

Réseaux 1

Couche liaison de données

Abdelkader OUALI
abdelkader.ouali@unicaen.fr

Université de Caen Normandie
Laboratoire GREYC

Plan

- ♦ **Rappel et encore plus sur la couche 2**
 - **Trame**
 - Découper une chaîne de 0/1 en fragments
 - **Contrôle d'erreur**
 - Détecter et corriger les trames reçues
 - **Protocoles d'accès au support**
 - ALOHA, CSMA/CD, Token ring
 - **Adressage**

Construction d'une trame

- ♦ **Décomposer** un flux de bits en plus petits, morceaux ou fragments appelés **trames**
 - Permet le partage du support de communication : plusieurs expéditeurs et/ou destinataires peuvent multiplexer la liaison dans le temps.
 - Chaque trame est identifiée séparément et d'une manière unique
- ♦ Fournit une **unité gérable** pour la gestion des erreurs
 - Facile de déterminer si quelque chose s'est mal passé lors d'un envoi
 - Et peut-être même le corriger si désiré

Trame

- ♦ **Complète** les données avec quelques **informations supplémentaires**
 - L'en-tête contient généralement des **informations d'adressage**
 - Peut-être inclut une **fin/queue** (avec la **somme de contrôle** - à expliquer)
- ♦ L'unité de base de réception
 - Lien fournit la totalité ou une partie (inférieure) des données utiles de la trame
 - Généralement, une unité de transmission maximale (MTU)
- ♦ Certaines couches de liaison nécessitent également l'absence de trames
 - un espace de temps minimum entre les trames



Identification de la présence d'une trame

- ♦ **Rappel**
 - Simulateur réseaux (TP)
- ♦ Le récepteur doit savoir quand une trame **commence** et **se termine**
 - Sinon, **erreurs** dues à une **mauvaise interprétation** du flux de données
- ♦ Plusieurs **alternatives différentes** :
 - Trames de **longueur fixe** (bits)
 - Trames **explicitement délimitées** :
 - à base de longueur
 - à base de sentinelle
 - Trames **Durée fixe** (secondes)

Trames de longueur fixe

- ♦ Facile à gérer pour le récepteur
 - Exigences bien maîtrisées de mise en mémoire tampon
- ♦ Introduit des **inefficacités** pour les trames de **longueur variable**
 - Peut **gaspiller** de **l'espace** (**bourrage**) pour des données de taille petite
 - Les données de grande taille doivent être fragmentées sur plusieurs trames
- ♦ Nécessite un **compromis** de conception **explicite**
 - ATM utilise des trames de 53 octets (cellules)
 - pourquoi 53 octets ?

Identification par une section longueur

- ♦ Pour éviter la surcharge, nous utilisons des trames de longueur variable
 - Chaque trame fournit sa longueur comme information
 - E.g. DECNet DDCMP
- ♦ **Problèmes ?**
 - Et si vous le **décodiez mal**?
 - Rappelez-vous, besoin de décoder la trame lors de sa réception
 - Encore faut-il que la trame commence correctement...



Identification à base de sentinelles

- ♦ **Idée** : marquer le **début/fin** de la trame avec un **marqueur** spécial
 - motif de bits, motif d'octets, motif de signal
 - Mais ... doit s'assurer que le marqueur **n'apparaît pas** dans les données
- ♦ **Deux solutions** :
 - **Symbole spécial de couche physique de données** (par exemple, 00000 dans 4B / 5B)
 - Impact sur l'efficacité (ne peut plus utiliser le symbole pour les données) et le code de contrôle (peut maintenant avoir de longues chaînes de 000 parfois)
 - **Bourrage (stuffing)**
 - Supprimer dynamiquement les motifs de bourrage du flux de données
 - Le récepteur décompose le flux de données pour reconstruire les données d'origine

Identification d'une trame

Bourrage de bits

- Éviter le motif de bits sentinelles dans les données utiles
- Généralement, sentinelle est le motif binaire **01111110** (0x7E)
- Inventé pour SDLC/HDLC, motif maintenant standard
- **Expéditeur** : à chaque apparition de **cinq 1** dans les données à envoyer, on insère un zéro, résultant en 011111**0**1
- **Récepteur** : à chaque apparition de **cinq 1**, supprime le **zéro suivant**

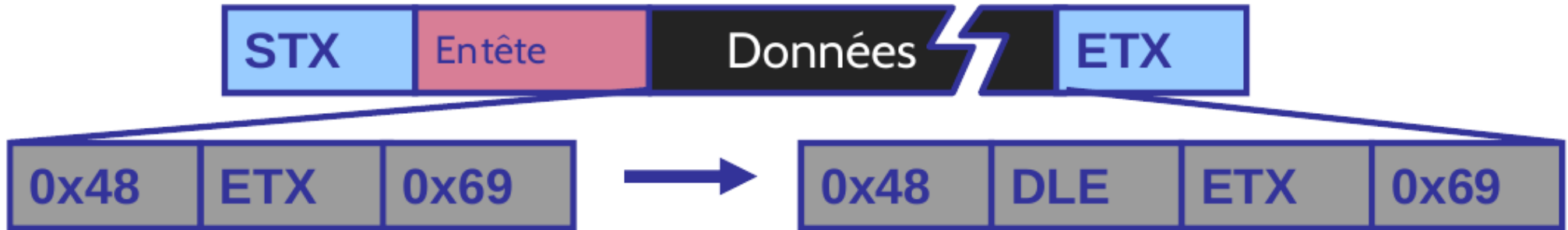
Bourrage de bits

```
011111100001110111011111011111001
011111010000111011101111100111110001
```

Identification d'une trame

Bourrage d'octets

- Similaire au bourrage de bits, sauf maintenant que c'est au niveau **octet** (**caractère**)
 - Ont généralement deux flags différents, **STX** et **ETX**
 - Trouvé dans **PPP**, **DDCMP**, **BISYNC**, etc.
- Besoin de bourrer si l'un ou l'autre apparaît dans les données utiles
 - Préfixe avec un autre caractère spécial, DLE (Data-Link Escape)
 - **Nouveau problème** : et si DLE apparaît?



Consistent-Overhead BS

- **Contrôle** de l'**expansion** de la taille des données utiles due au bourrage
 - Important pour les liens à faible bande passante ou les tampons de taille fixe
- L'idée est d'utiliser **0x00** comme sentinelle et de remplacer tous les zéros dans le flux de données avec la **distance** au prochain **0x00**



Identification d'une trame

Durée fixe

- ♦ Jusqu'ici, nous avons fait une identification sur ce **qui est** sur le support de transmission :
 - Toutes les erreurs de bits peuvent **jeter** notre trame
 - Que se passe-t-il avec le flag **manqué**? Flag **parasite**?
- ♦ Une alternative consiste à identifier une trame sur une **horloge externe**
 - Un peu comme la signalisation de couche phy
 - **échantillonnage** à des intervalles spécifiques
- ♦ Des **compromis d'ingénierie** importants
 - Pas de bits supplémentaires nécessaires dans le flux de données lui-même, mais...
 - Besoin d'une synchronisation d'horloge étroite entre l'expéditeur et le destinataire

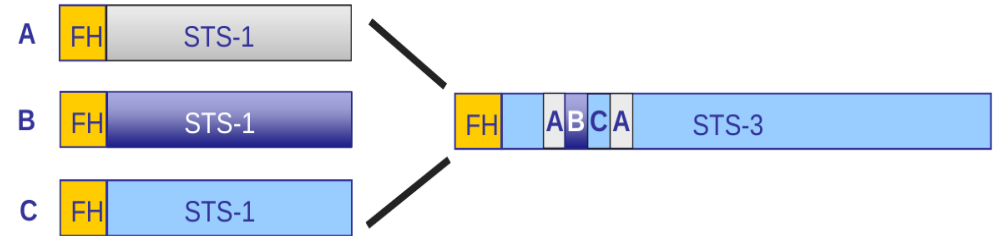
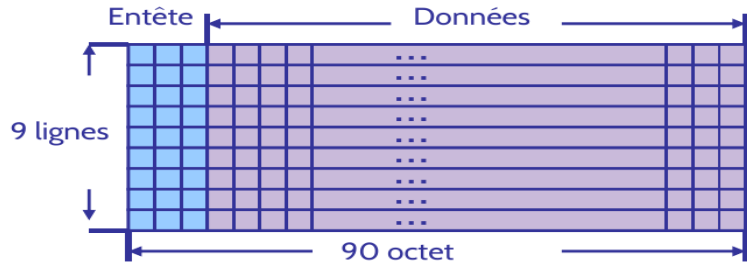
Identification d'une trame

SONET

- ♦ Toutes les trames prennent le **même temps**
 - **Indépendant** du débit binaire!
- ♦ Chaque trame commence par des bits de signal
 - Peut **synchroniser** une horloge en cherchant des bits en signal périodique
 - Pas besoin de bourrage; le **motif** du signal est **peu probable**
- ♦ Garder la synchronisation dans les trames avec les transitions
 - Encodé en utilisant **NRZ**, mais
 - On applique **xor** avec un **motif spécial** de 127 bits
 - Crée de nombreuses transitions, rend le motif du signal **improbable**

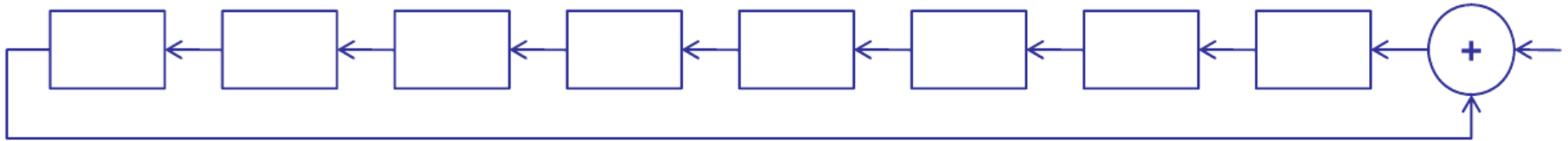
Identification d'une trame SONET

- Chaque trame STS (Signal de Transport Synchrone) fait 125 us de long (8000 trames par seconde)
- Prend en charge plusieurs débits dans le même réseau
- STS-1 est la vitesse de base (la plus lente): 51,84 Mbps
 - La trame contient 9 **lignes de 90 octets** (810 octets)
 - Les **3 premiers octets** de chaque ligne constitue l'en-tête :
 - 2 octets motif de synchronisation, un octet pour le flag
- Les liaisons à grande vitesse sont des multiples de trames STS-1
 - Par exemple, STS-3 est trois fois plus rapide que STS-1
- **Difficile de synchroniser** sur des topologies **complexes**

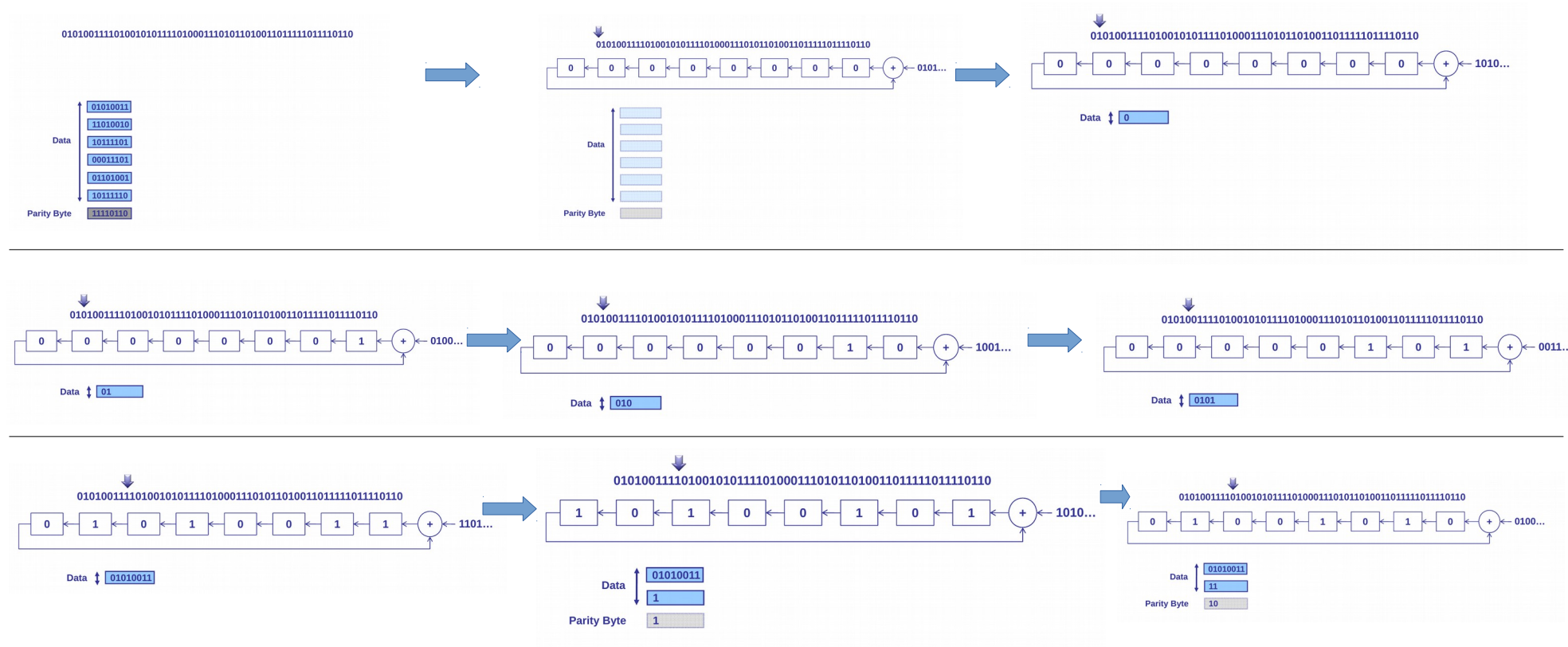


Plus sur la détection d'erreurs : Somme de contrôle (checksums)

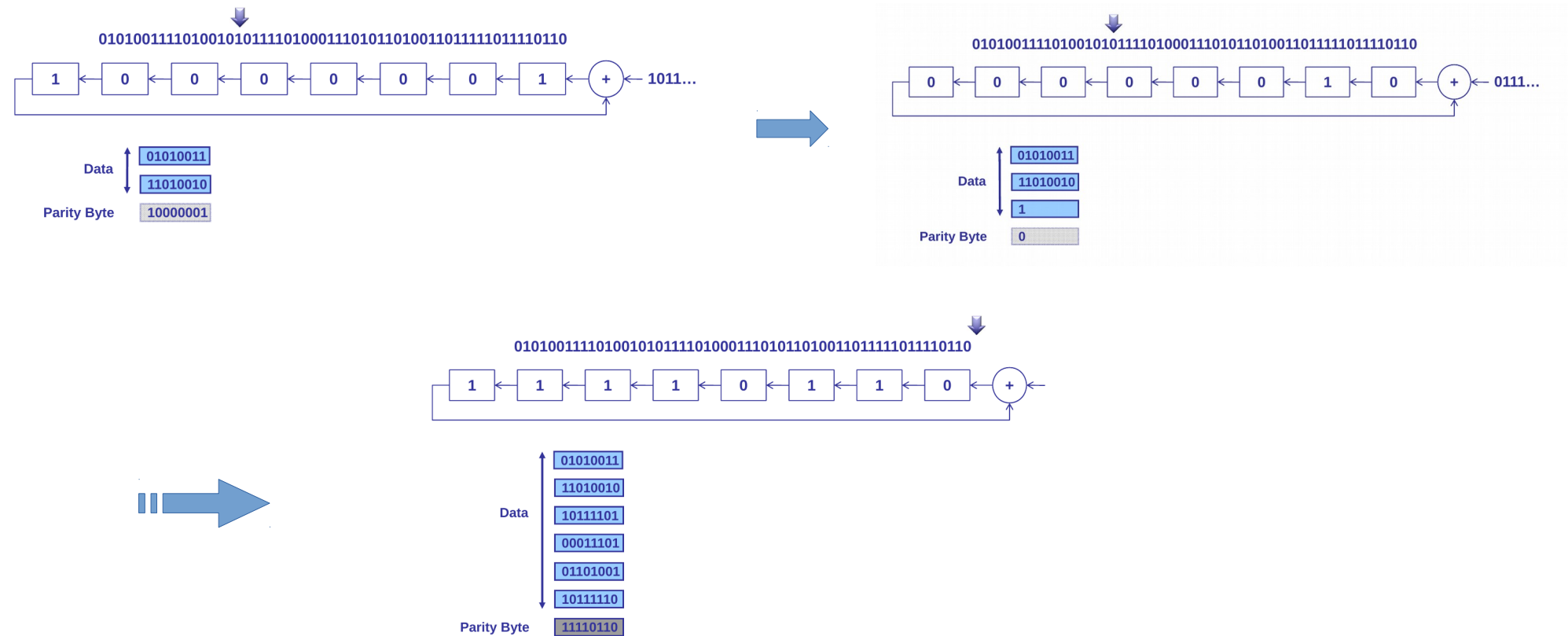
- **Idée** :
 - il suffit de faire une **somme** sur toutes les données dans la trame
 - Transmettre cette **somme** en tant que **EDC** (code de détection)
- Extrêmement **léger**
 - Implémentation facile rapide avec le matériel
 - **Fragile** : Hamming Distance de 2
- Calcul de la somme de contrôle dans l'arithmétique Modulo-2
 - L'addition / soustraction est simplement une opération XOR
 - Équivalent au calcul de la parité verticale (VRC)
- Besoin seulement d'un registre à décalage d'une longueur d'un octet et d'un port XOR
 - En supposant que les données arrivent en série
 - Tous les registres sont initialement 0



Somme de contrôle



Somme de contrôle



Accès au support (Continuer CM3)

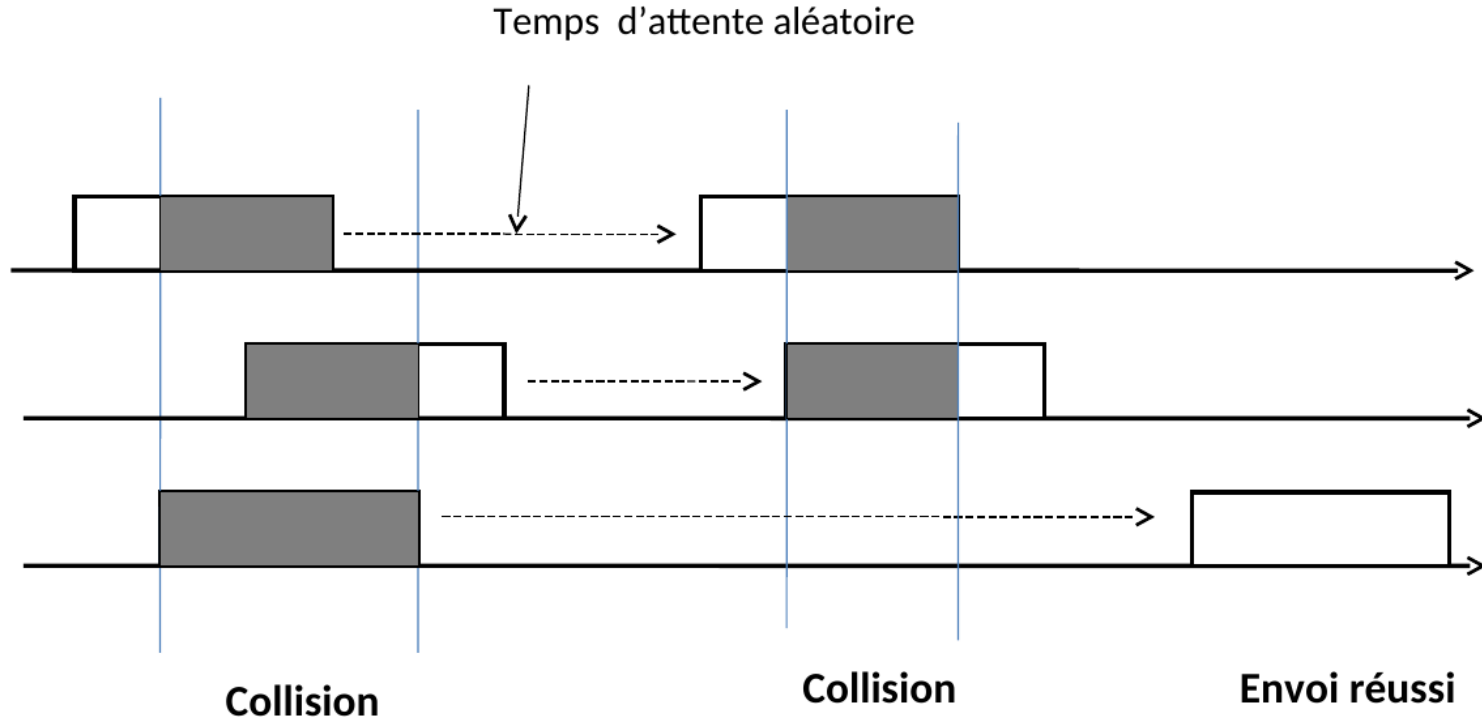
Méthode ALOHA

- ♦ **Origine** : l'Université de Hawaï (1972) a élaboré un système de communication radio ALOHA (ALOHA : bonjour en hawaïen) qui reliait les différentes îles de l'archipel.
- ♦ **Problème** : Une seule fréquence disponible partagé entre l'ensemble des stations → comment permettre à l'ensemble d'émettre des données en même temps .
- ♦ Il existe deux versions de ce protocole :
 - **ALOHA PUR**
 - **ALOHA DISCRITISE**

ALOHA PUR

- ♦ Une station qui veut émettre une trame sur le réseau, **commence immédiatement** à le transmettre.
- ♦ Si deux émetteurs ou plus émettent en **même temps** → il y a **collision** (Cela a engendré des collisions d'ondes radioélectriques)
- ♦ Les stations peuvent **détecter les collisions**. Alors elle doivent **réémettre** leur données **ultérieurement**
- ♦ L'émetteur attend une durée **aléatoire** avant de retransmettre la trame
- ♦ Cette méthode utilise **18%** du débit total dans le meilleur cas

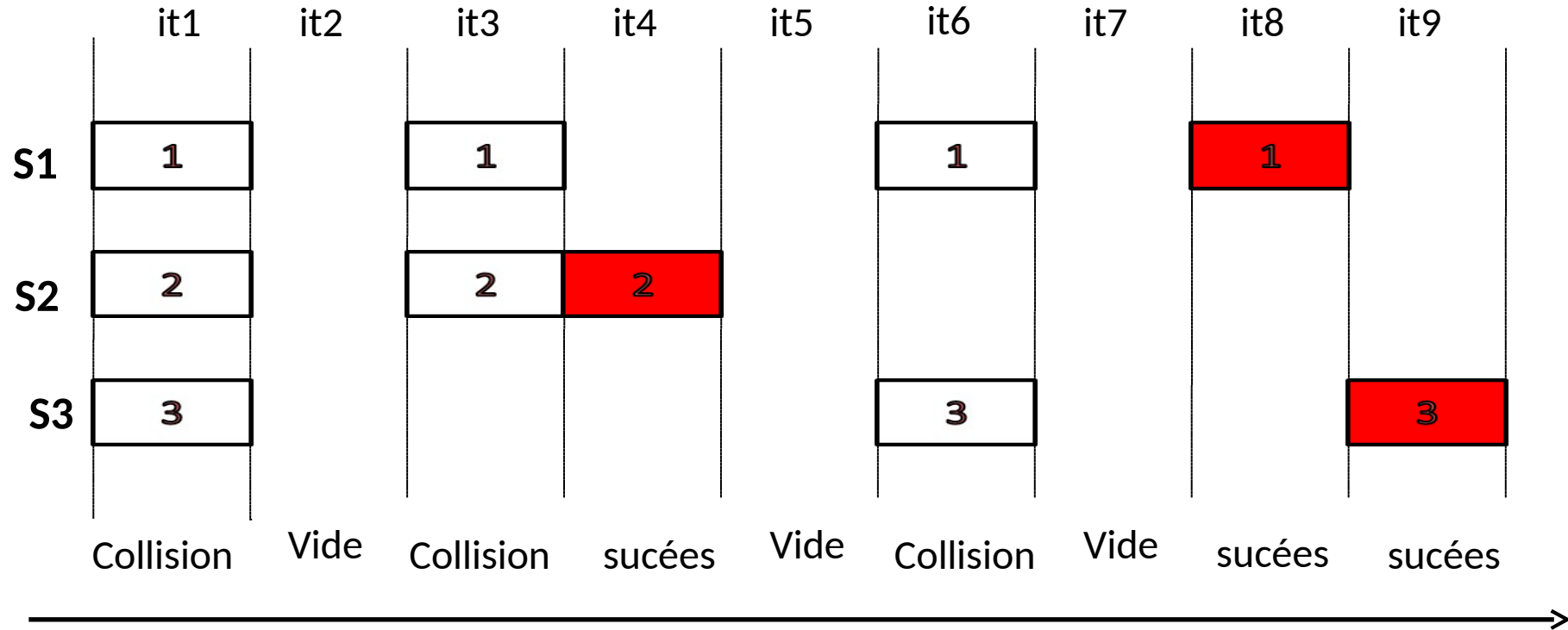
Exemple ALOHA PUR



ALOHA discrétisé

- ♦ Le temps est **découpé** en intervalles de durée T
- ♦ T : **temps d'émission** d'une trame
- ♦ Les horloges de toutes les stations sont **synchronisées**
- ♦ Le principe est le même que **ALOHA pur**, sauf que les trames ne peuvent pas être transmis qu'en début d'intervalle de temps
- ♦ Avec cette **condition** le taux d'utilisation du canal a **doublée** (37%)

Exemple : ALOHA discrétisé



Dans cette technique il y des **intervalles vide** (non utilisés) bien que les stations **veulent émettre**

Le protocole CSMA (Carrier Sense Multiple Access)

- ♦ **CSMA (accès multiple avec détection de porteuse) :**
- ♦ **Écouter** avant d'émettre
- ♦ Si le canal **libre**
 - **Transmettre** la trame
- ♦ Si le canal **occupé**
 - **Différer** la transmission
- ♦ **CSMA** une méthode qui permet de **réduire** le nombre de **conflits**
- ♦ Ce protocole **retarde** l'émission lorsque le canal est **détecté** comme **occupé**

Détection des collisions

- ♦ Ce protocole est connu sous le nom de CSMA/CD (Carrier Sense Multiple Access with Collision Détection).
- ♦ Dès que la collision a été détectée par une des stations **émettrices**, elle **envoie** sur le canal un **signal avertissant les autres stations** de la collision. Les station **arrêtent** la transmission de leurs trames immédiatement.
- ♦ Les stations **attendent** un temps aléatoire après une collision avant d'écouter une **autre fois** sur le canal
- ♦ Remarque : la détection d'une collision peut se faire soit :
 - mesurer de la **puissance du signal**,
 - comparer les signaux **émis** et **reçus**

Protocole CSMA

Persistant et non-persistant

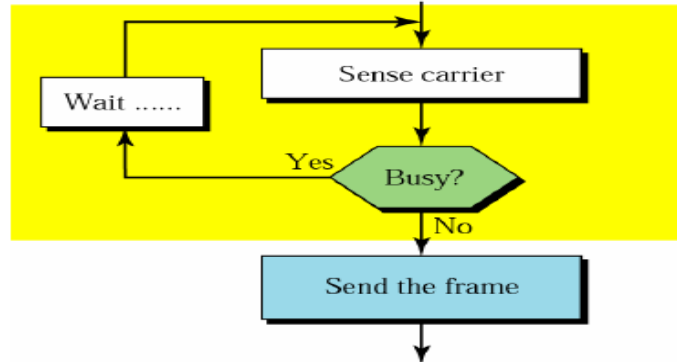
Non-persistent :

- Si la ligne est active, la station attend une période aléatoire, puis détecte à nouveau la ligne pour l'envoi
- Réduit l'efficacité du réseau

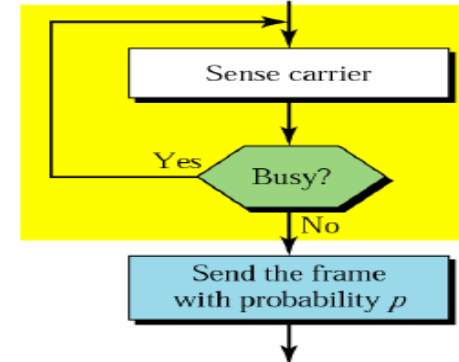
Persistant :

- 1-persistent : si la ligne est inactive, envoie la trame immédiatement (avec une probabilité de 1)
- P-Persistent : si la ligne est inactive, la trame peut ou ne pas être envoyée. Il envoie avec probabilité p et s'abstient d'envoyer avec probabilité $1-p$.

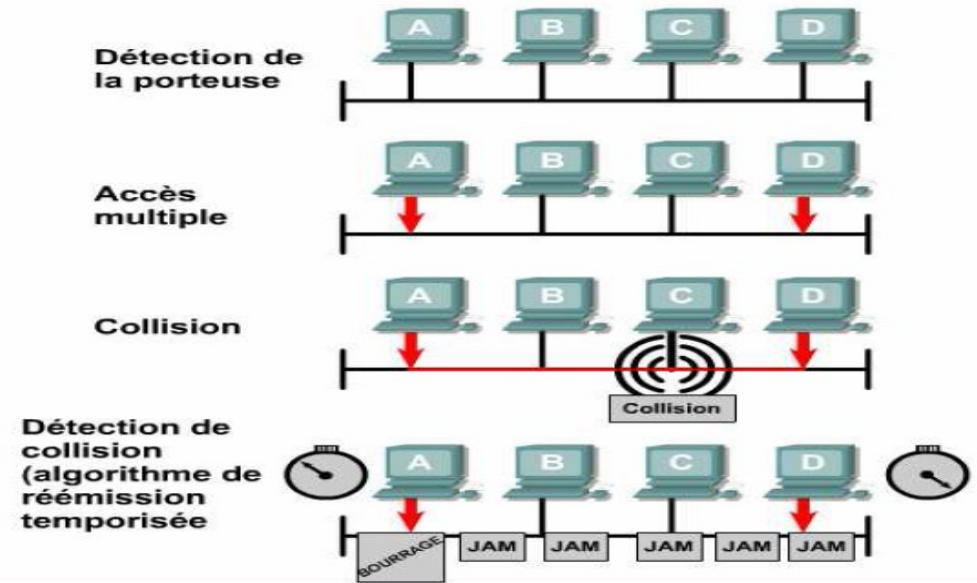
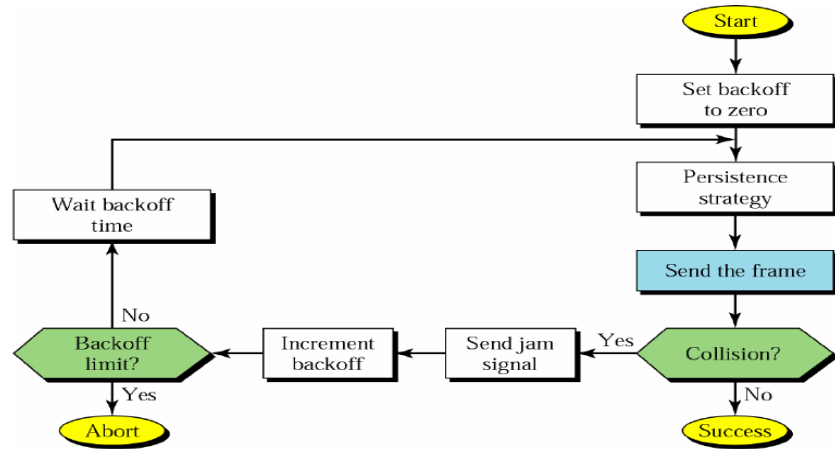
Nonpersistent strategy



Persistent strategy



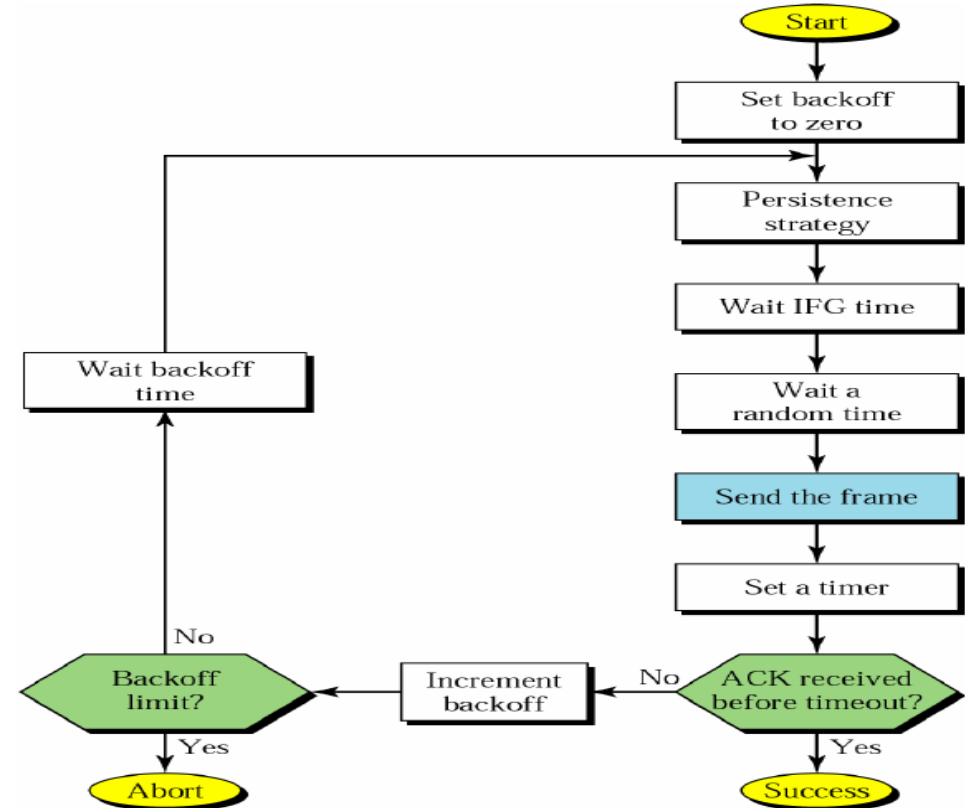
Principe du CSMA/CD (Collision Detection)



CSMA/CA (Collision Avoidance)

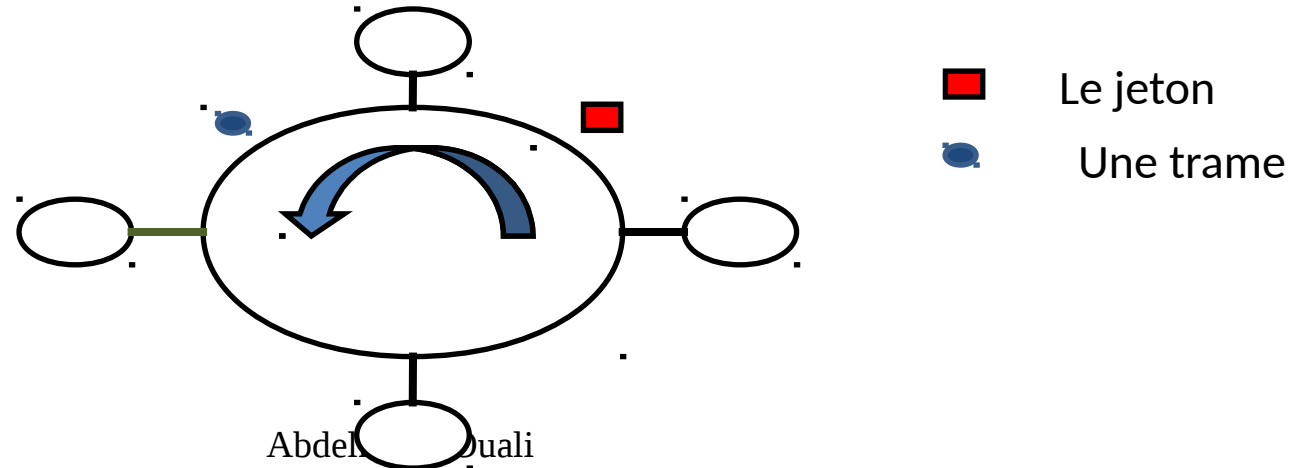
Principe

- Une fois la ligne inactive,
 - la station attend un laps de temps **IFG** (Inter-frame gap).
 - elle attend ensuite un autre laps de temps **aléatoire**
 - Après cela, elle envoie la trame et définit un timer



Les protocoles MAC déterministes : le jeton (Token)

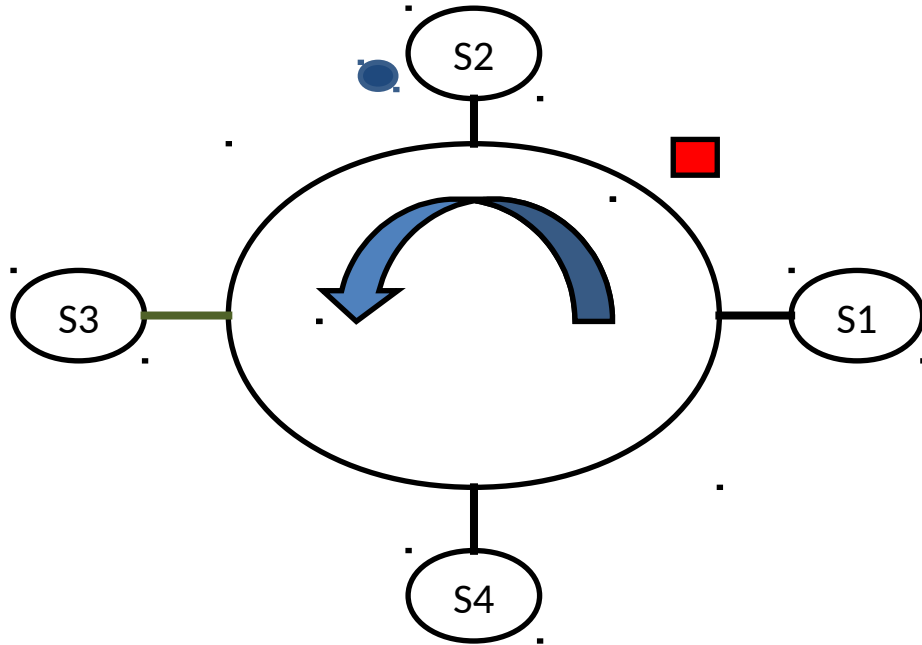
- Ce protocole est utilisé dans une **topologie en bus** ou bien en **anneau**
- Les stations sont organisés de manière à former un anneau
- Chaque station est **reliée à la suivante** et à la précédente (liaisons point à point)
- Les trames circulent dans un **seul sens** et elle **passent par toutes les stations** (autour de l'anneau)
- Un jeton est une donnée (trame) spécial qui circule autour de l'anneau.



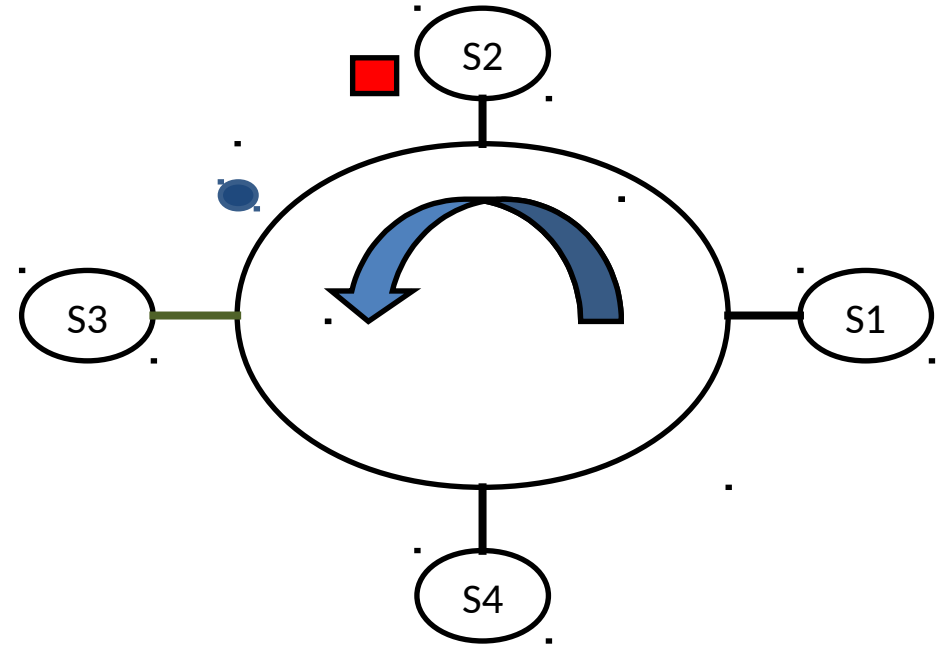
Principe du jeton

- ♦ Lorsqu'une machine désire transmettre
 - Elle prend le **jeton**,
 - Transmet les données (trames) pendant un **temps limité** et **remet** le jeton dans l'anneau.
- ♦ Une machine **ne peut pas détenir** le jeton plus qu'une durée de temps limité. Au delà, elle doit le **libérer** pour permettre au autre station d'émettre
- ♦ **Avantage**
 - Pas de collisions
- ♦ **Inconvénient**
 - Problème perte de jeton

Exemple

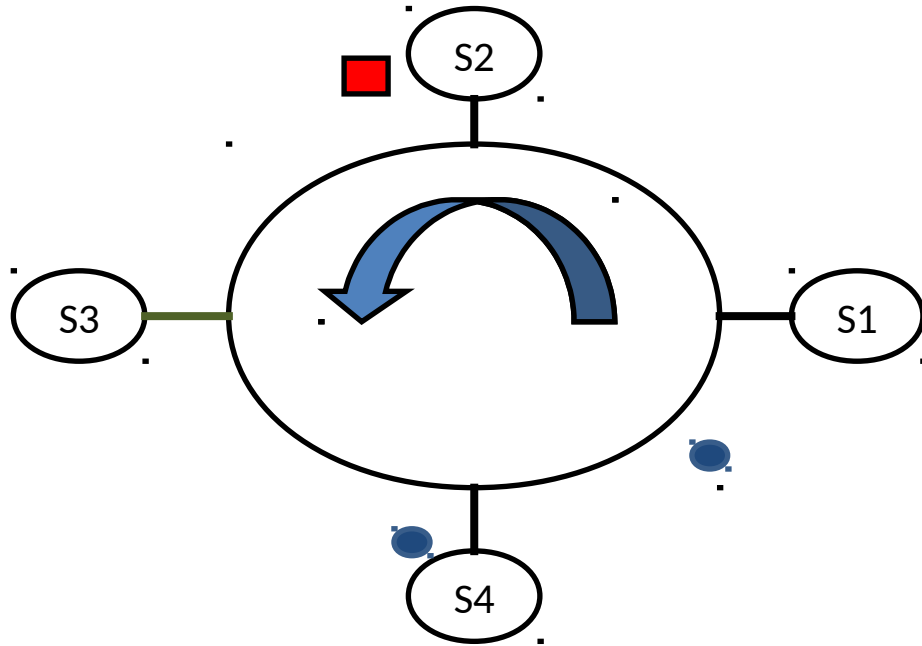


La station S2 veut émettre une trame vers la station S4. Elle doit attendre le passage du jeton

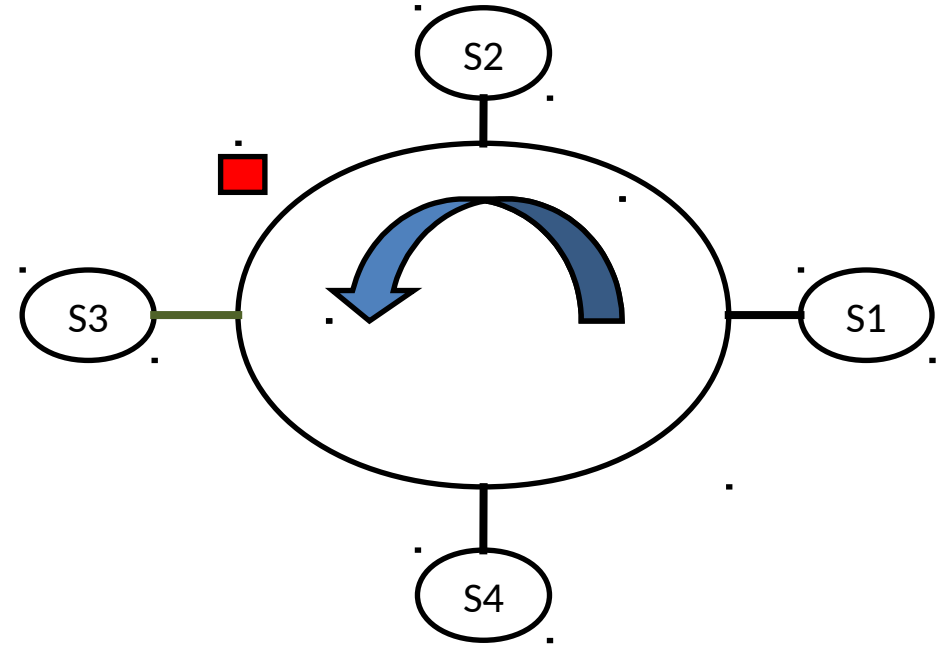


La station détient le jeton et envoie la trame. La trame doit passer par la station S3. S3 doit la retransmettre à la station S4

Exemple



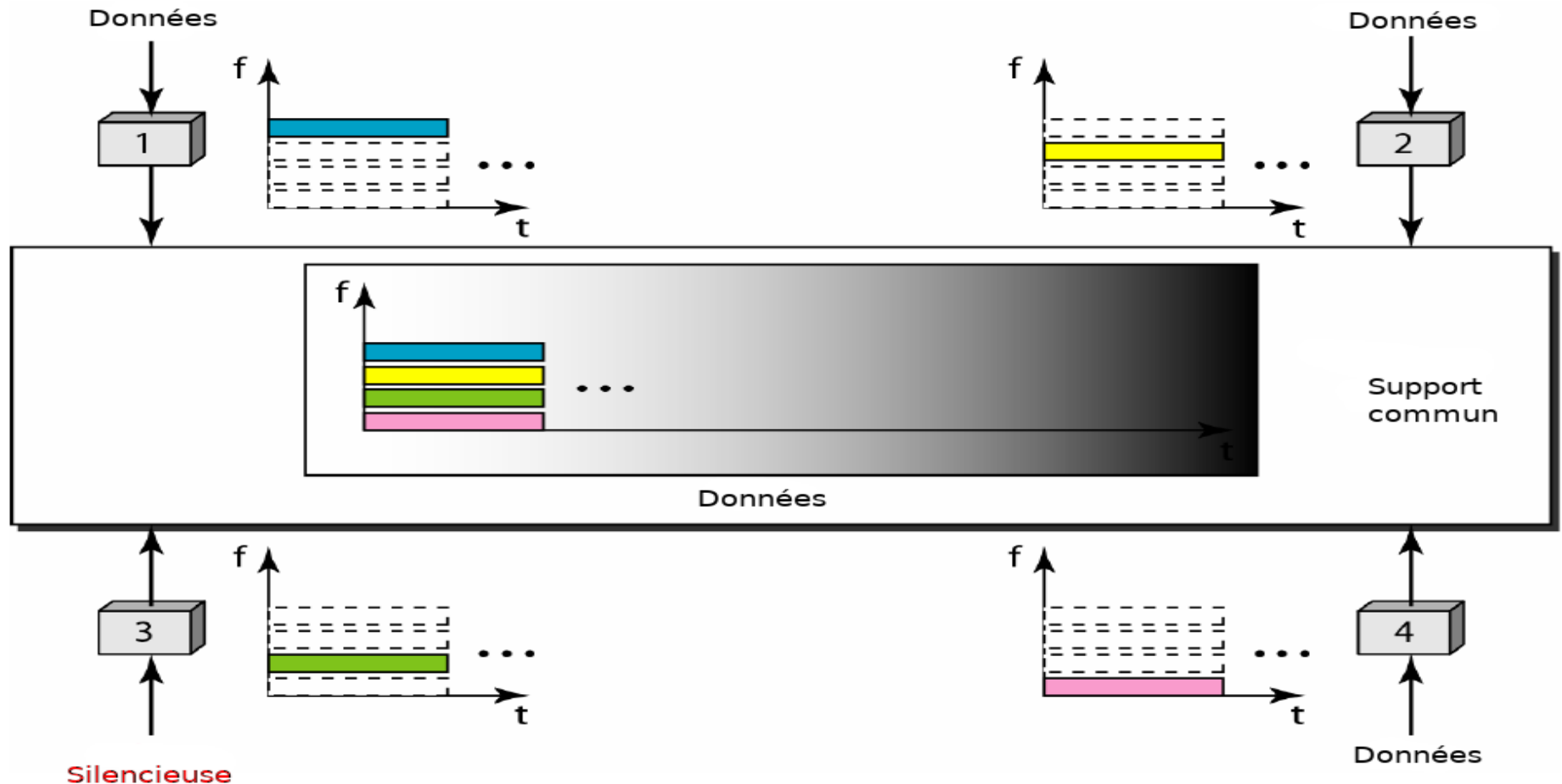
La station S4 reçoit la trame, elle garde une copie et elle laisse passer une copie



La station S2 reçoit la trame qu'elle a envoyée
Donc elle libère le jeton et détruit la trame
Le jeton peut être par la suite récupéré par n'importe quelle station.

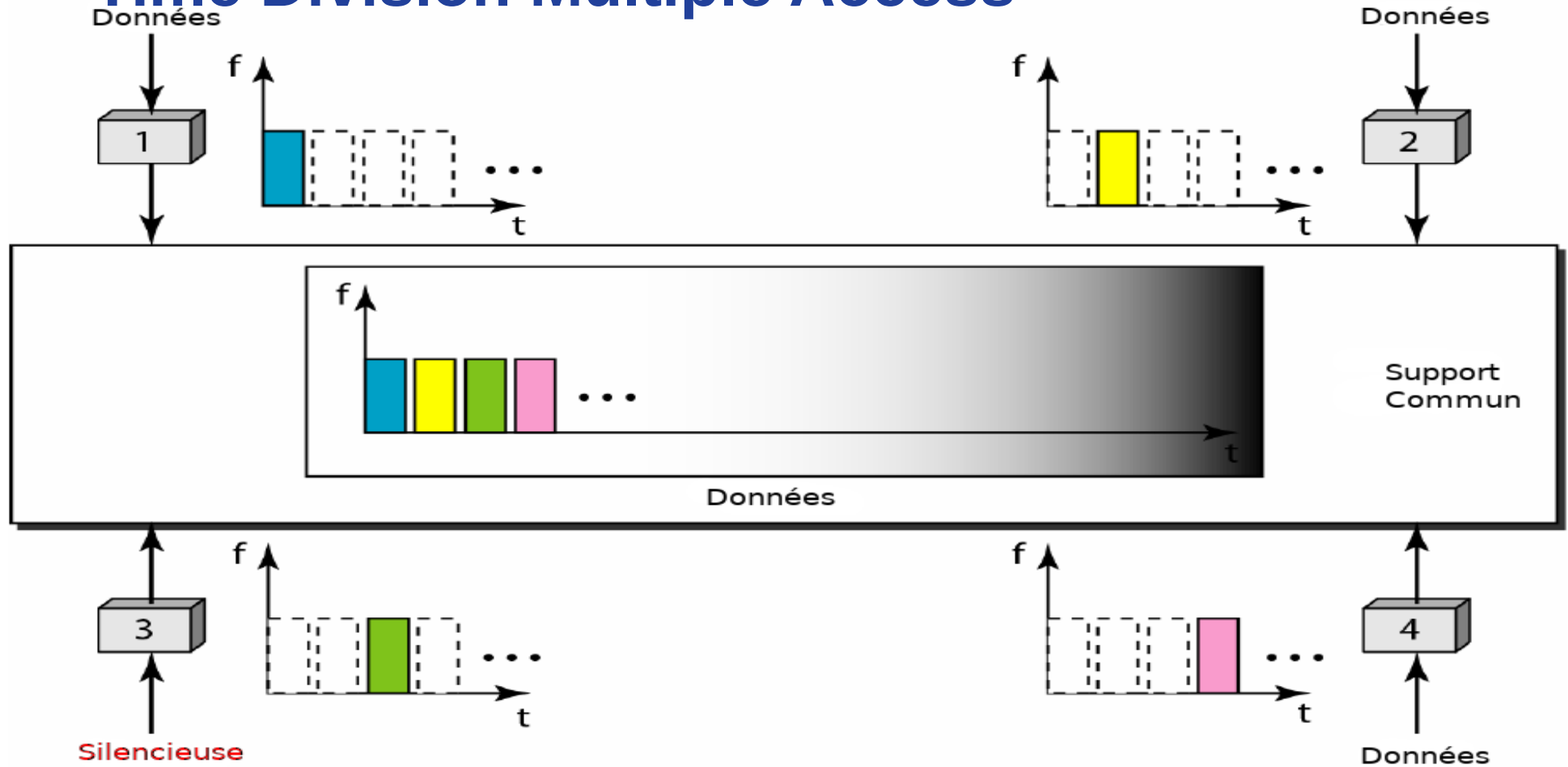
Protocole FDMA

Frequency Division Multiple Access



Protocole TDMA

Time Division Multiple Access



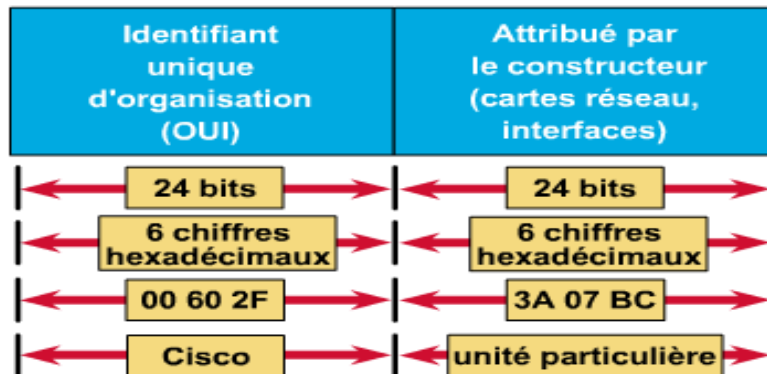
Adressage des machines

Adressage physique

- ♦ Chaque station est associée à un **identificateur appelé adresse**
 - L'adresse physique, appelée adresse **MAC** (Media Access Control), se trouve sur la **carte réseau**
 - Une adresse **universelle** est attribuée par l'IEEE à chaque constructeur.
 - Il n'y a **jamais** deux adresses physiques identiques
 - Tout ordinateur, qu'il soit relié à un réseau ou non, **possède** une adresse physique
- ♦ une station examine toutes les trames et **ne retient que celles contenant son adresse**

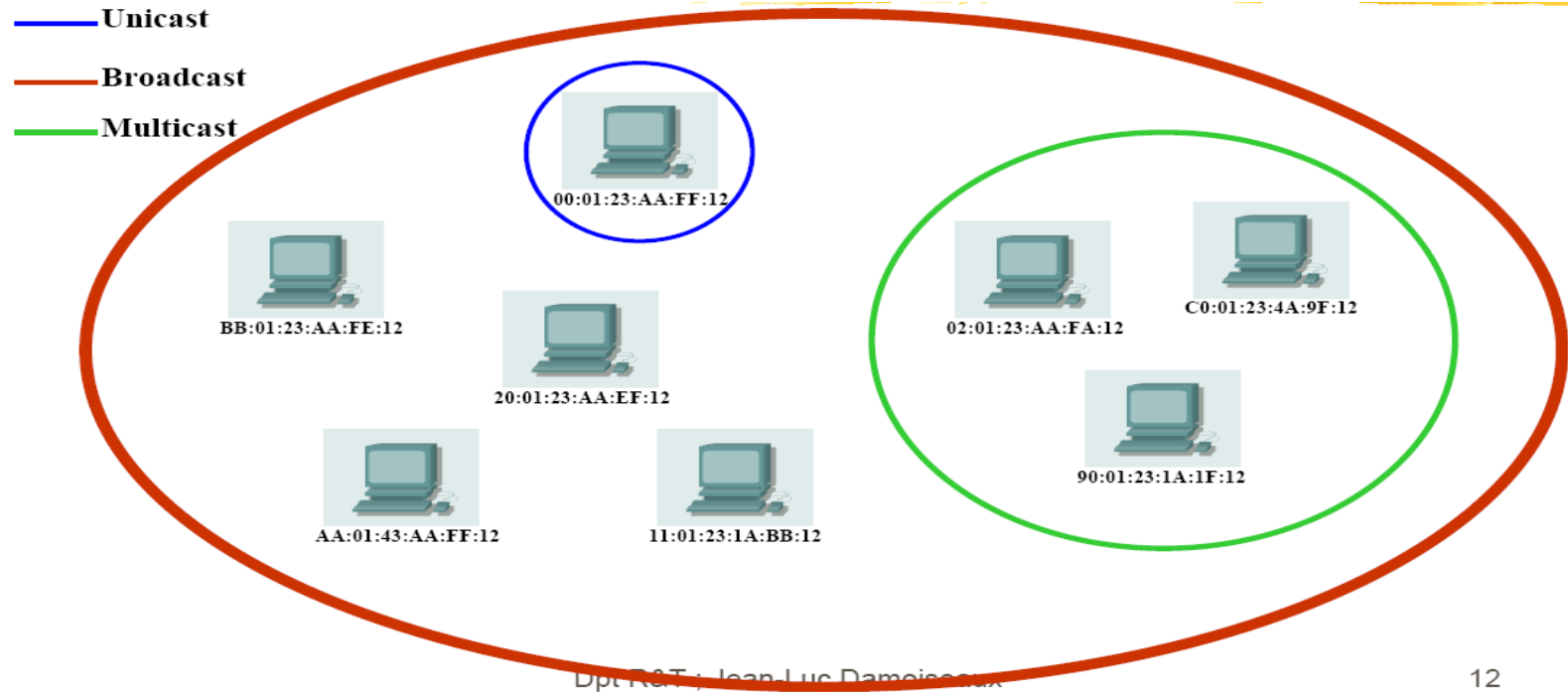
• Adressage physique

- ♦ Les adresses **MAC** comportent 6 octets (48 bits) et sont **exprimées** à l'aide de **douze chiffres hexadécimaux**:
 - Les six premiers chiffres hexadécimaux, qui sont administrés par l'IEEE, identifient le fabricant ou le fournisseur et constituent donc l'identifiant unique d'organisation (OUI - Organizational Unique Identifier).
 - Les six autres chiffres hexadécimaux forment le numéro de série d'interface ou une autre valeur administrée par le fournisseur



Constructeur	Adresse (3 octets)
Cisco	00000C
3Com	0000D8, 0020AF, 02608C, 080002
Intel	00AA00
IBM	08005A

Adresse Broadcast et multicast



Dpt R&T - Jean-Luc Damiens

12

Fin