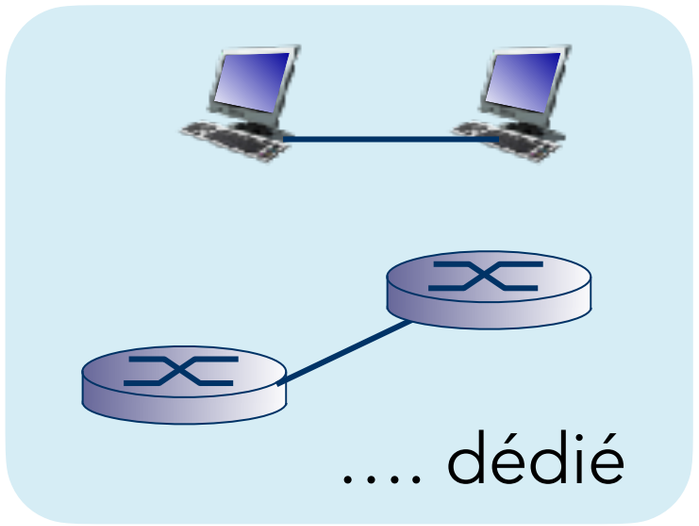
Liaison de données - Plan

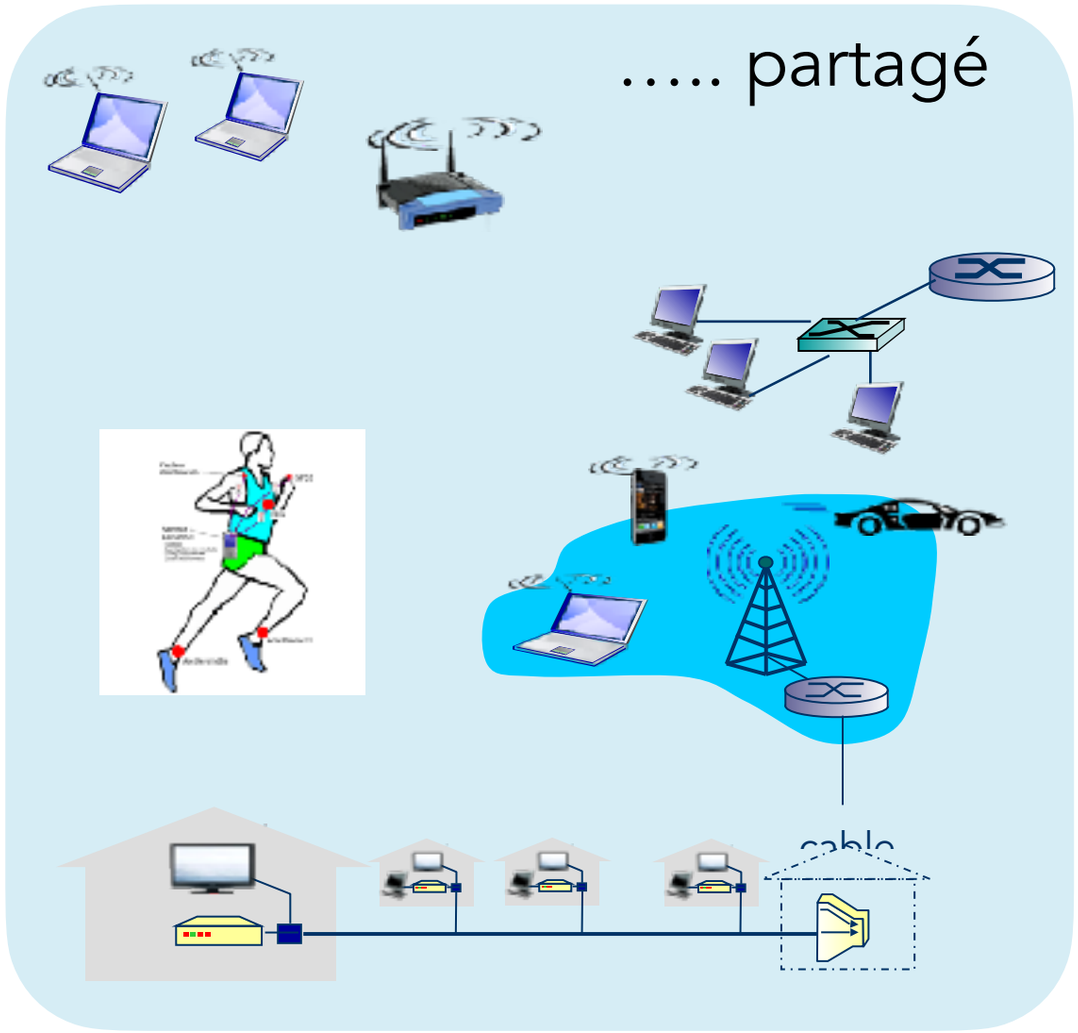
- **Rôle de la liaison de données**
- Fiabiliser la transmission
  1. Déterminer si tous les bits ont été reçus
  2. Contrôle d'erreurs
     a. codes de détection d'erreurs
     b. reprise sur erreurs
- Améliorer l'efficacité de la liaison
  1. Support dédié : efficacité de la liaison
  2. support partagé : partage du canal/réduire les collisions

SORBONNE UNIVERSITÉ

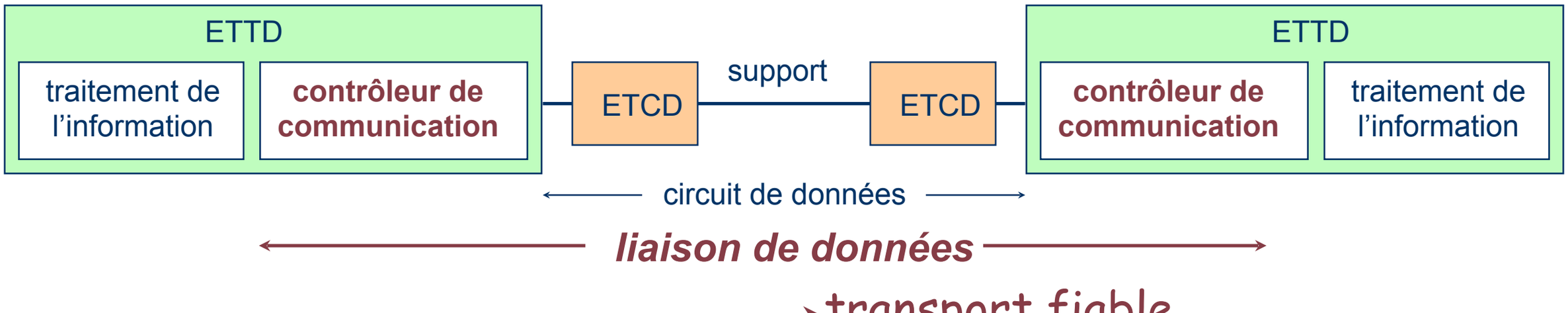
Pourquoi une liaison de données ?

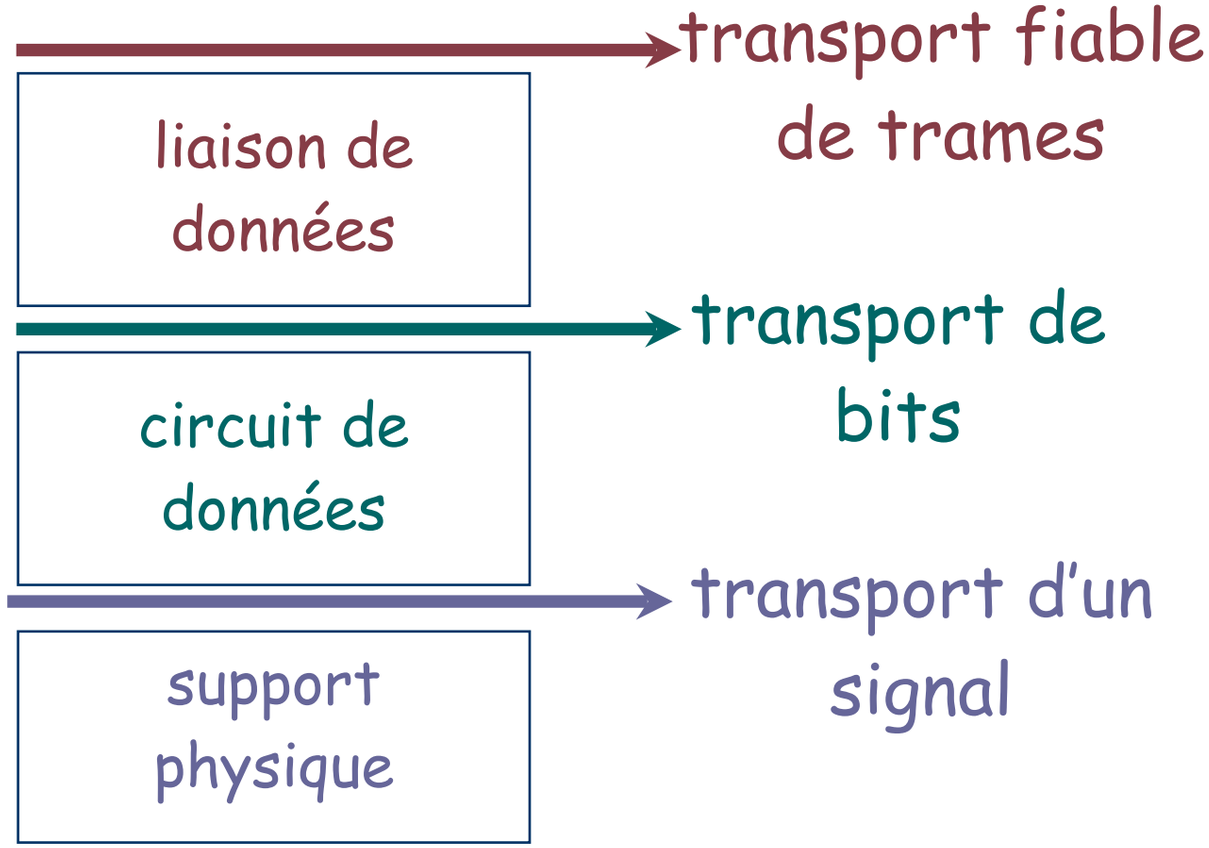
- Le circuit de données permet d'émettre et/ou de recevoir des bits en série avec :
  - un certain débit : $D = R \cdot \log_2 V$
  - un certain délai : $T_t = L / D + d / v_p$
  - un certain taux d'erreurs
- sur un canal de communication





Vue en couches

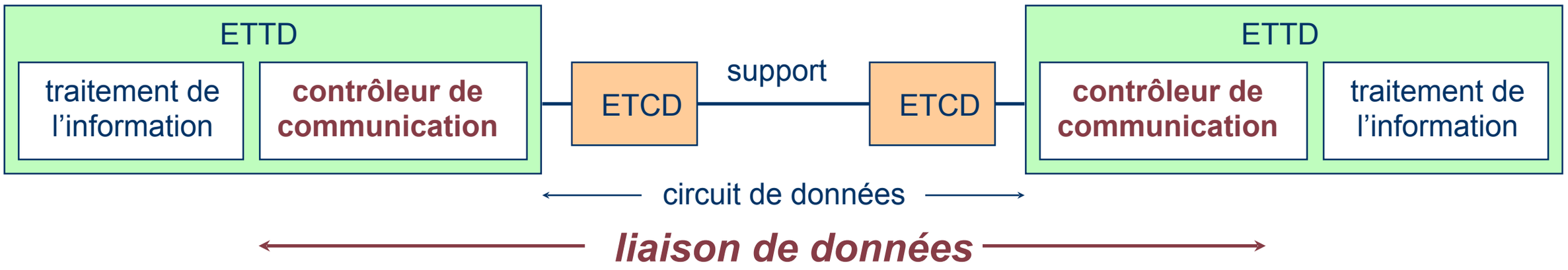




Les problèmes possibles...

- erreurs de transmission
- rupture du circuit de données
- pertes de trames
- débordement du récepteur
- panne d'un des ETTD
- panne d'un des ETCD
- etc.

Rôle d'une liaison de données ?



- Rôle : le transfert (fiable ou non) d'information entre 2 équipements directement reliés au même support de communication
  - Vers le bon destinataire
  - Fiabilité :
    - pas d'erreur
    - pas de perte
    - pas de déséquencement
    - pas de duplication

Protocoles de liaison de données

- une liaison de données : un canal physique capable de transmettre des bits, raccordant 2 (ou *N*) stations et leur permettant d'échanger de l'*information structurée en trames*
- Chaque **protocole** de liaison de données définit l'*ensemble de règles* permettant de gérer la liaison
  - règles de codage
  - règles de structuration
  - règles d'échange

Certains protocoles :

- offrent un service de transport **fiable**
- fonctionnent sur un **support dédié** (2 stations) et/ou **partagé** (N stations)

Leur performances varient

Protocoles de liaison de données

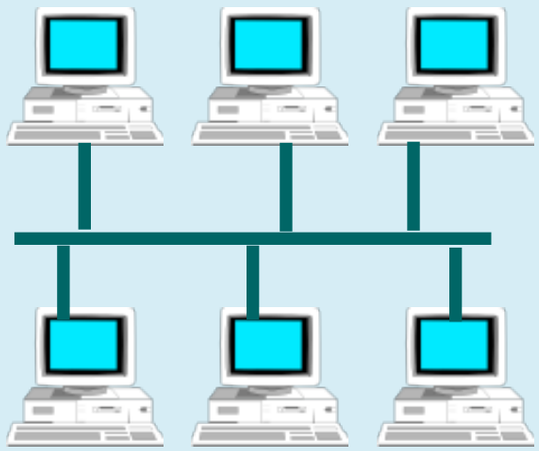
Deux familles de protocoles de communication existent

protocoles en mode connecté

- livrent un **flux de données** (données reçues dans le même ordre que celui dans lequel elles ont été envoyées).
- se déploient principalement sur des supports dédiés
- nécessitent d'établir une session de communication

protocoles en mode non connecté

- envoient les données ***indépendamment*** les unes des autres.
- se déploient principalement sur des supports partagés



Protocoles de liaison de données

Services des **protocoles en mode connecté**

- peuvent **fiabiliser l'échange** : intégrité des données, séquencement (numérotation des données), pas de duplication
- peuvent mettre en place un **contrôle de flux.**
- **améliorer l'efficacité** : utilisation d'une fenêtre d'anticipation

Exemples : PPP, HDLC, SLIP,….

Services des **protocoles en mode non connecté**

- peuvent garantir qu'une donnée reçue est correcte (mais pertes possibles)
- Mettent en oeuvre des méthodes d'accès pour le partage du canal
- L'efficacité sur un médium partagé revient réduire les collisions

Exemples : Ethernet, Wi-Fi, Bluetooth,….

Liaison de données - Plan

- Rôle de la liaison de données
- **Fiabiliser la transmission**
  1. **Déterminer si tous les bits ont été reçus**
  2. Contrôle d'erreurs
     a. codes de détection d'erreurs
     b. reprise sur erreurs
- Améliorer l'efficacité de la liaison
  1. Support dédié : efficacité de la liaison
  2. Support partagé : partager le canal / éviter les collisions

Fiabiliser la transmission

1 - Déterminer si tous les bits envoyés ont bien été reçus

Il faut délimiter les blocs de bits envoyés

Un bloc de bits s'appelle une ***trame***.
Elle est délimitée par une suite particulière de bits appelée ***fanion***.

Chaque protocole définit son propre format de trame (Ethernet, Wi-Fi, PPP, HDLC…)

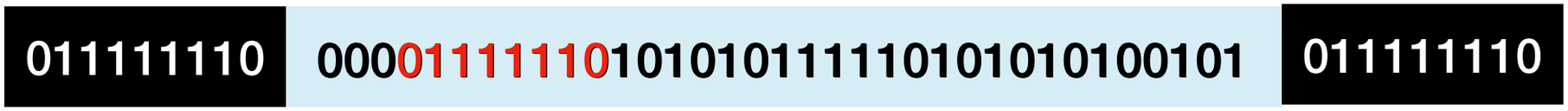
Exemple, pour le protocole HDLC, le fanion est '01111110'

TRAME DE DONNEES

Fiabiliser la transmission

1 - Déterminer si tous les bits envoyés ont bien été reçus

Problème : si les bits de données comportent la séquence



solution

L'émetteur ajoute '0' après cinq '1' consécutifs que le récepteur détruira à la réception

Liaison de données - Plan

- Rôle de la liaison de données
- **Fiabiliser la transmission**
  1. Déterminer si tous les bits ont été reçus
  2. **Contrôle d'erreurs**
     a. **codes de détection d'erreurs**
     b. reprise sur erreurs
- Améliorer l'efficacité de la liaison
  1. Support dédié : efficacité de la liaison
  2. Support partagé : partager le canal / éviter les collisions

Fiabiliser la transmission

Garantir l'intégrité

données reçues = données émises.

Protection contre les erreurs

fiabiliser le support de transmission

→ couche PHY

mécanismes logiques

données erronées

détection → détruites → retransmises

correction → corrigées

Contrôle d'erreurs

Codes de détection / correction d'erreurs

L'émetteur E veut transmettre une suite de bits.

Exemple : E peut envoyer 2 bits 00 01 10 11

Si E envoie 01 et que le récepteur R reçoit 00 ,

R peut-il détecter une inversion de bits ?

=> Il faut rajouter de l'information : **bits de redondance**

Exemple de codage: 0000 0000 0000 1111 1111 0000 1111 **1111**

E envoie 0000 1111 *R reçoit* 0100 1111

L'erreur est-elle détectée ?

Ce code vous parait-il performant ?

Contrôle d'erreurs

- Détection et correction d'erreurs
  - Utilisation d'information de *redondance* pour détecter et/ou corriger les erreurs de transmission. Deux types de codes :
    - les **codes correcteurs** d'erreurs
    - les **codes détecteurs** d'erreurs

Mot de code : unité de n bits contenant des données et un entête

m : bits de données r: bits de redondance **n = m + r**

- 2^m **mots de code légaux**
- 2^n **mots de code possibles et** $2^m < 2^n$ (car m<n)

Soient 2 mots de code différent. Si 2 mots de code sont distants de *d* il faudra *d* bits en erreur pour passer de l'un à l'autre.

distance de Hamming : nombre de bits minimum pour lesquels 2 mots de code différent.

Contrôle d'erreurs - Détection d'erreurs

- Le nombre de bits de redondance nécessaire pour détecter ou corriger les erreurs dépend de la distance de Hamming du code :

> Pour détecter **x** erreurs on a besoin d'une distance de **x+1**
>
> Pour corriger **x** erreurs on a besoin d'une distance **2x+1**

Combien d'erreurs ce code permet-t-il de détecter ? de corriger ?

0000 0000 **0000 1111** **1111 0000** **1111 1111**

Exemples de **code de détection** d'erreurs **:**

- codes de parité
- codes polynomiaux

Détection d'erreurs - codes de parité

Code de parité

- parité paire (ou 0) : nombre pair de 1 dans le mot;
- parité impaire (ou 1) : nombre impair de 1 dans le mot.

E et R se mettent d'accord sur la parité retenue
chacun recalcule les bits de parité associé à une suite de bits

Code de parité croisée

- Séquence binaire décomposée en mots de n bits placé sur chaque ligne
- **Code de parité** (paire/impaire)
  - par ligne : **VRC**
  - par colonne : **LRC**

Données :
« 10111011 » « 10101100 » « 10111010 »

				LRC					
	1	0	1	1	1	0	1	1	**0**
VRC	1	0	1	0	1	1	0	0	**0**
	1	0	1	1	1	0	1	0	**1**
	1	**0**	**1**	**0**	**1**	**1**	**0**	**1**	**1**

- VRC Vertical Redundancy Checking
- LRC Longitudinal Redundancy Checking

Détection d'erreurs - codes polynomiaux

Pour la transmission de données
on utilise des ***codes polynomiaux***
CRC (Cyclic Redundant Code)

Une chaîne de bits est
représentée par un polynôme

Une séquence binaire « 10111011 »
est transformée en un polynôme

x^7	x^6	x^5	x^4	x^3	x^2	x^1	x^0
1	0	1	1	1	0	1	1
x^7	**0**	**x^5**	**x^4**	**x^3**	**0**	**x^1**	**1**

$$M(x) = x^7 + x^5 + x^4 + x^3 + x + 1$$

E et R se mettent d'accord sur un ***polynôme générateur G(x)***
(dont les bits de poids faible et fort doivent être égaux à 1) .

Détection d'erreurs

Cyclic Redundancy Control

E et R utilisent le même *polynôme générateur G(x)* de degré *r*

en émission

- **M(x)** la forme polynomiale de la trame à émettre, de degré k-1
- division mod2 : $M(x).x^r = G(x).Q(x) + R(x)$
- **T(x)** = $M(x).x^r + R(x)$

T(x) = | M(x) | R(x) |

T(x) divisible par G(x)

k premiers bits : bits d'information

r derniers bits : bits de contrôle

en réception

- E reçoit une trame **T'(x)**
- R'(x) div mod2 de T'(x) par G(x)
  - **si** R'(x) ≠ 0 alors erreur(s) de transmission
  - si R'(x) = 0 pas d'erreurs **(**il y a une très forte probabilité que la trame soit exempte d'erreurs)

Détection d'erreurs

$M(x).x^r = x^{13}+x^{12}+x^{10}+x^8+x^7+x^5+x^4$

$G(x)=x^4+x+1$ *r=4*

Au niveau de l'émetteur

1- $M(x).x^r$ <=> 11010110110000

2- Diviser $M(x).x^r$ par G(x) en utilisant la division modulo 2

3- Ajouter le reste R(x) à $M(x).x^r$

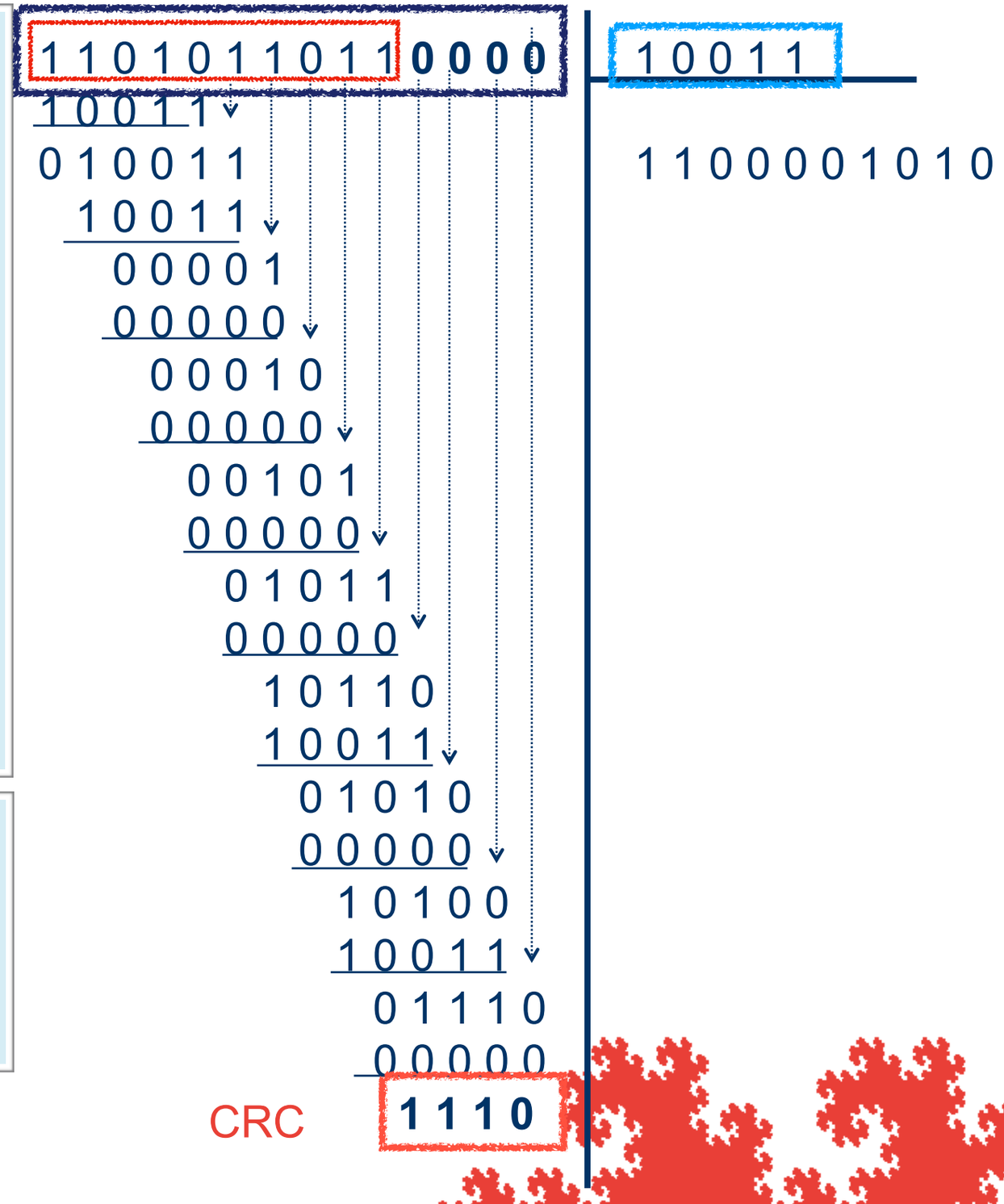
Le résultat représente le checksum de la trame à transmettre T(x)

T(x) = 11010110111110

Au niveau du récepteur

Si T'(x) div mod2 G(x) == 0

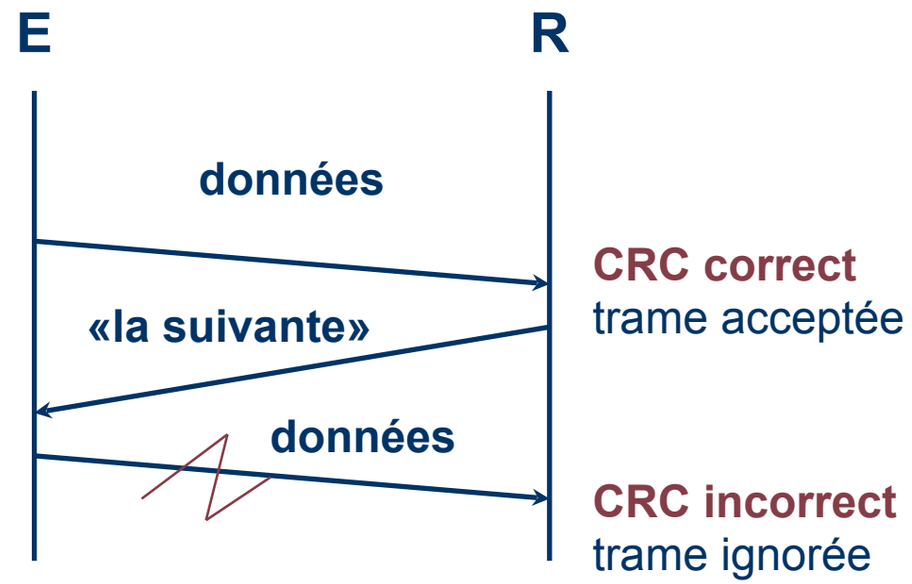
Pas d'erreurs !



SORBONNE UNIVERSITÉ

Détection d'erreurs

l'information de contrôle permet de détecter la présence d'erreurs de transmission dans une trame.



solution

Ajout d'un champ de contrôle de type CRC

Les trames **reçues sont correctes** mais…. comment E sait-il si sa trame a bien été reçue ?

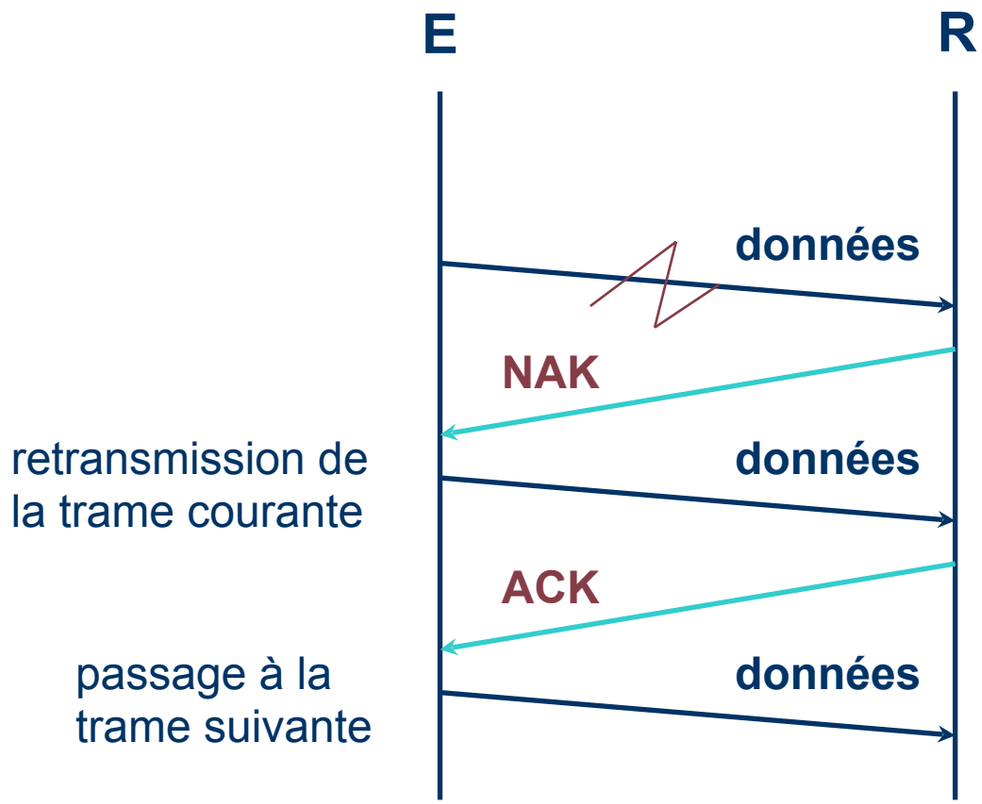
SORBONNE UNIVERSITÉ

Liaison de données - Plan

- Rôle de la liaison de données
- **Fiabiliser la transmission**
  1. Déterminer si tous les bits ont été reçus
  2. **Contrôle d'erreurs**
     a. codes de détection d'erreurs
     b. reprise sur erreurs
- Améliorer l'efficacité de la liaison
  1. Support dédié : efficacité de la liaison
  2. support partagé : empêcher/réduire les collisions

Reprise sur erreurs

Problème : récupérer une trame de données en erreur



solution

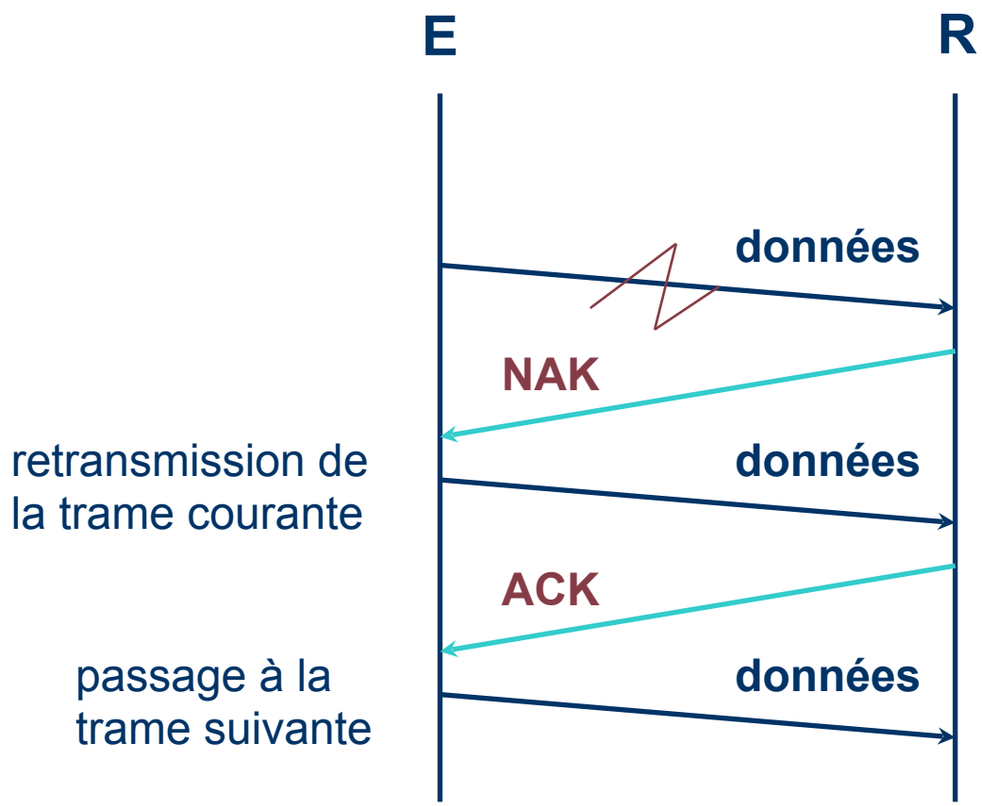
Introduction de trames de contrôle

ACK (« la suivante »)

NAK (« non reçu »)

Reprise sur erreurs

Problème : pouvoir réémettre la dernière trame de données envoyée



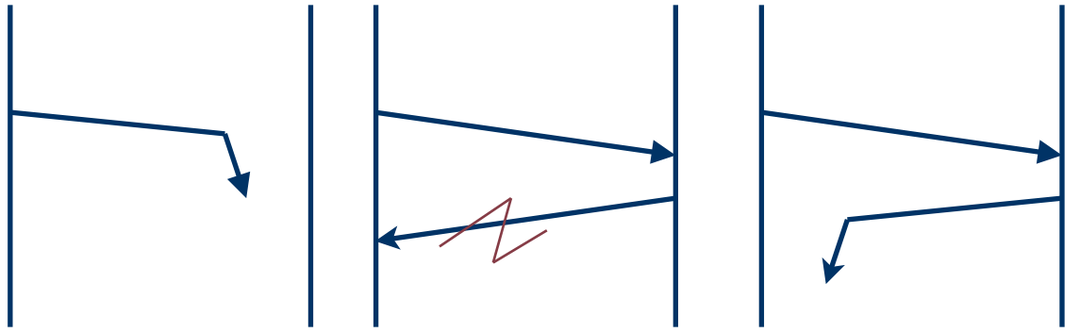
solution

Rétention d'une copie des trames non acquittées :

buffer d'émission

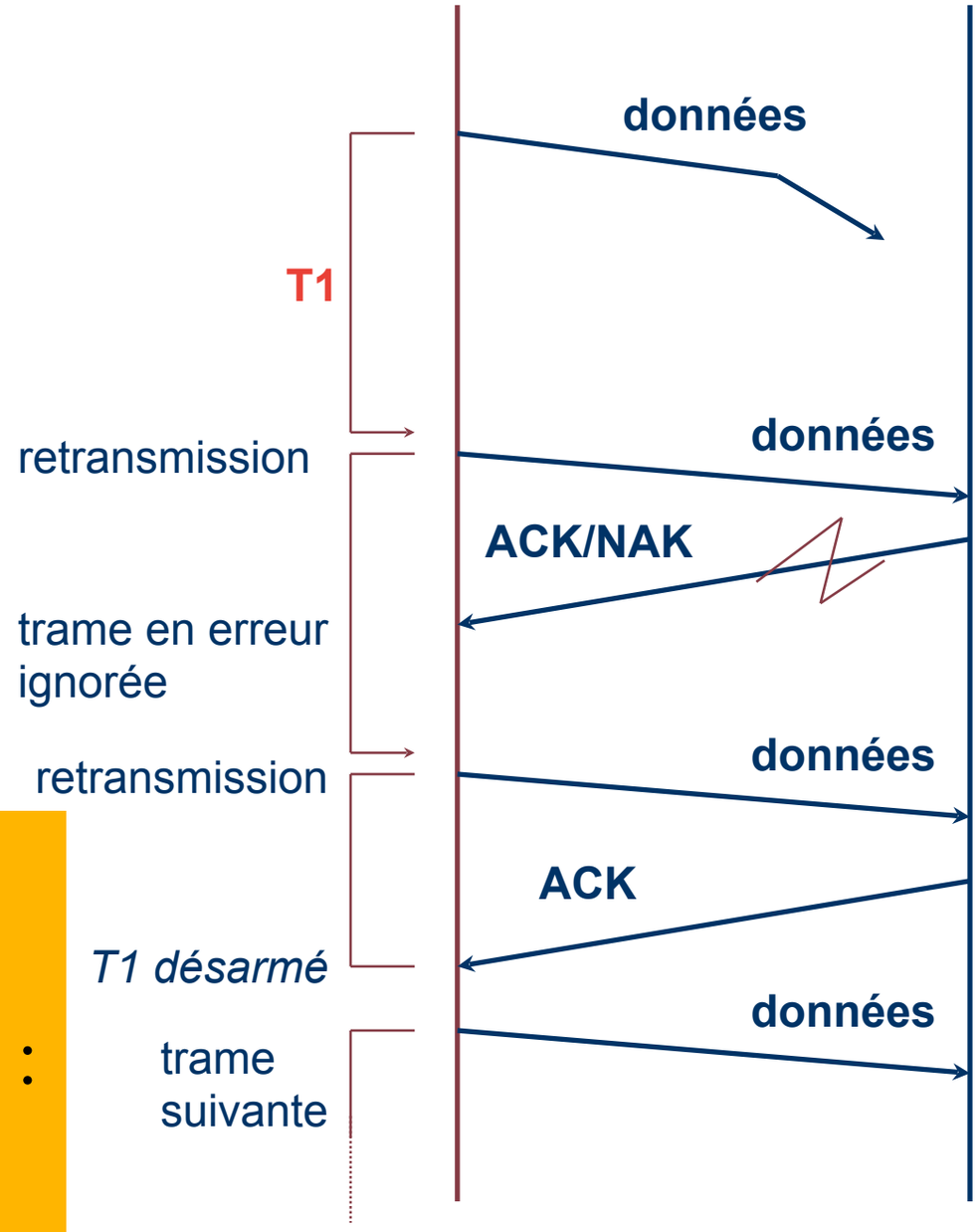
Reprise sur erreurs

Problème : pertes de trames possibles - ***interblocages***



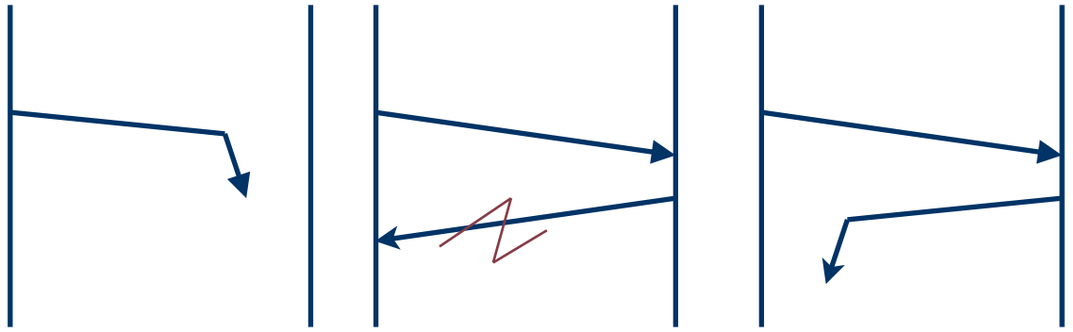
solution

Mécanisme de temporisation pour définir la durée d'attente d'une réponse : temporisateur de retransmission ***T1***



Reprise sur erreurs

Problème : pertes de trames possibles - ***interblocages***



solution

Mécanisme de temporisation pour définir la durée d'attente d'une réponse : temporisateur de retransmission *T1*

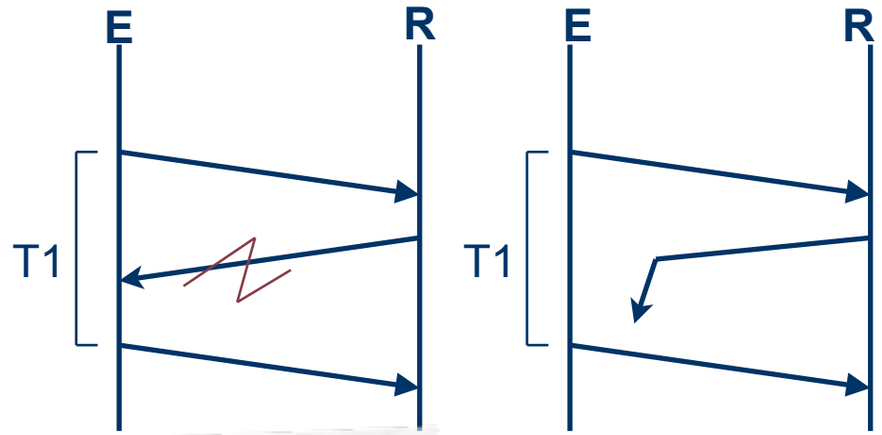
comment dimensionner T1 ?

- Trop petit : retransmissions inutiles
- Trop grand : reprise tardive
- T1 = f(?)

Les trames NAK ne sont plus nécessaires mais elles permettent d'accélérer la reprise

Reprise sur erreurs

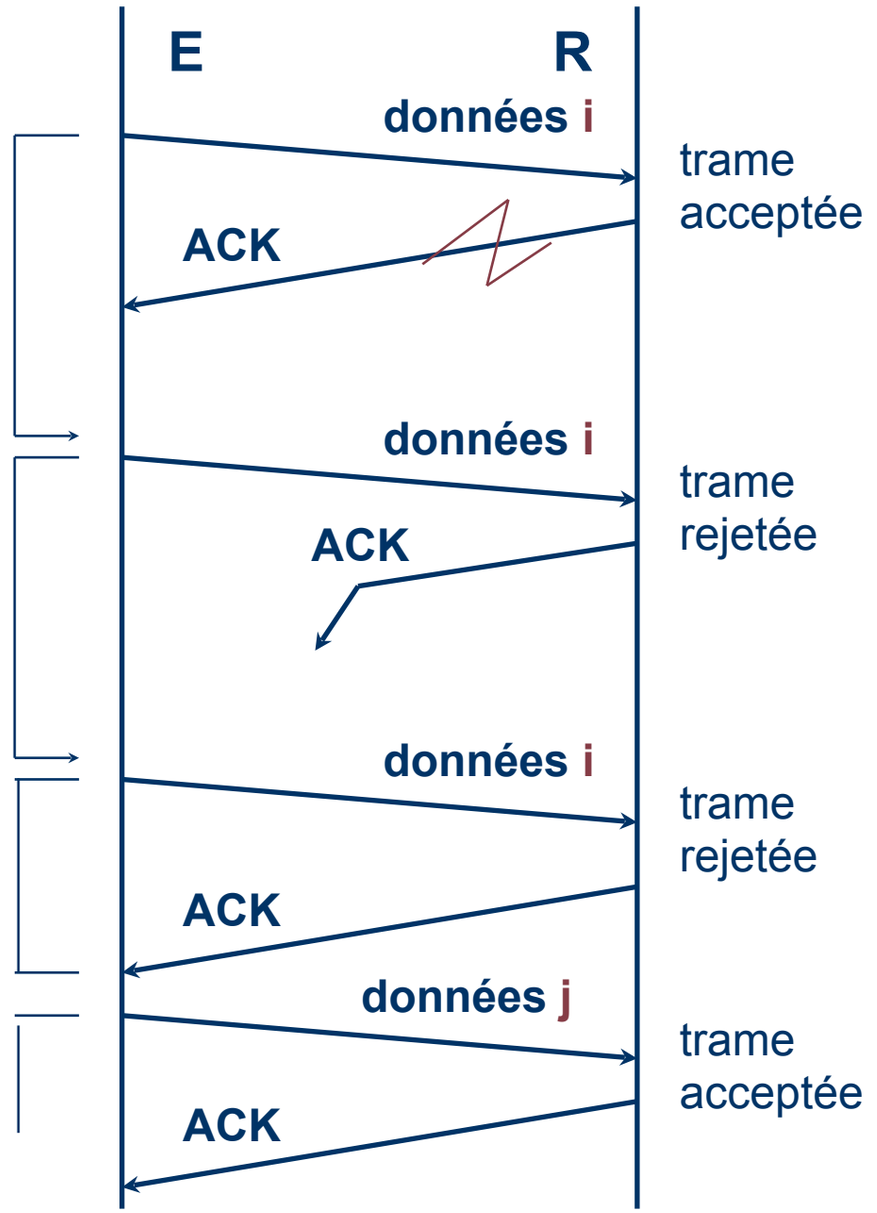
Problème : duplications de données possibles



solution

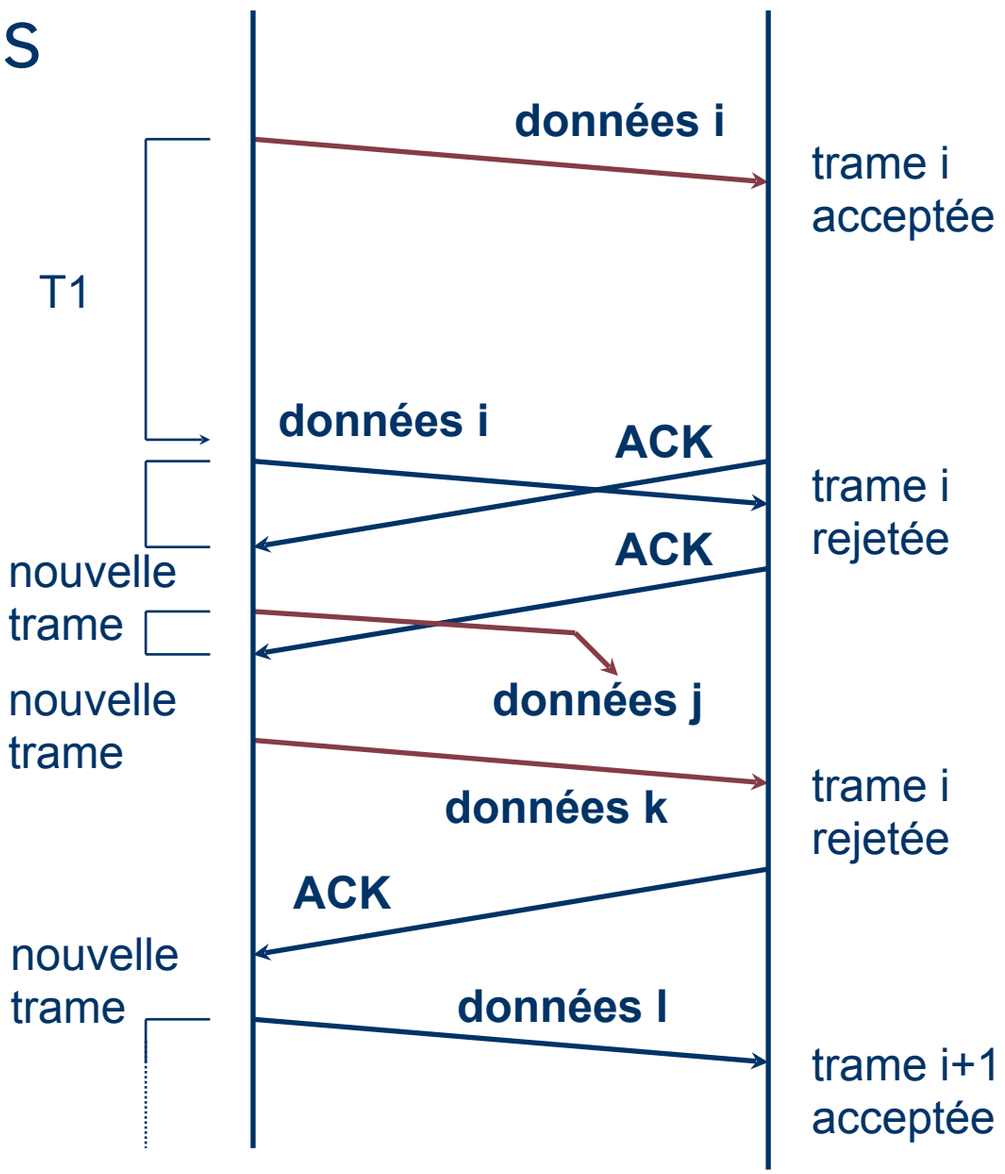
Identifier chaque trame de données :

- identifiant unique (protocole non-connecté)
- numéro de séquence (protocole connecté)



Reprise sur erreurs

Problème : pertes de données non récupérables possibles



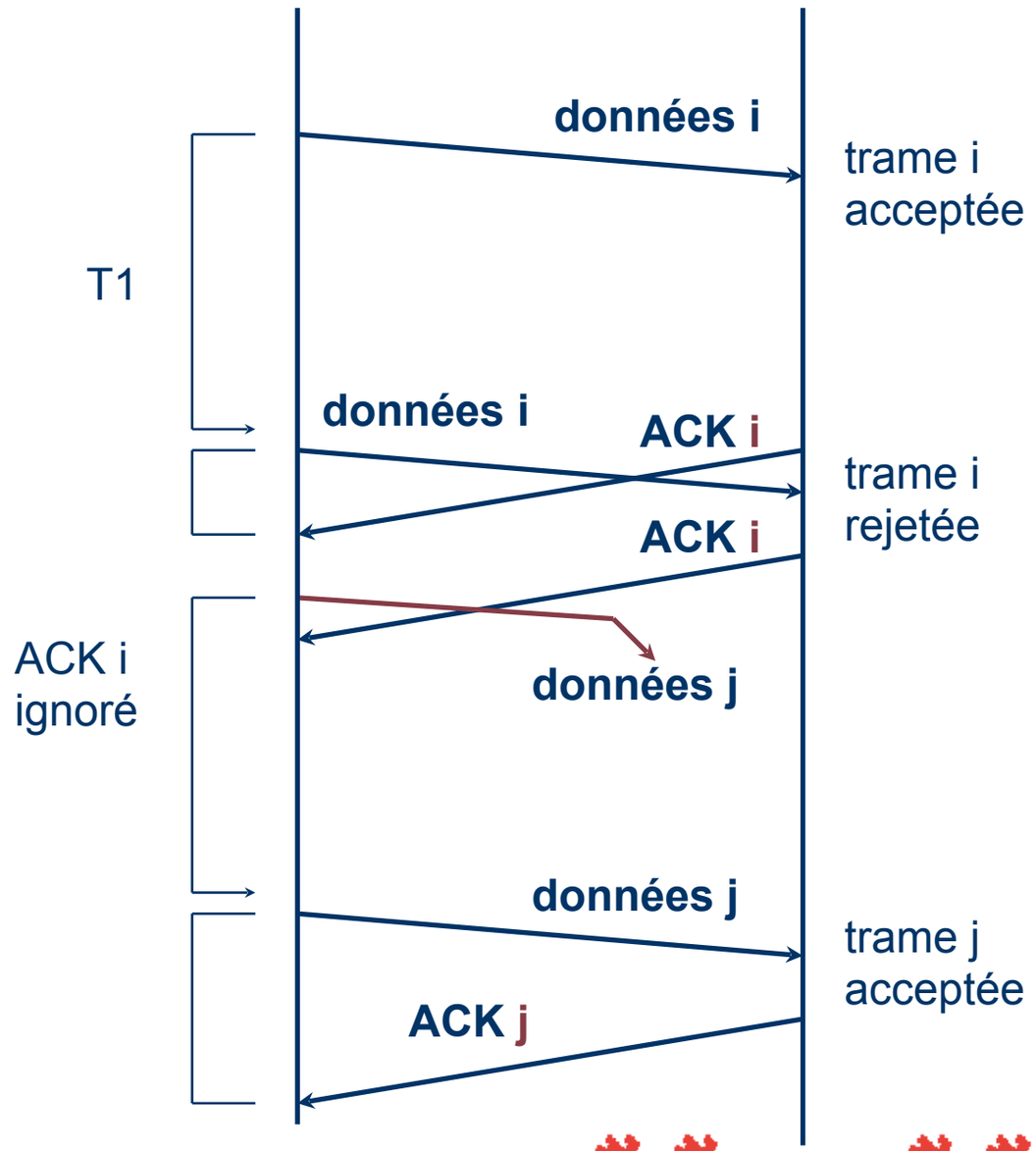
SORBONNE UNIVERSITÉ

Reprise sur erreurs

Problème : pertes de données non récupérables possibles

solution

Numérotation des ACK



SORBONNE UNIVERSITÉ

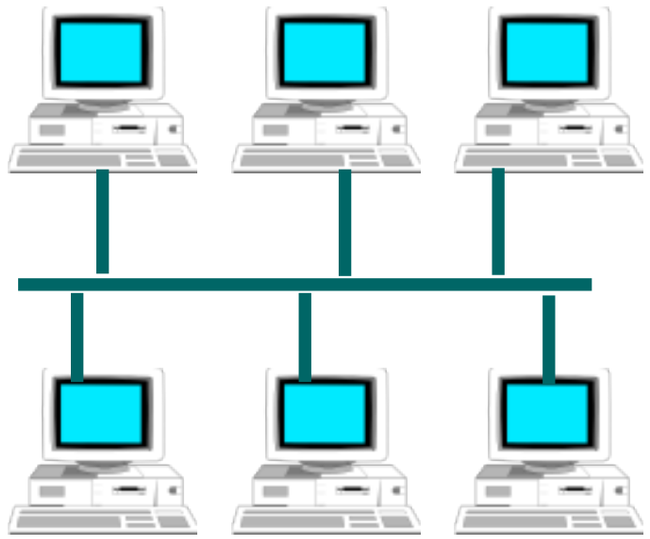
Liaison de données - Plan

- Rôle de la liaison de données
- Fiabiliser la transmission
  1. Déterminer si tous les bits ont été reçus
  2. Contrôle d'erreurs
- **Efficacité des protocoles de la couche liaison de données**
  1. **Support dédié : efficacité de la liaison**
  2. support partagé : empêcher/réduire les collisions

Efficacité des protocoles

L'efficacité des protocoles de la couche liaison de données se mesure en regardant :

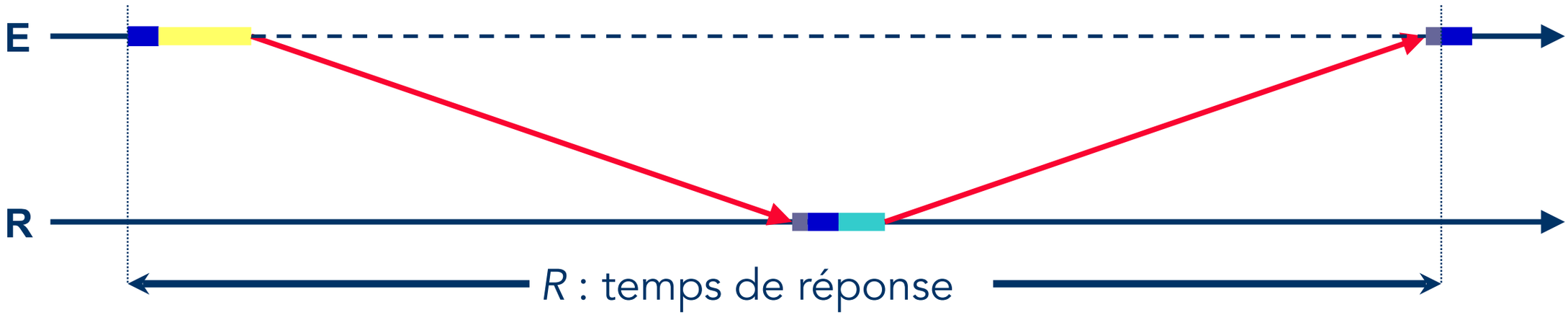
le **taux d'utilisation** de la liaison



La proportion de temps pendant laquelle les transmissions se font avec succès (pas de collisions)

Efficacité d'une liaison dédiée

Problème : mauvaise utilisation du circuit lorsque le temps de propagation est important



- T_{te} : temps de traitement en émission
- T_{td} : temps de transmission des données
- T_p : temps de propagation
- T_{tr} : temps de traitement en réception
- T_{ta} : temps de transmission de l'acquittement

SORBONNE UNIVERSITÉ

Efficacité d'une liaison dédiée

- Soit ρ le taux d'utilisation du canal de transmission

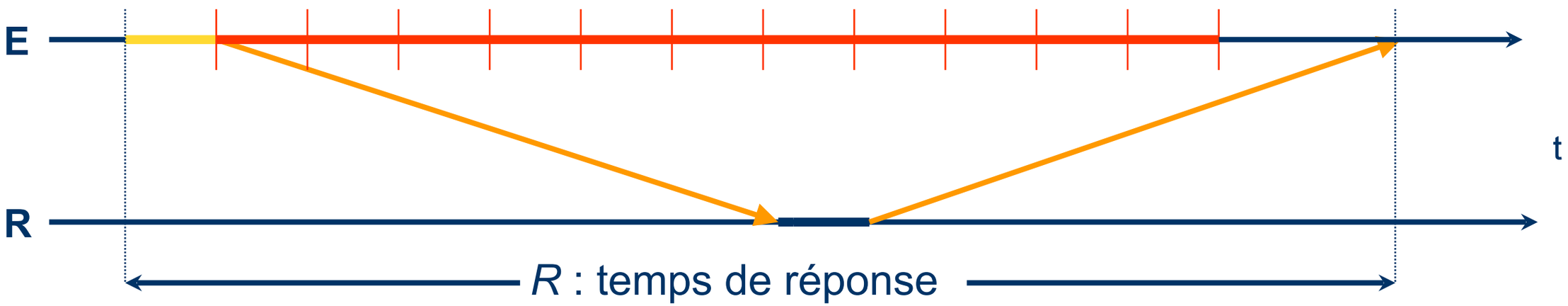
$$\rho = \frac{t_t}{R} \Rightarrow \rho \approx \frac{t_t}{t_t + 2t_p} = \frac{l}{l + 2t_p D}$$

- Cas d'une liaison satellite
  - D = 2 Mbit/s
  - t_p = 270 ms
  - l = 128 octets

ρ = 0,1% !!!

Efficacité d’une liaison dédiée

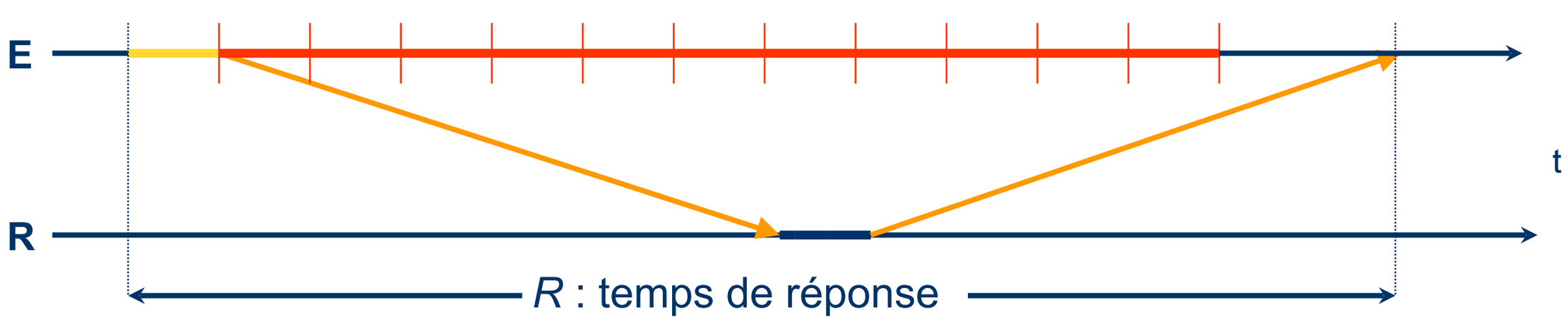
permettre à l’émetteur d’envoyer plusieurs trames consécutives avant de se bloquer en attente d’acquittement



si $N.T_{td} < R$ $\rho' \approx \frac{N.T_{td}}{R}$ sinon $\rho' = 1$

Efficacité d'une liaison dédiée

Idée : permettre à l'émetteur d'envoyer plusieurs trames consécutives avant de se bloquer en attente d'acquittement



Il faut au préalable un **protocole en mode connecté**

- ouvrir/fermer la liaison
- détecter les pannes sur la liaison ou les erreurs
- numéroter les trames en séquence

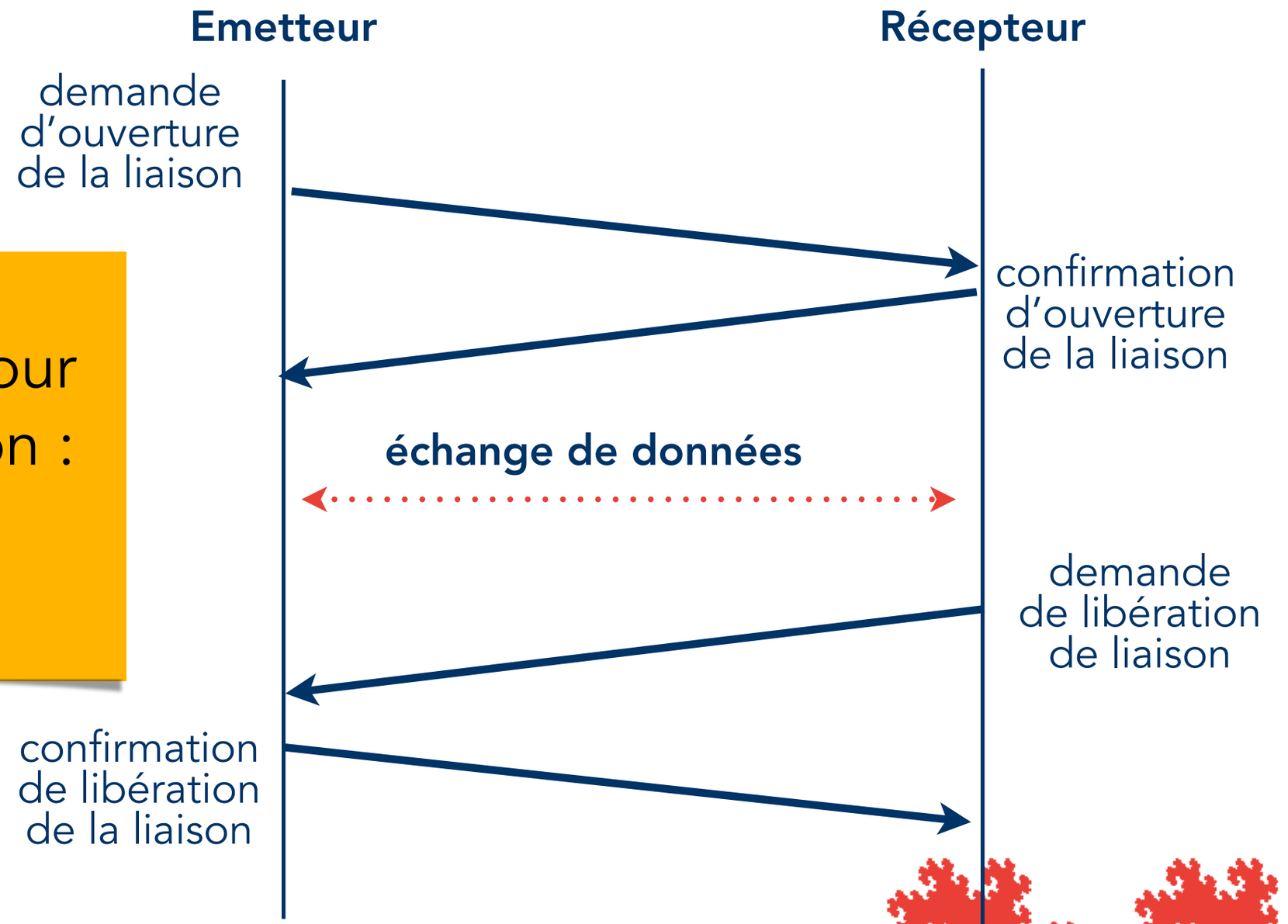
Protocole en mode connecté

Etablir/fermer une liaison

solutions :

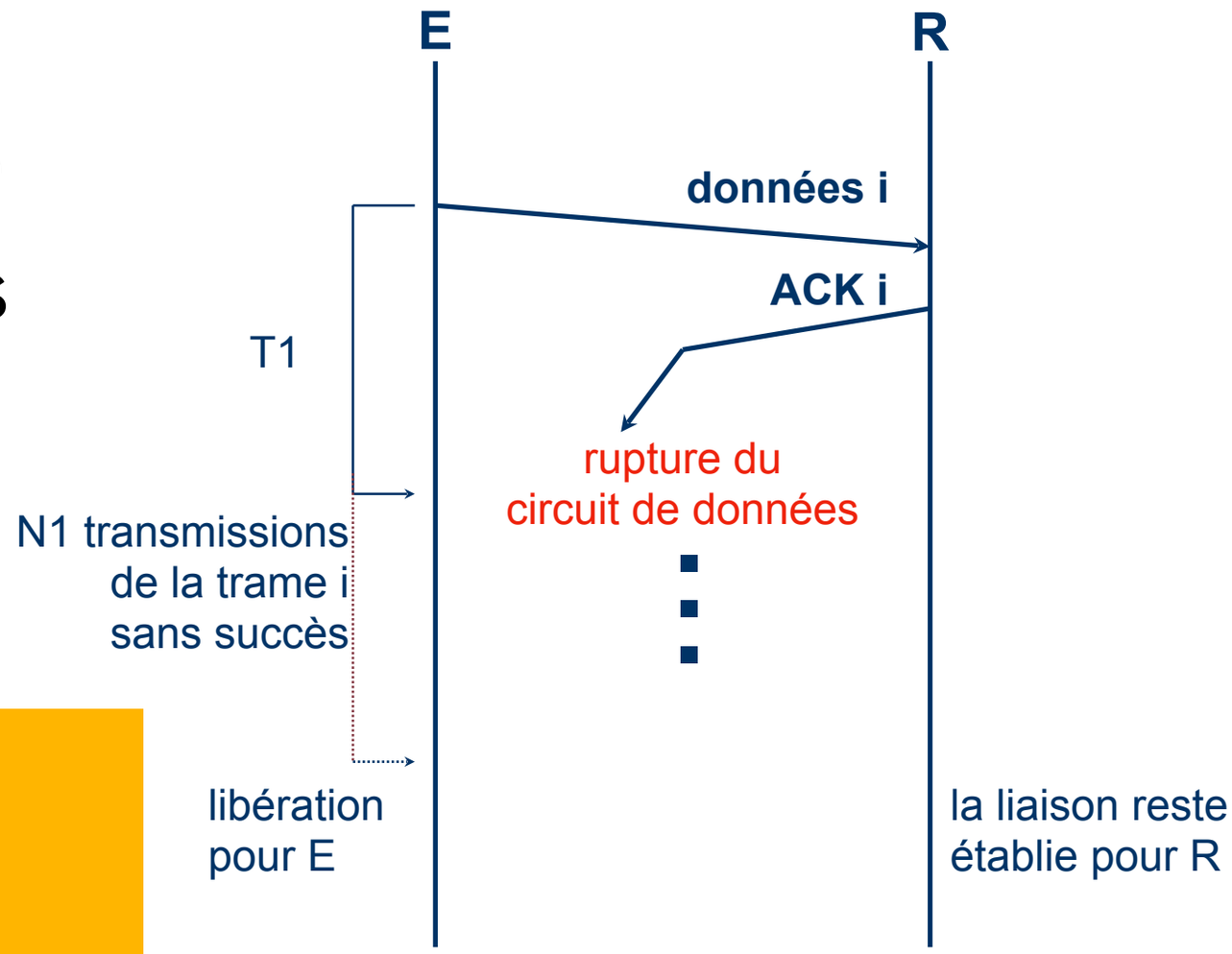
Trame de contrôle pour la gestion de la liaison :

- ouverture
- libération



Protocole en mode connecté

Problème : ressources de communication bloquées inutilement



solutions :

- Temporisateur de détection d'inactivité
- Envoi de trames pour maintenir une activité

Protocole en mode connecté

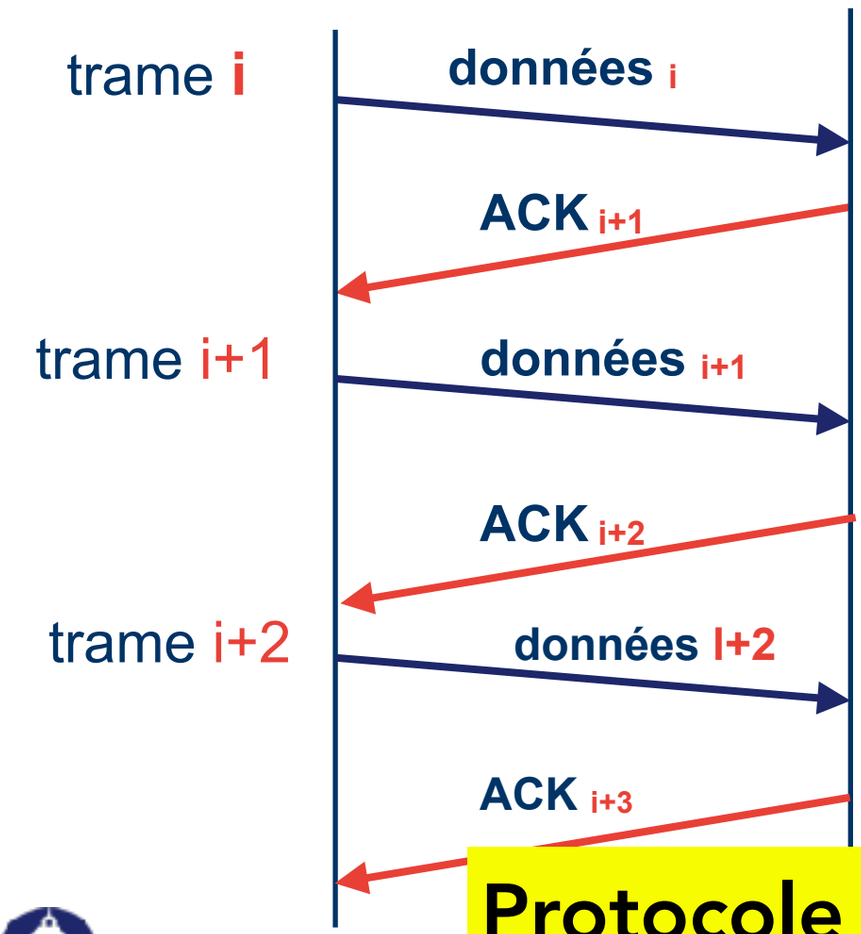
Problème : en cas d'incident grave du circuit de données, retransmissions d'une même trame à l'infini

solutions :

Compteur du nombre maximum *N1* d'émissions d'une trame

Protocole en mode connecté

Avec les protocoles en mode connecté les trames de données sont **numérotées** *en séquence*



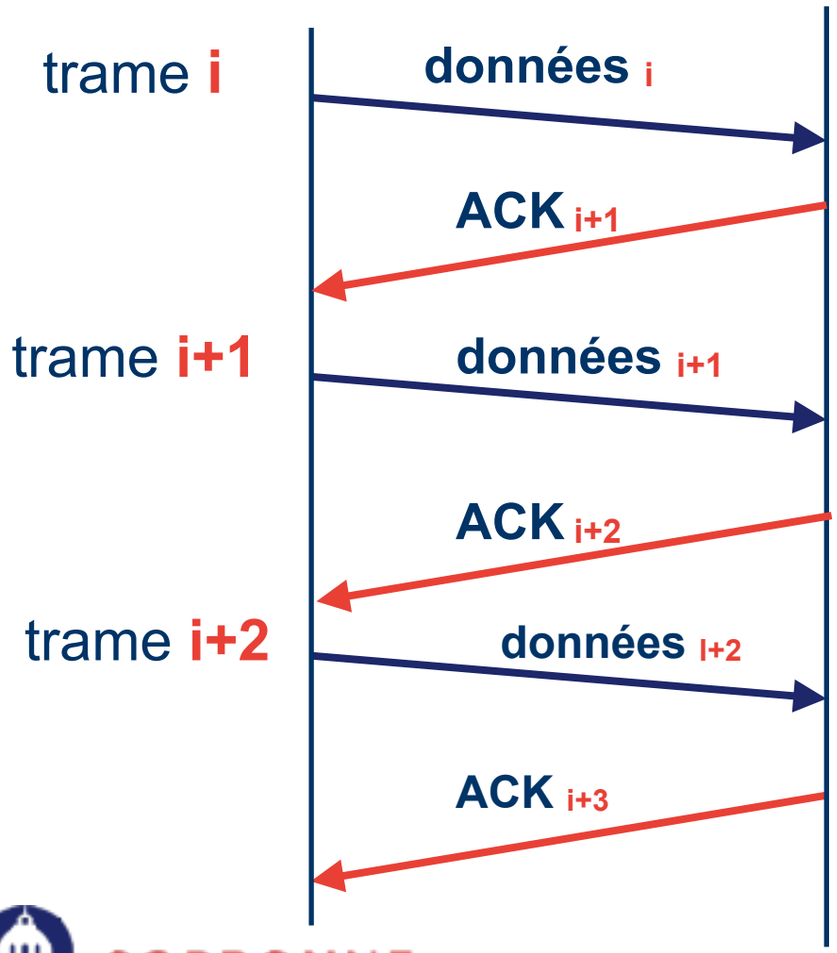
Par convention

- le numéro d'**ACK$_{i+1}$** indique le numéro de la prochaine trame attendue
- L'**ACK$_{i+1}$** acquitte toutes les trames dont le numéro est inférieur i+1

Protocole Send & Wait : chaque trame de données est acquittée avant l'envoi de la suivante

Protocole en mode connecté

Avec les protocoles en mode connecté les trames de données sont **numérotées** *en séquence*



Possibilité d'utiliser des

ACK groupés

(Le protocole n'est plus ~~Send&Wait~~)

Combien de trames peut-on envoyer successivement sans avoir reçu d'ACK ?

ACK i+3
acquitte la trame **i+2** et toutes les précédentes

Protocole en mode connecté

La **fenêtre d'anticipation** définit l'ensemble des numéros de trames que l'on est autorisé à transmettre

- la fenêtre est une liste de ***W*** numéros de séquence
- l'émetteur est autorisé à envoyer les ***W*** trames de données dont le ***N(S)*** est tel que :

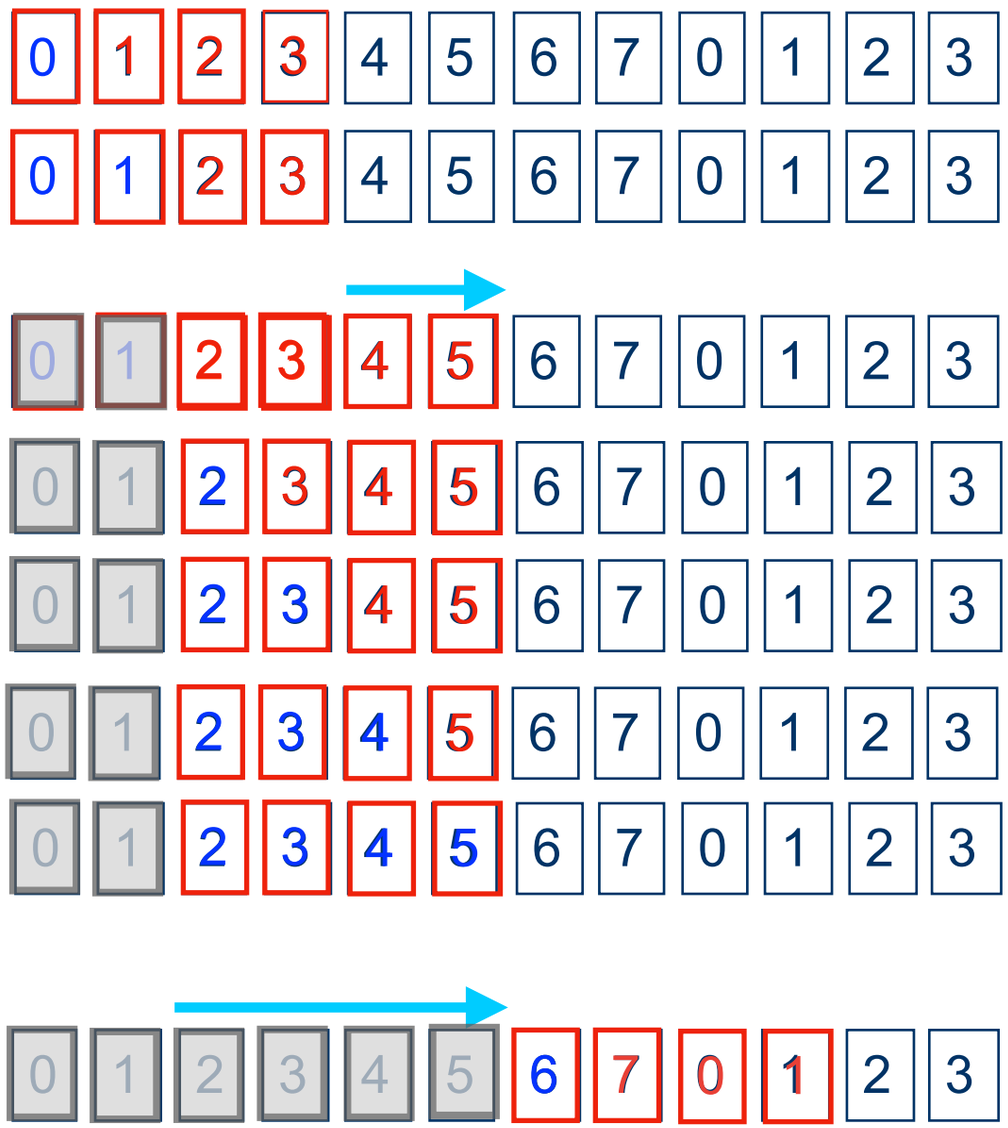
dernier *N(R)* reçu ≤ ***N(S)*** < dernier *N(R)* reçu + *W + 1*

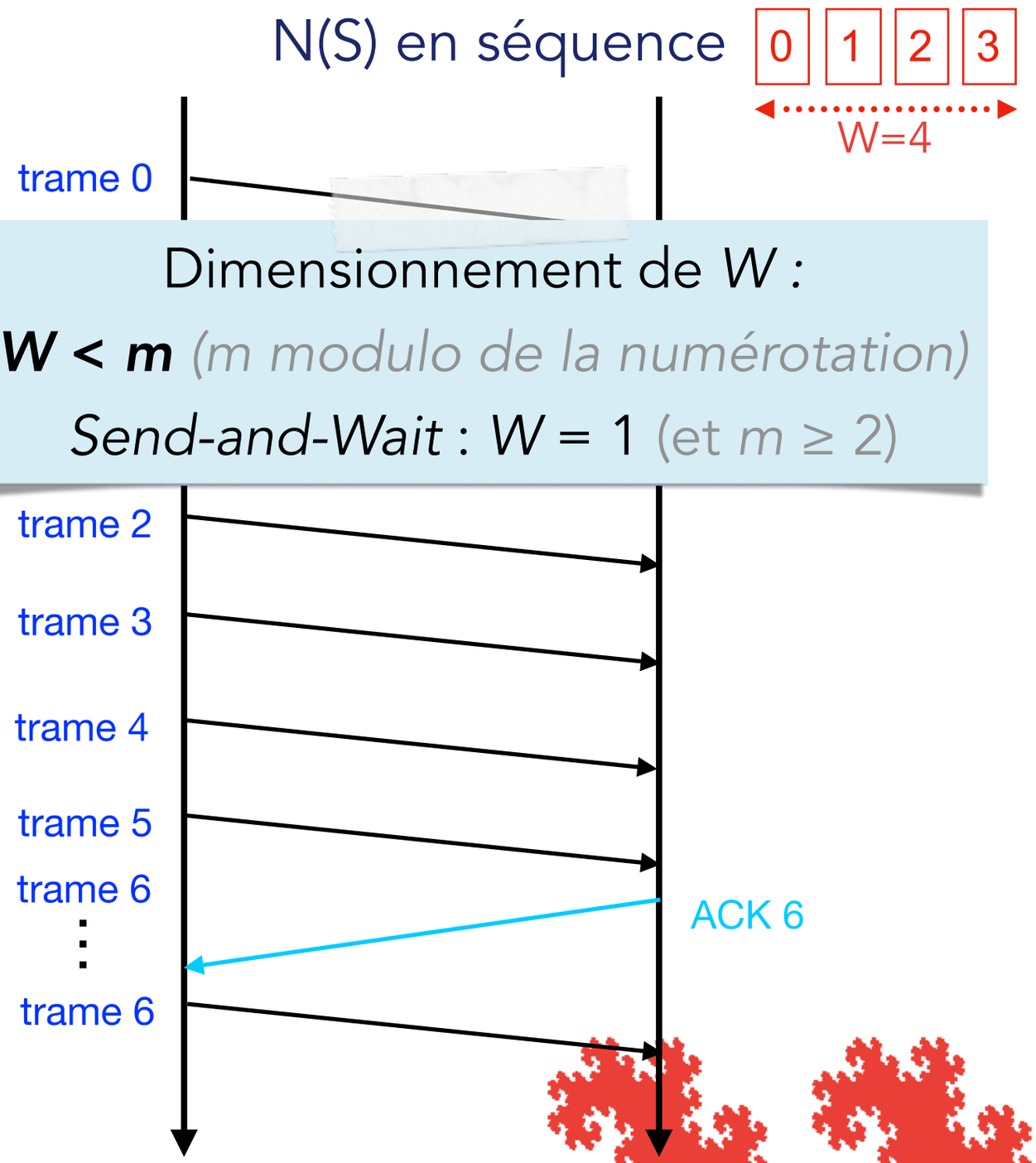
N(R) : numéro d'ACK

N(S) : numéro de la trame de données

Fenêtre d'anticipation

W = Fenêtre de trames qui peuvent être transmises

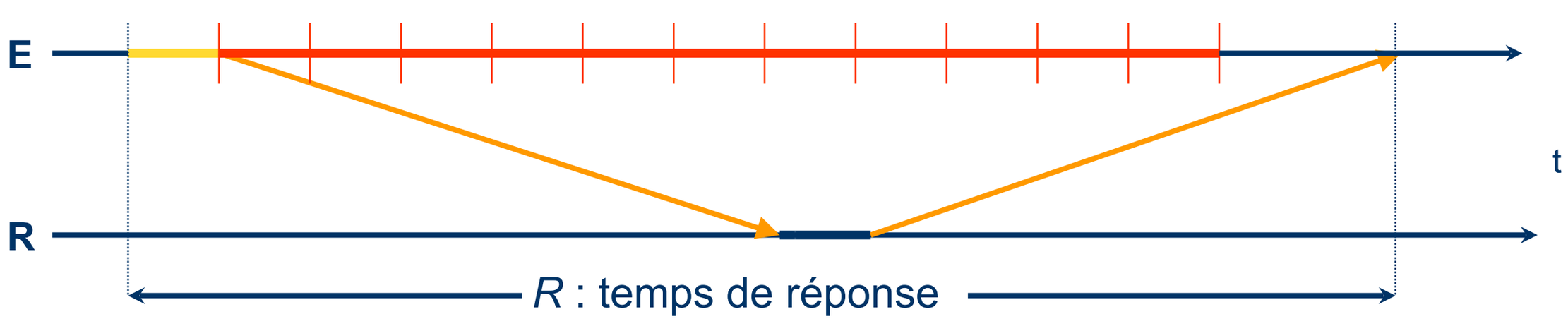




SORBONNE UNIVERSITÉ

Efficacité d'une liaison dédiée

Idée : permettre à l'émetteur d'envoyer plusieurs trames consécutives avant de se bloquer en attente d'acquittement



- Efficacité est ainsi améliorée sur une liaison dédiée.
- Qu'en est-il si le support de communication est partagé ?

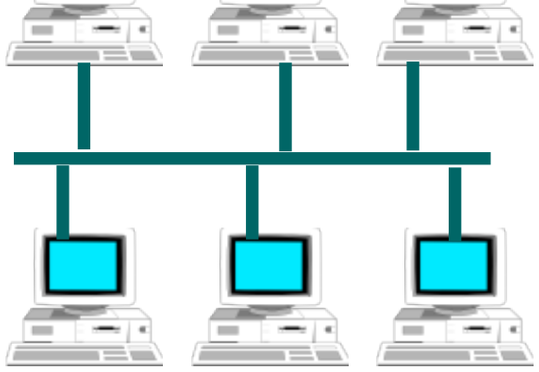
Liaison de données - Plan

- Rôle de la liaison de données
- Fiabiliser la transmission
  1. Déterminer si tous les bits ont été reçus
  2. Contrôle d'erreurs
- **Efficacité des protocoles de la couche liaison de données**
  1. **Support dédié : efficacité de la liaison**
  2. **support partagé : empêcher/réduire les collisions**

L'accès au médium

- Problème : lorsque le support est partagé par plusieurs stations, celles-ci ne peuvent pas utiliser simultanément le support

réseau d'accès mobile



LAN

Il faut déterminer, dans les réseaux à diffusion, qui accède au médium quand il n'y a qu'un seul canal.

C'est le rôle des **méthodes d'accès**

L’accès au médium

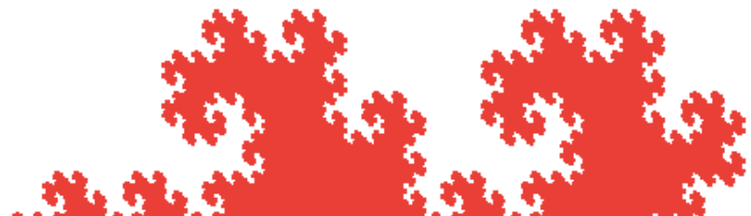
Politique d’accès

dynamiques

- allocation déterministe
  - anneau à jeton
  - méthode par scrutation (polling)
  - …
- allocation aléatoire
  - Aloha
  - CSMA/CD
  - CSMA/CA
  - …

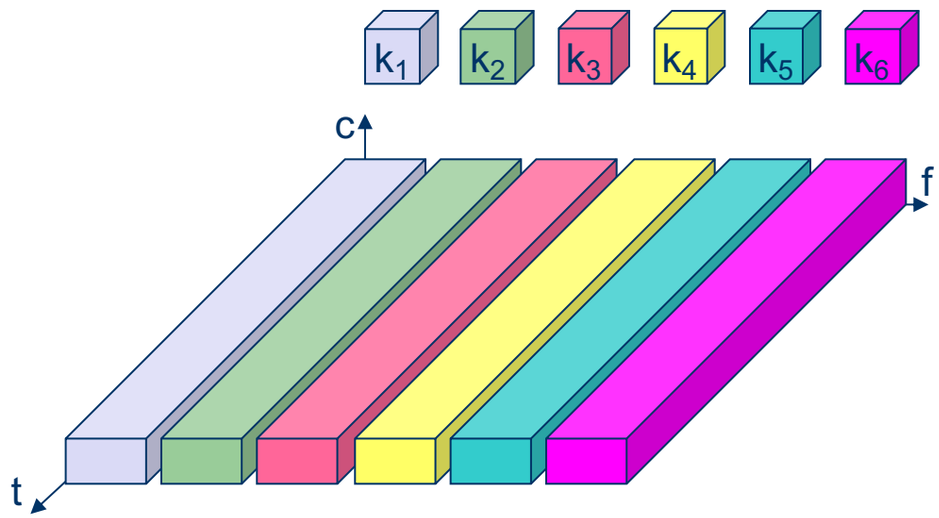
statiques

- Multiplexage fréquentiel
- Multiplexage temporel
- …



Méthodes d'accès - statiques

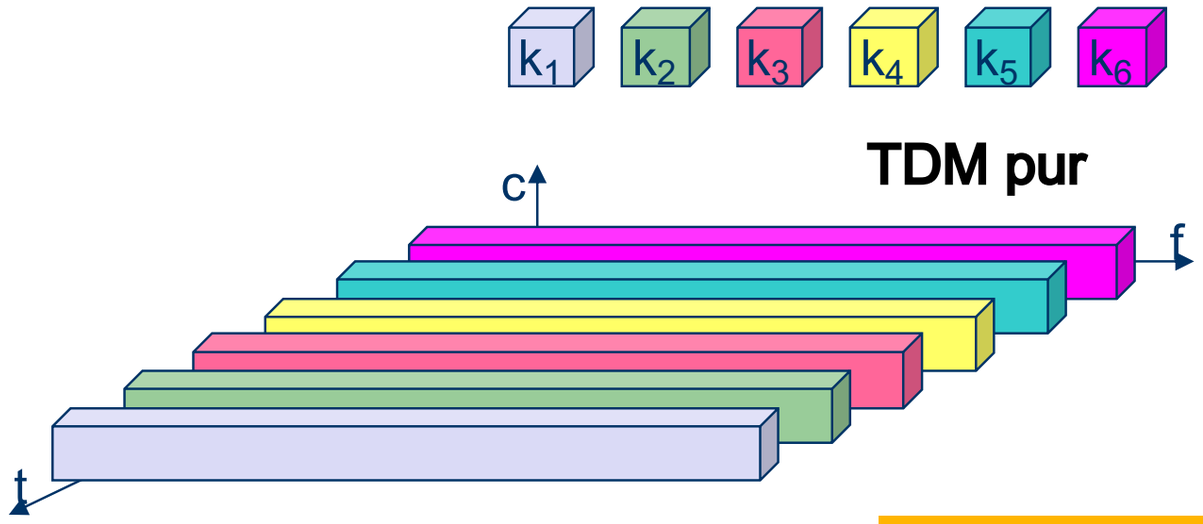
Multiplexage fréquentiel (AMRF)



+ Pas besoin de coordination dynamique (AMRF)

canal réservé
pas de collisions

Multiplexage temporel (AMRT)



inflexible

Gaspillage potentiel de bande passante

besoin d'espaces de garde (AMRF)

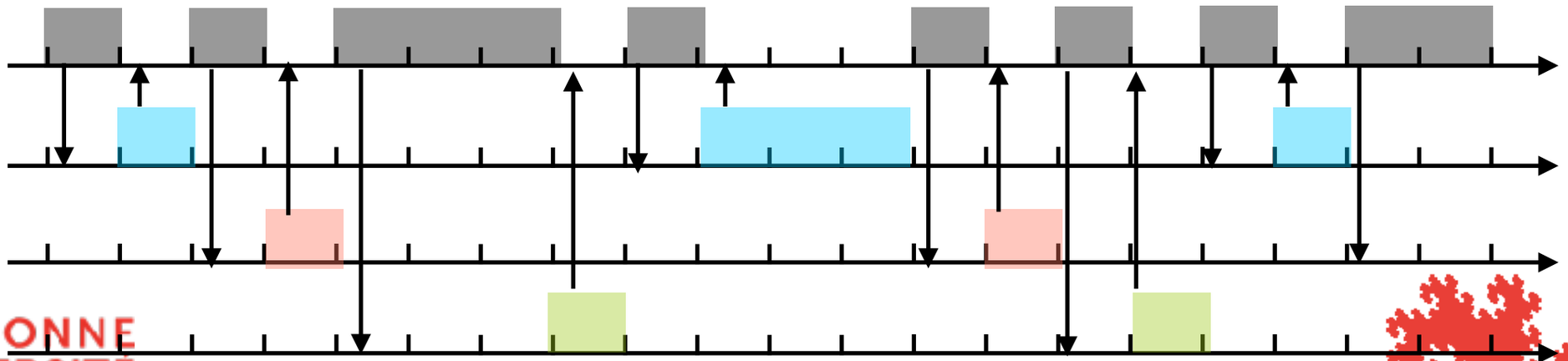
besoin d'une synchronisation précise (AMRT)

SORBONNE UNIVERSITÉ

Méthodes d’accès - dynamiques

Méthodes d’accès déterministes :
garantissent que chaque station peut transmettre au bout d’un certain temps (ex : temps du cycle)

Mécanisme d’accès par scrutation (polling)
une station *maître* interroge à tour de rôle toutes les stations (*esclaves*) pour savoir si elles ont des données à transmettre



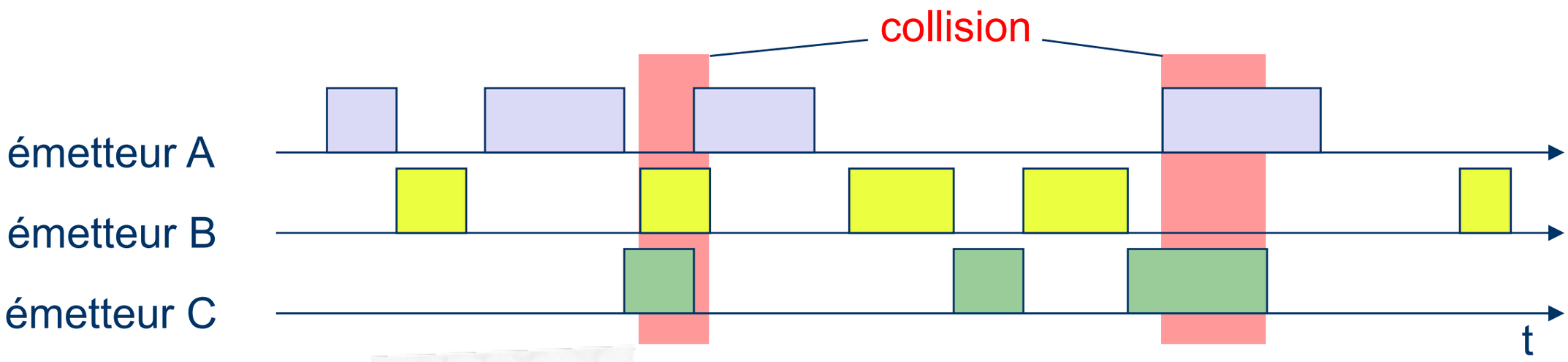
Méthodes d’accès - dynamiques

Méthodes d’accès aléatoire :

aucune garantie sur l’accès au canal

1 . Méthode **Aloha** :

Si E a des données à transmettre, il émet immédiatement



Risque de collision

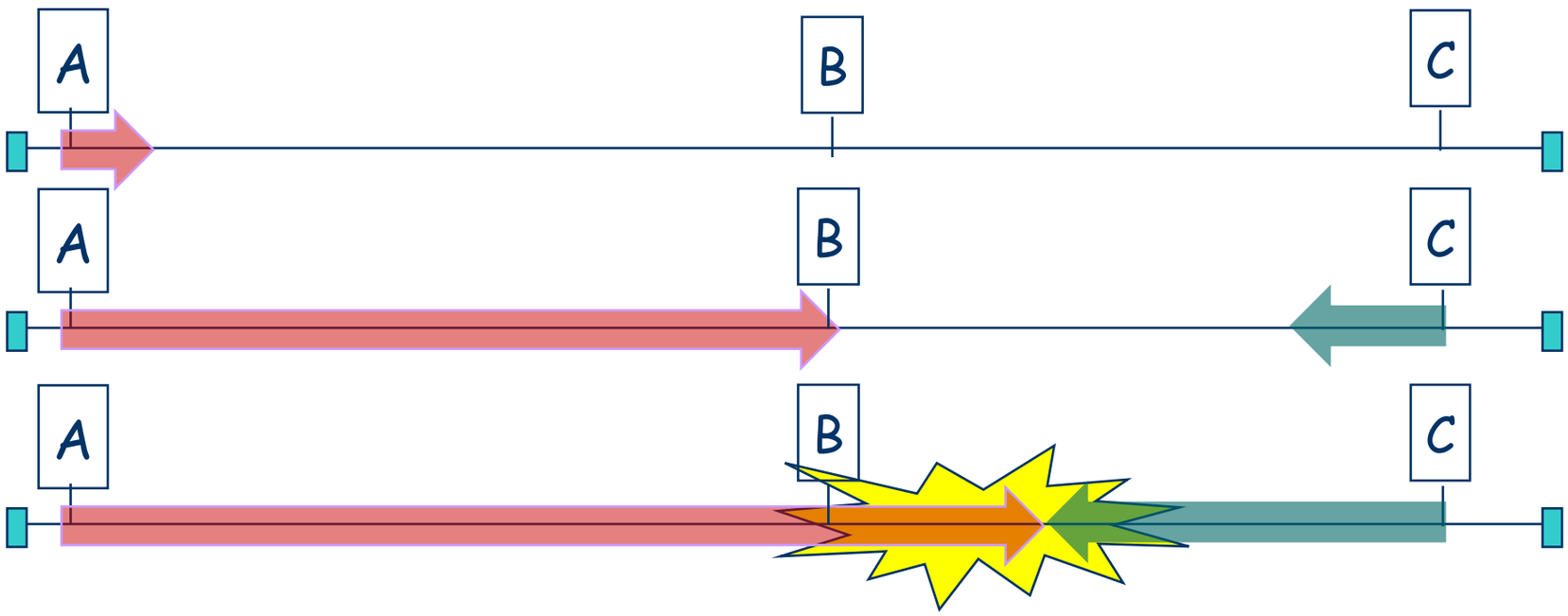
Méthodes d'accès - dynamiques

Méthodes d'accès aléatoire :

aucune garantie sur l'accès au canal

2. Méthodes du type **CSMA (carrier Sense Multiple Access)** :

- **écoute avant de transmettre** : l'émetteur transmet si le support est libre



Risque de collision

Méthodes d'accès - dynamiques

Méthodes à accès aléatoire :

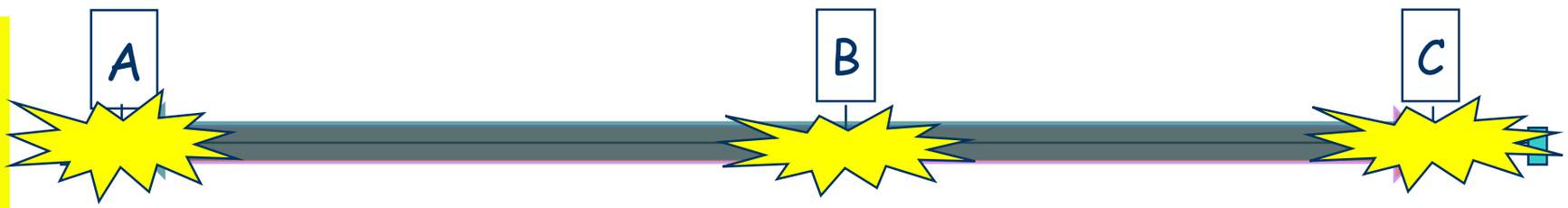
aucune garantie sur l'accès au canal

2. Méthodes du type **CSMA (carrier Sense Multiple Access)** :

- **écoute avant de transmettre** : l'émetteur transmet si le support est libre
- **l'émetteur** doit être en capacité de **détecter les collisions** (CSMA/CD : collision détection)

contrainte pour garantir la détection : $T_t > 2T_p \Leftrightarrow \frac{L}{D} > \frac{2d}{v_p}$

Risque de collision



Méthodes d'accès - dynamiques

Méthodes à accès aléatoire :

aucune garantie sur l'accès au canal

2. **CSMA/CD (carrier Sense Multiple Access/Collision Détection)** :

- **écoute** : l'émetteur E transmet si le support est libre
- *E* doit être en capacité de **détecter les collisions**
- **éviter de nouvelles collisions** : une fois la collision détectée par E et que le support redevient libre, E attend alors un délai tiré aléatoirement* (nombre d'intervalles de temps (IT))

*selon l'**algorithme du retrait exponentiel**

Algorithme du retrait exponentiel

Après 1 collision :	délai choisi entre 0 et 1 IT
Après 2 collisions successives :	délai choisi entre 0 et 3 IT
Après **i** collisions successives :	délai choisi entre **0** et **2^i-1** IT

Méthodes d'accès - dynamiques

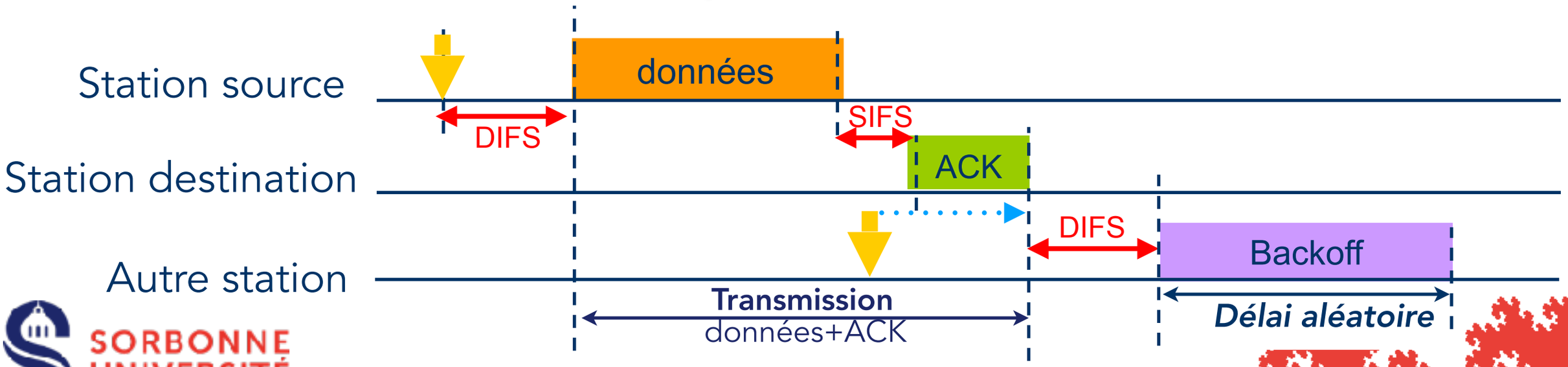
Méthodes à accès aléatoire :

3. Méthodes du type **CSMA/CA** du Wi-Fi
(carrier Sense Multiple Access/Collision Avoidance)

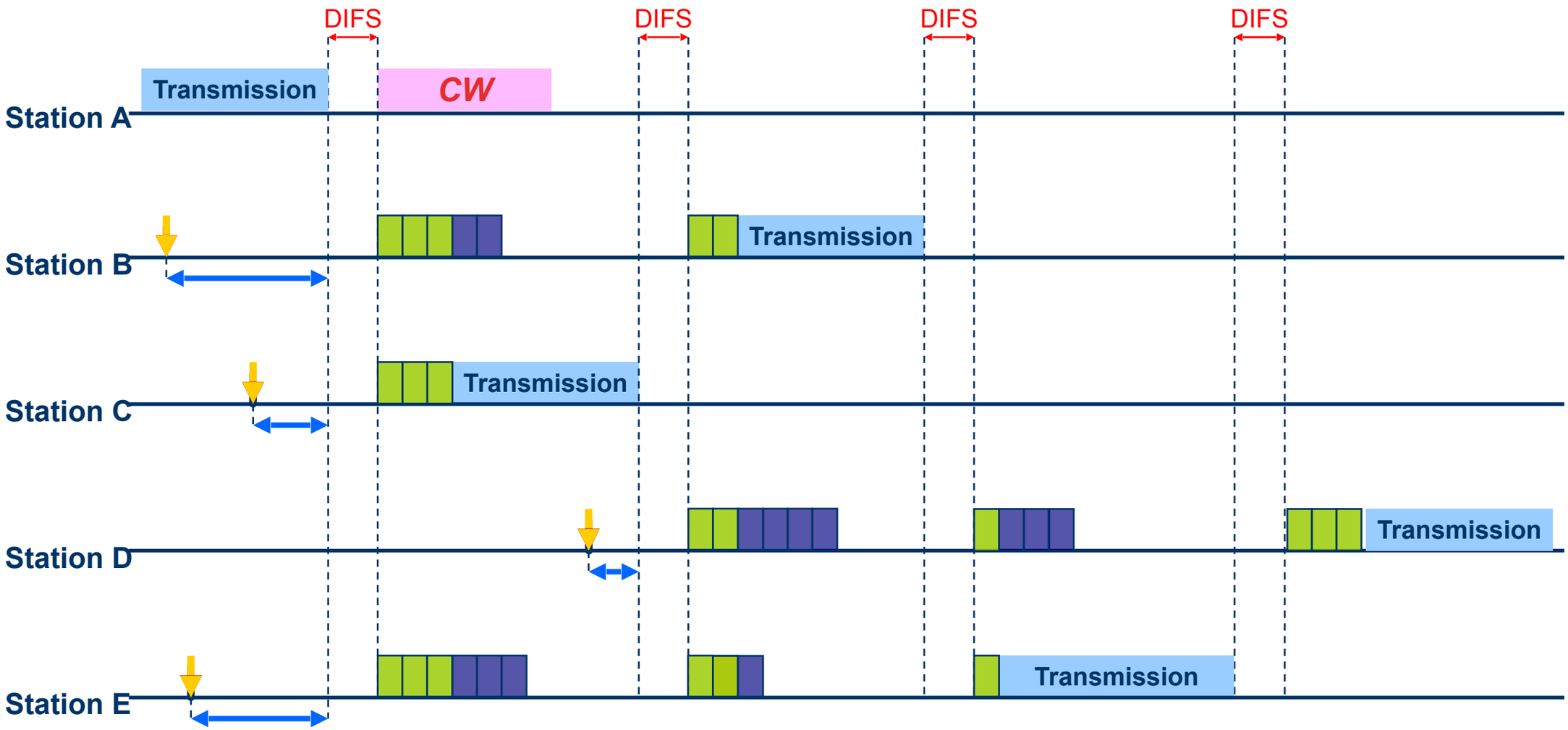
1er Backoff (pas de collision) ∈ [0,7]
2e Backoff (après 1 collision) ∈ [0,15]
3e Backoff (après 2 collisions) ∈ [0,31]

Temporisateurs permettent d'instaurer un système de priorités mais pas de garanties fortes :

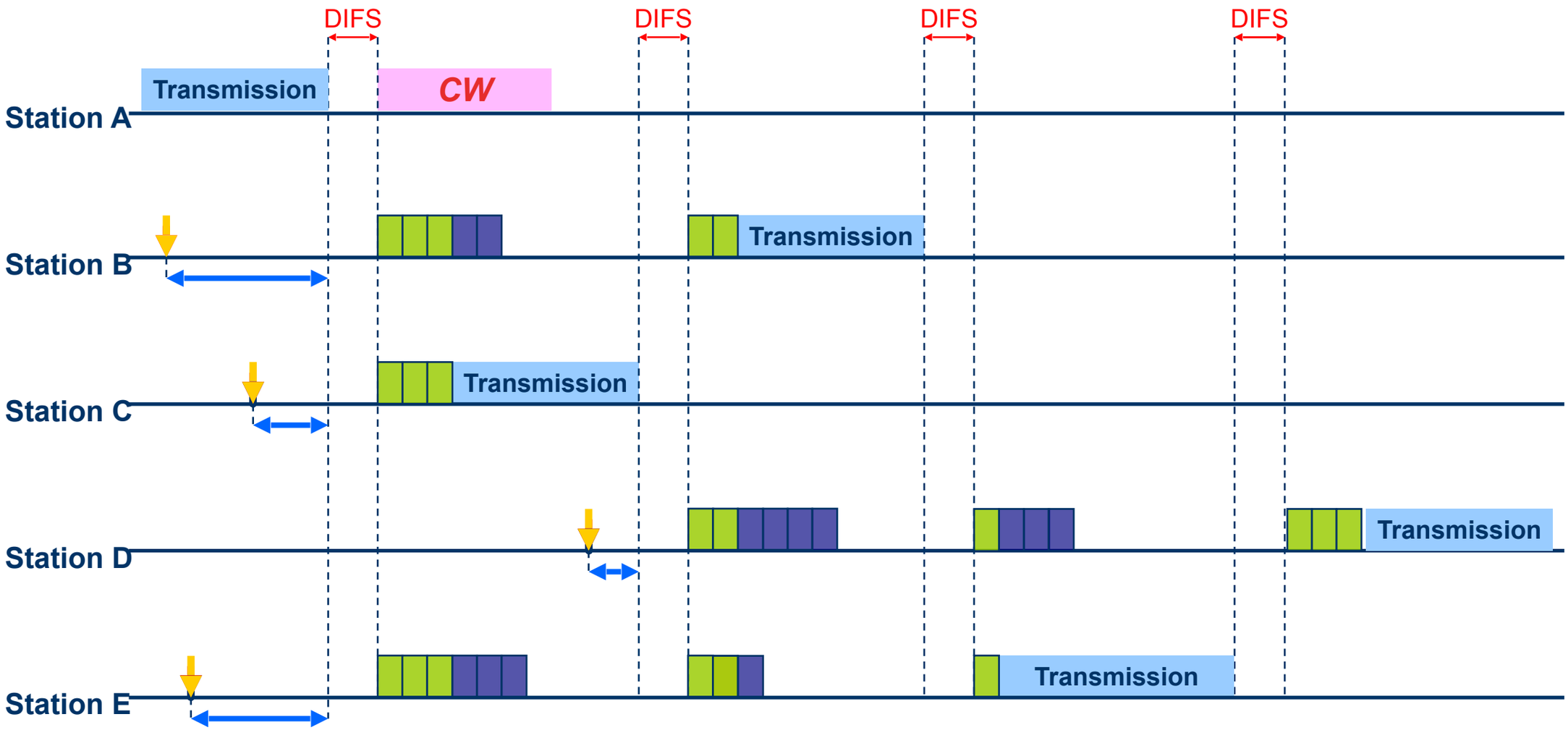
- SIFS (Short Inter Frame Spacing),
- DIFS (Distributed Coordination Function IFS)



Méthodes d'accès - dynamiques



Méthodes d'accès - dynamiques



Liaison de données - conclusion

Rôle de la couche liaison de données : Transmettre des données entre des E/R sur un support dédié ou partagé

Les protocoles de la couche liaison de données :

- échangent des **trames** (suite de bits + entêtes) de *données /contrôle*
- **fiabilisent la liaison** grâce à des mécanismes :
  - détection d'erreurs (codes polynomiaux)
  - pour la retransmission (ACK, tampon, temporisateurs…)
- le service du **protocole** dépend du support de communication
  - Support dédié : service **fiable** (si le protocole est en mode connecté) et l'efficacité est améliorée (**fenêtre d'anticipation**)
  - Support partagé : service principal consiste à partager l'accès au canal (méthodes d'accès) et empêcher/réduire les **collisions**