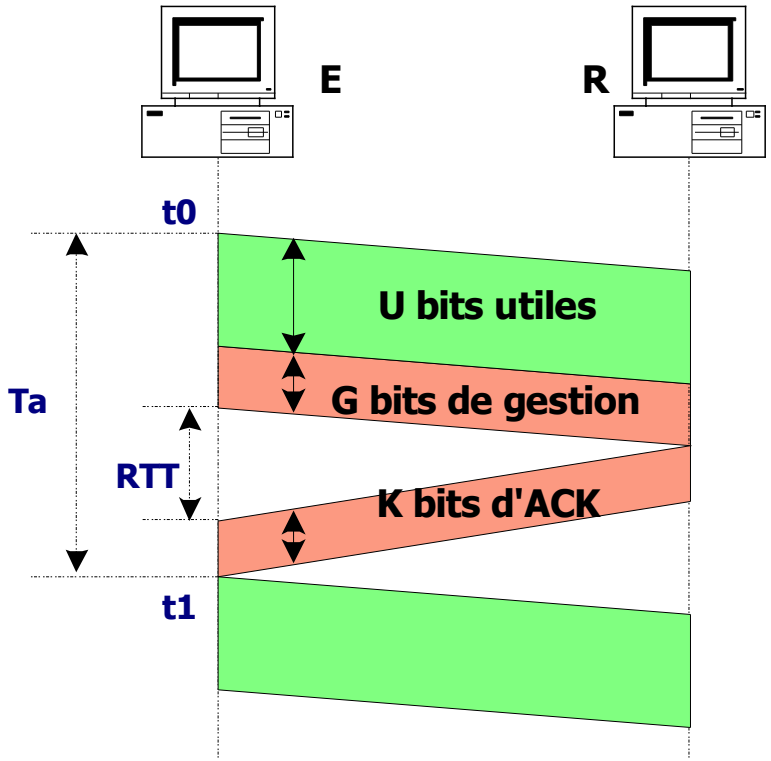
Les mécanismes de base

- Principe du **piggybacking**
  - Quand une trame arrive de A, l'acquittement est envoyé par B dans la trame suivante à destination de A. Quand B n'a pas de message à envoyer à A, il envoie une trame d'acquittement pour éviter le déclenchement du temporisateur.
  - Avantages : meilleure utilisation de la bande passante et moins de trames isolées

Efficacité d'un protocole

- Notion de données utiles





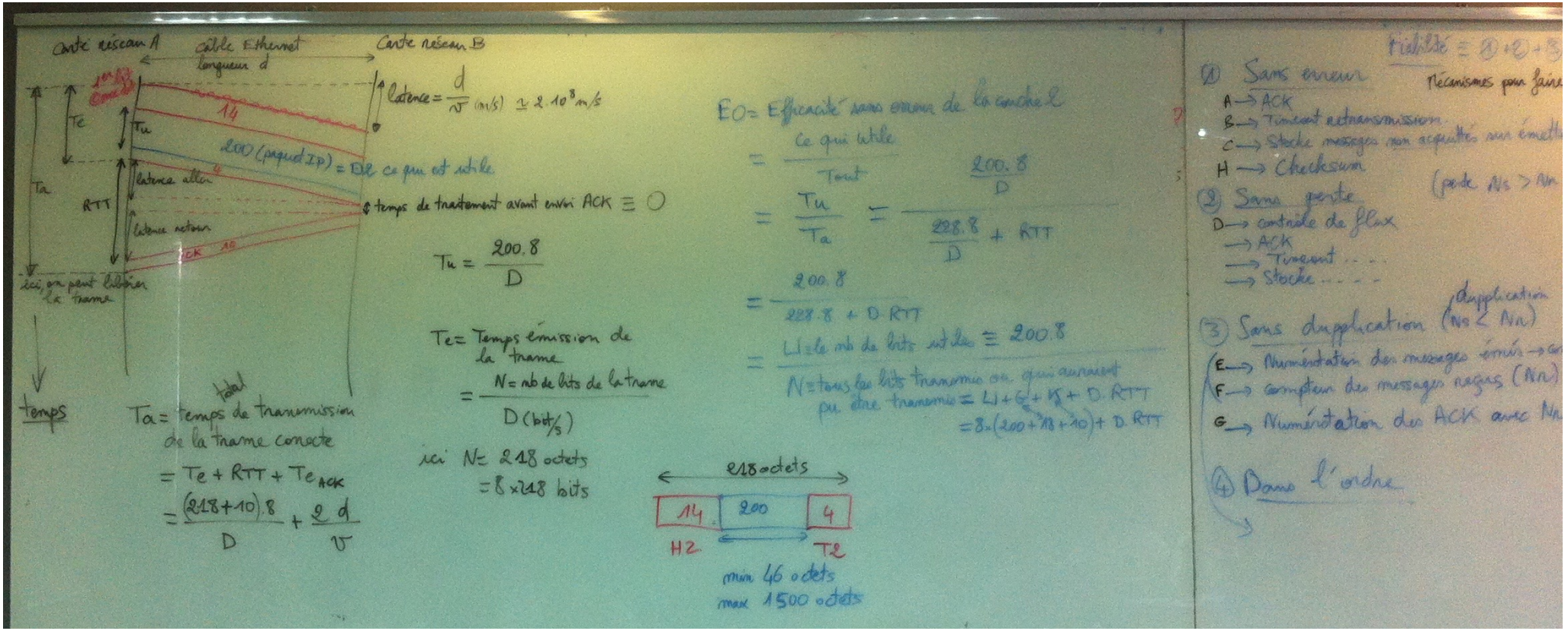
- RTT - *Round Trip Time*
- Ta - temps d'attente entre la transmission du premier bit de Mi et le premier bit de Mi+1
- Efficacité du protocole sans erreur :

  E0 = U/N

  N=nb de bit total transmis (ou qui auraient pu être transmis)

  N=U+G+K+D*RTT (D = débit nominal)

Efficacité d'un protocole



Faire Du = Ehttp.Etcp.Eip.Eeth Dnominal avec la taille des en-têtes et la taille max des paquets-trames

Efficacité du protocole

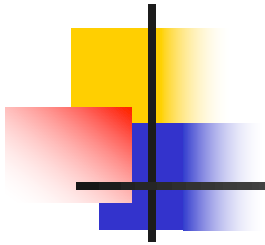
- Cas d'une transmission avec erreur
  - $p = (1\text{-}Teb)^n$ probabilité pour qu'un bloc de n bits soit correctement transmis
  - ici, n = U+G+K
  - L'efficacité du protocole avec erreur est alors :

    $E = E0*p$

    $E = U*(1\text{-}Teb)^{U+G+K}/(U+G+K+D*RTT)$

- Débit réel = débit vu par l'application

  Débit réel = Débit nominal (D) * E

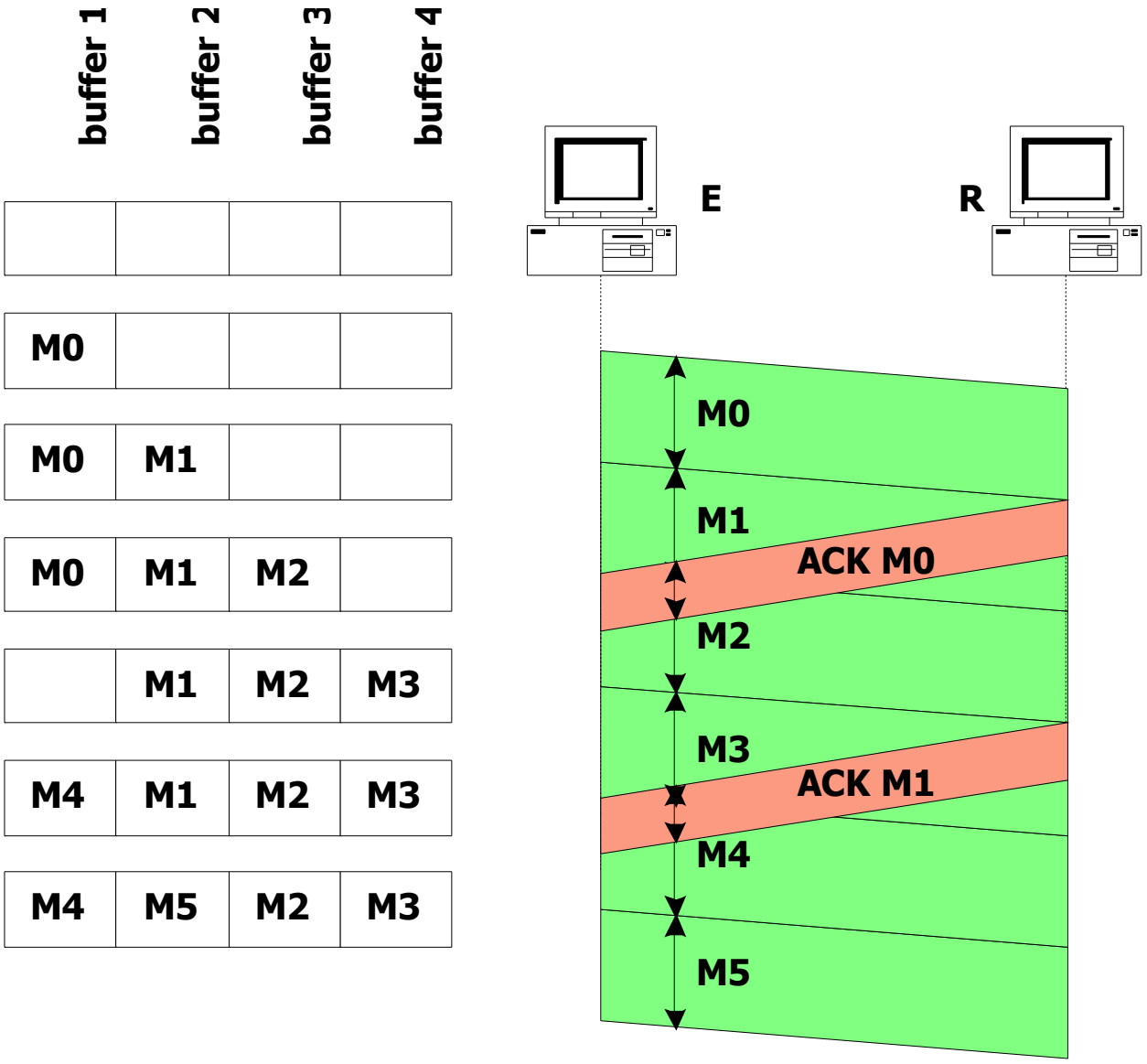


Les protocoles à anticipation

- Dans le mode Send & Wait, les performances sont dégradées du fait de l'attente de l'ACK avant d'envoyer un nouveau message.
- Protocole à anticipation
  - l'émetteur peut faire plusieurs émissions successives sans attendre l'ACK des messages précédents
  - -> il faut numéroter les acquittements
  - -> il faut mémoriser TOUS les messages non acquittés sur l'émetteur dans des "buffers"
  - -> quand un ACK arrive, l'émetteur peut libérer le buffer correspondant au(x) message(s) acquitté(s)
  - -> s'il n'y a plus de buffer libre, l'émetteur doit attendre l'arrivée d'un ACK pour continuer d'émettre

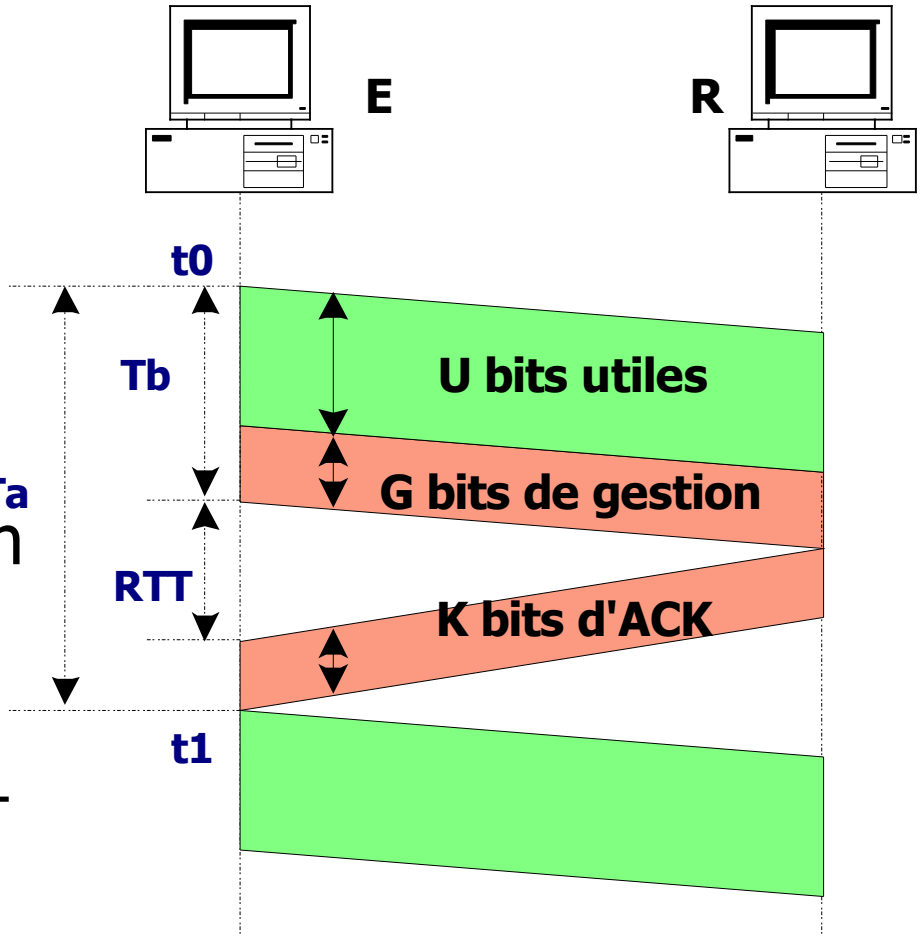
Les protocoles à anticipation

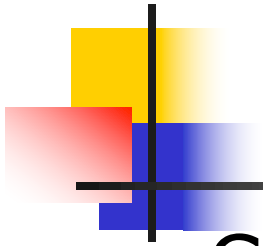
- Principe



Les protocoles à anticipation

- Fenêtre d'anticipation (notée W)
  - crédits d'émission dont dispose l'émetteur
- Taille optimale de la fenêtre
  - quand l'émission se fait en continue (l'émetteur n'attend jamais un ACK)
  - W optimale = E[Ta/Tb]+1
  - dépend de RTT et de la taille de trame maximale



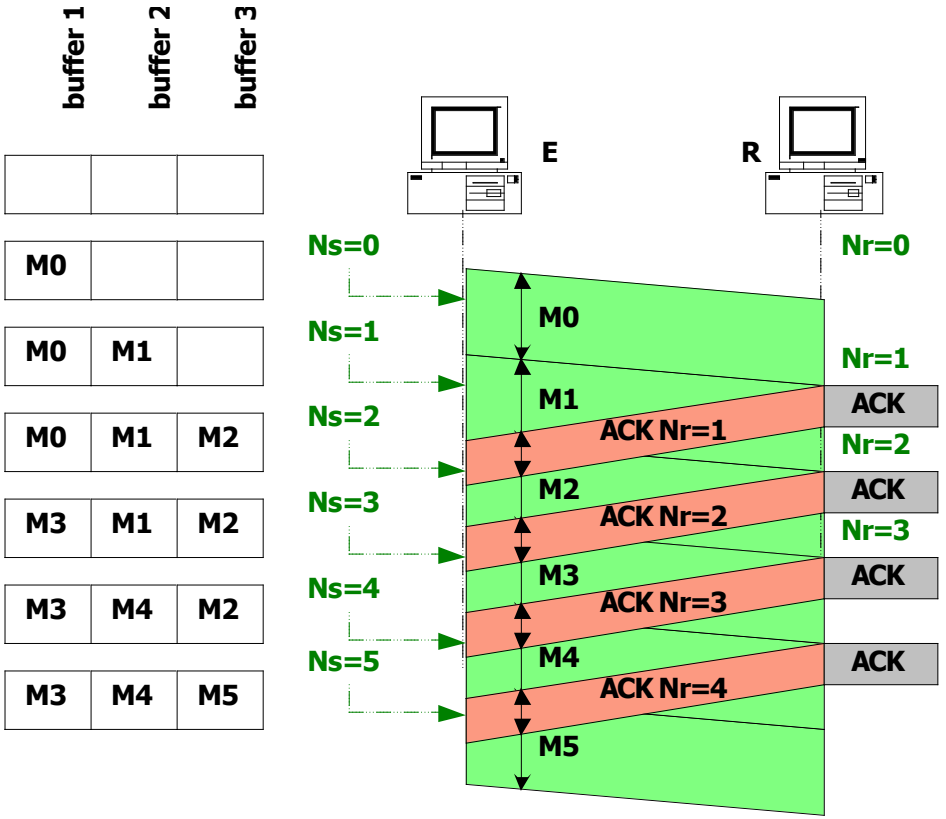


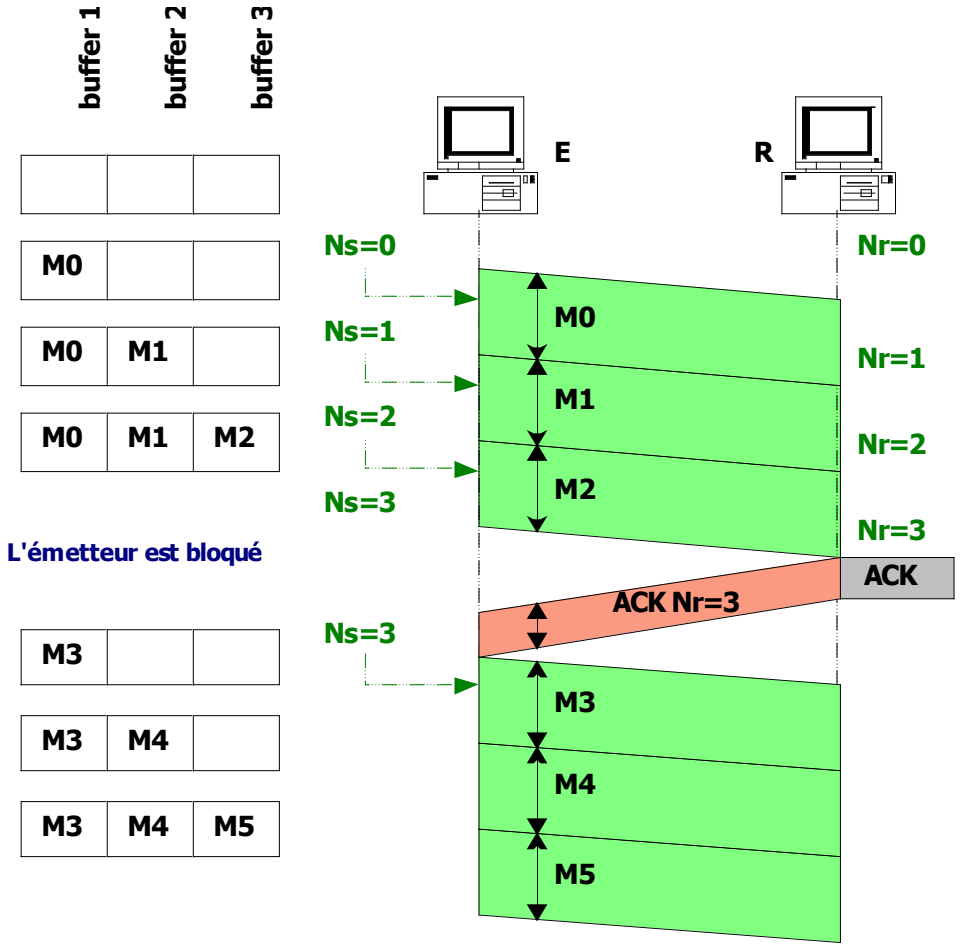
Les protocoles à anticipation

- Gestion glissante de la fenêtre
  - quand un ACK avec Nr=i arrive, l'émetteur libère le buffer qui contient le message Mi-1
  - exemple avec Ns et Nr stockés sur 3 bits chacun et W=3 (taille de la fenêtre)
- Gestion sautante de la fenêtre
  - l'acquittement est différé et concerne plusieurs messages
  - si W=3, M0, M1 et M2 sont acquittés en une seule fois
  - les émissions s'arrêtent quand les crédits d'émission sont épuisés
  - plus efficace car moins d'acquittements sont transmis mais moins efficace si l'acquittement est perdu car l'émetteur est alors bloqué pendant au moins 1 RTT

Les protocoles à anticipation

- Gestion de la fenêtre avec W=3

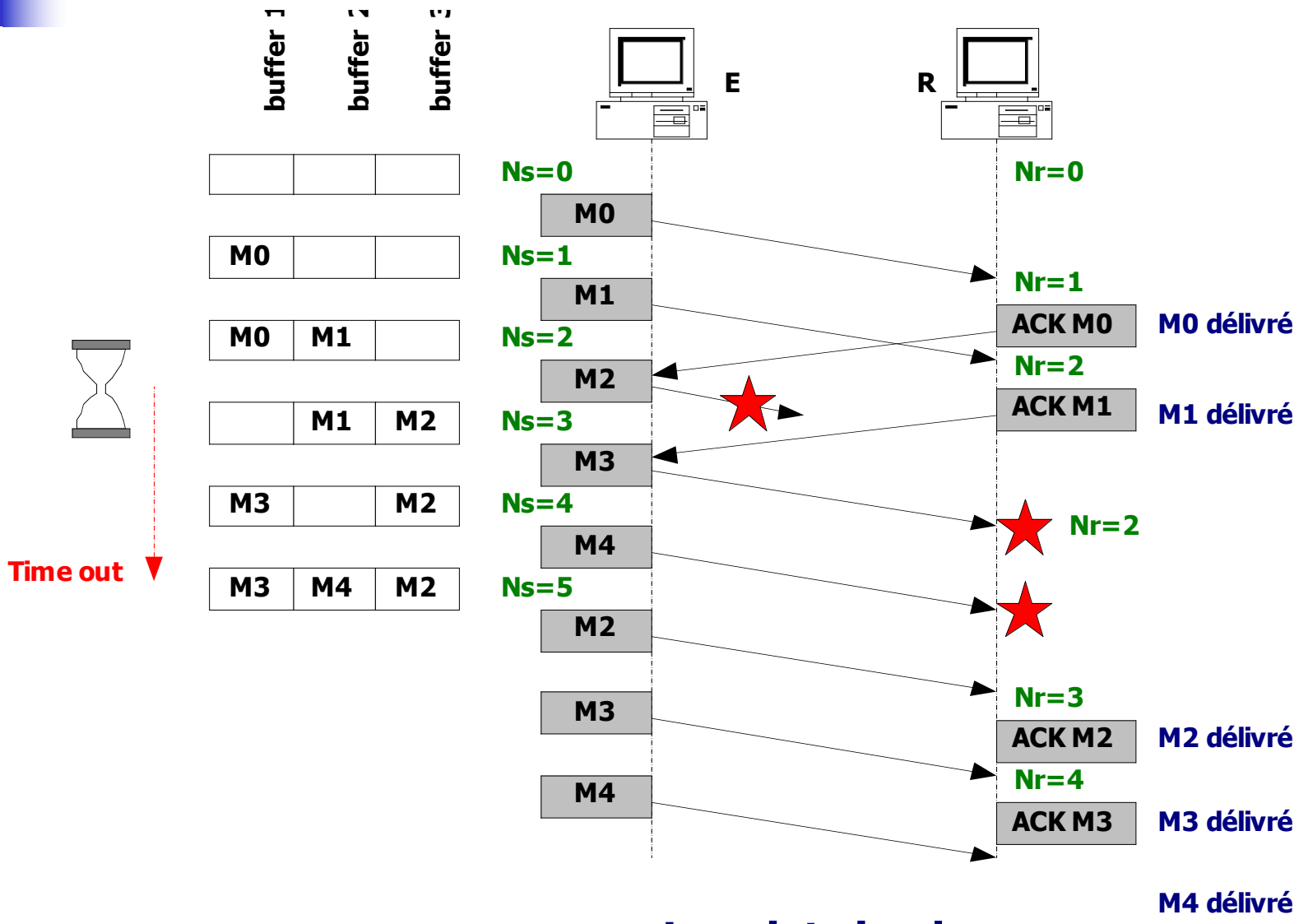




Gestion glissante de la fenêtre

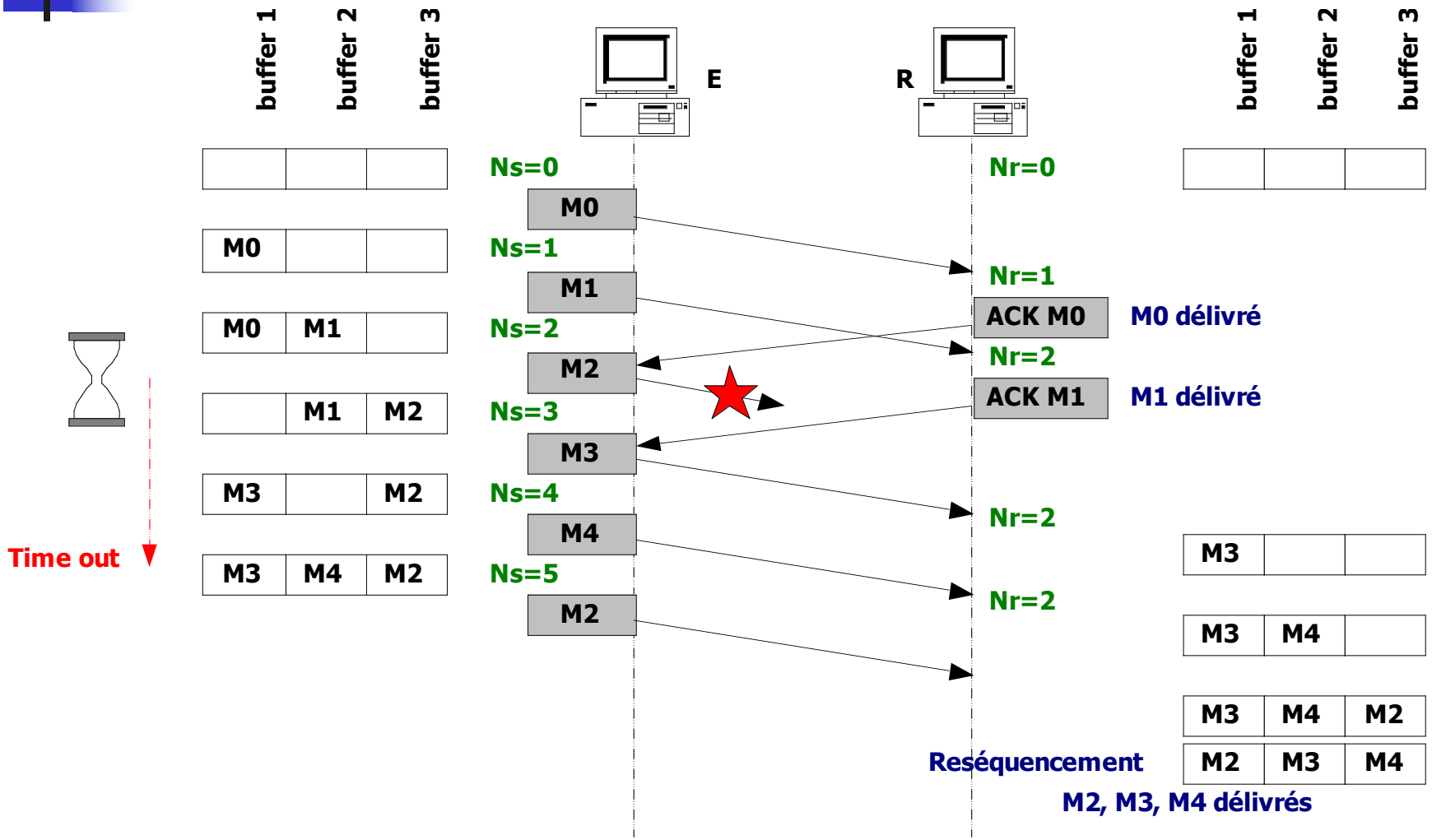
Gestion sautante de la fenêtre

Les politiques de reprise sur erreur

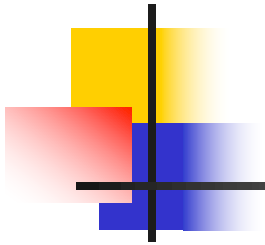


Le rejet simple

Les politiques de reprise sur erreur

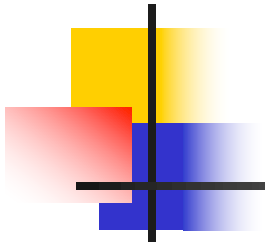


Le rejet sélectif



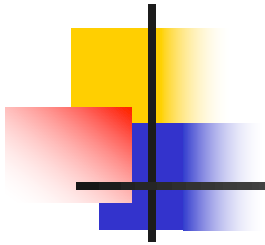
Les politiques de reprise sur erreur

- Rejet simple ou sélectif ?
  - Rejet simple
    - tous les blocs reçus hors séquencement sont rejetés
    - le protocole est dit "**Go Back N**"
    - l'émetteur reprend la transmission à partir du message perdu
    - **mémoire du récepteur minimisée**, Wréception = 1
  - Rejet sélectif
    - le récepteur mémorise les messages hors séquencement
    - l'émetteur ne retransmet que les messages erronés
    - Wréception = nombre de messages déséquencés pouvant être reçus
    - **transmission optimisée - mémoire importante en réception**



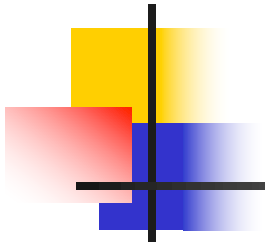
Récapitulatif sur les fenêtres

- Principe de la fenêtre
  - Autorisation pour l'émetteur d'envoyer un certain nombre de trames avant de recevoir un acquittement du récepteur
  - Nombre déterminé par la taille de la fenêtre, correspondant à un ensemble de numéro de séquence
  - Numéros de séquences dans la fenêtre ➡ numéros des trames envoyées et pas encore acquittées
  - Le récepteur maintient une fenêtre qui détermine l'ensemble des trames qu'il peut accepter hors séquence
  - La taille de la fenêtre de l'expéditeur peut être différente de celle du récepteur



Le contrôle de flux

- Le nombre de buffer sur le récepteur limité : l'émetteur ne doit pas émettre plus de données que le récepteur ne peut en accepter sinon les paquets en sus seront perdus
- **Le contrôle de flux est le mécanisme qui consiste à asservir la cadence d'émission de l'émetteur sur les capacités de réception du récepteur**



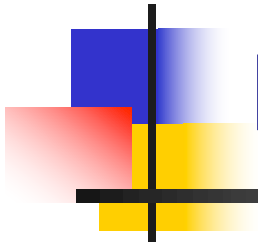
Le contrôle de flux

- On appelle **crédit d'émission** (Ct) le nombre de blocs que l'émetteur est autorisé à transmettre
- Contrôle de flux implicite
  - le nombre de crédits est fixé une fois pour toute ; quand l'émetteur a épuisé ses crédits, il attend l'autorisation du récepteur pour reprendre l'émission
- Contrôle de flux explicite ou dynamique
  - le récepteur informe en permanence l'émetteur sur ses capacités de réception ; le message du récepteur contient le nouveau nombre de crédits disponibles

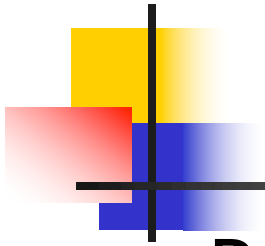
Le contrôle de flux

http://wps.aw.com/aw_kurose_network_2/0,7240,227091-,00.html.

- Il y a plusieurs types de contrôle de flux
  - contrôle de flux à l'interface
    - entre couches adjacentes
    - entre le terminal et le point d'accès au réseau
  - contrôle de flux de bout en bout
    - entre deux terminaux distants
- Contrôle de flux et réseaux haut-débit
  - L'application a du mal à consommer les données reçues
  - Entre le moment où le récepteur constate qu'il est plein et le moment où l'émetteur reçoit le message lui indiquant qu'il n'est plus autorisé à émettre (plus de crédit), beaucoup de messages sont perdus



La signalisation



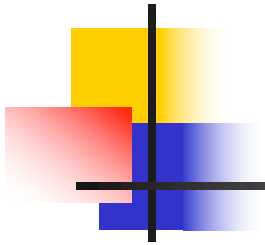
La signalisation

- Pour transférer des données sur une liaison, il est nécessaire de transférer des messages de signalisation pour :
  - établir la liaison, demande de la ligne, composition d'un numéro téléphonique, …
  - contrôler la liaison durant l'échange (messages ACK, …)
  - libérer les ressources en fin de communication
- La signalisation est l'ensemble de ces informations de supervision (ou de contrôle)
- Deux types
  - signalisation dans la bande
  - signalisation hors bande (par canal dédié)

La signalisation dans la bande

- Les informations de signalisation et de données empruntent le même canal de communication
- Un champ spécifique permet de distinguer la nature des informations

Données de contrôle		En-tête protocolaire (fanion, adresses, ...)	**Données de contrôle**	
0	**xxxxxxx**		**1**	**xxxxxxx**
Données applicatives		Champ d'information	**Informations de signalisation**	
CRC		Contrôle d'erreur	**CRC**	



La signalisation hors bande

- Les informations de signalisation empruntent un canal dédié
- Les canaux de signalisation et de données peuvent être physiquement distincts ou emprunter des voies virtuelles (mécanismes de multiplexage)
- Exemples : RNIS, Frame Relay, ATM