COUCHE LIAISON DE DONNÉES

RÉSEAUX ET TÉLÉCOMMUNICATIONS

Le modèle a 5 couches de ce cours:

Application
Transport
Réseau
Liaison de données
Support physique

Couche Liaison de données:

Liaison de données

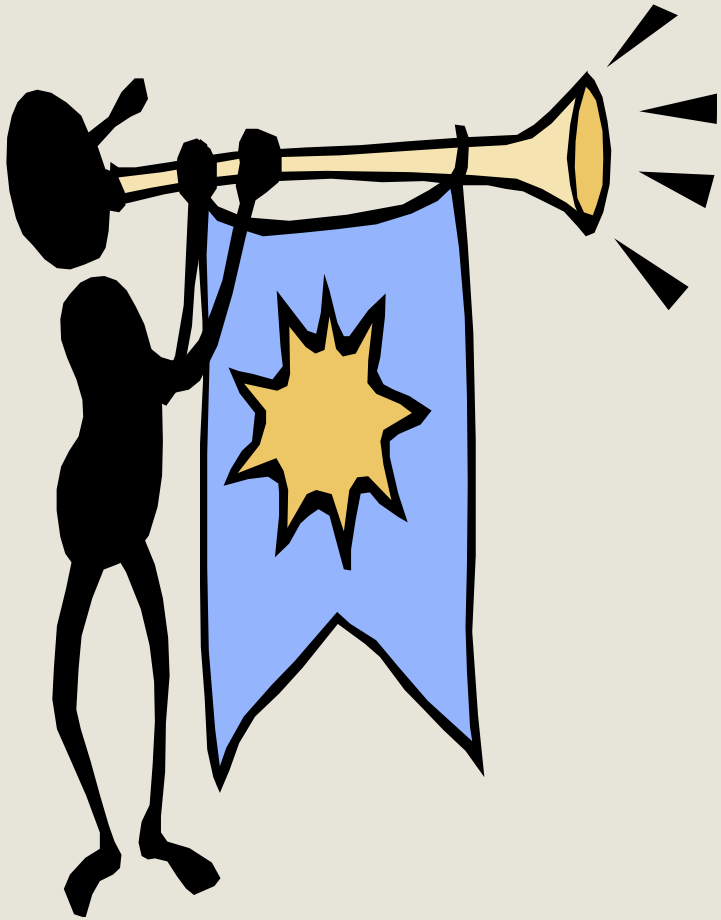
Contrôle et gestion de l accès au support physique:
Cas des réseaux a diffusion

Support physique

La couche liaison de données

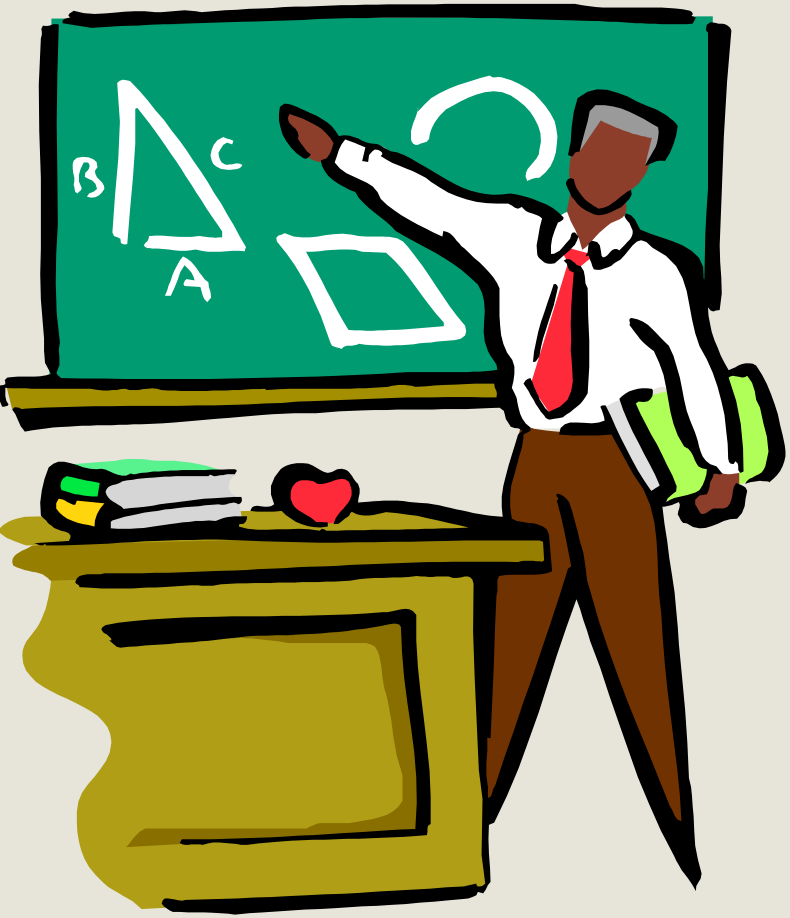
- Contrôle et gestion de l accès au support physique
- Au delà du contrôle et de la gestion de l accès au support physique
- Ethernet: de la couche physique à la couche liaison de données

Table des matières: Contrôle et gestion de l accès au support physique



- Le problème
- Les protocoles avec collision
- Les protocoles sans collision
- Les protocoles a contention limitée

Le problème



1. Contexte et modèle
2. Les questions

Contexte et terminologie ...

Réseaux a diffusion

- Cannal (support physique) pouvant être accédé par toutes les stations
  - Cannal a accès multiple (cannal aléatoire)
- Découpage des bits a envoyer sur le cannal physique
  - Trames (Frames)

Contexte et terminologie ...

- Modèle
  - N stations
  - Cannal aléatoire unique
  - Collision
    - Avec
    - Sans
  - Temps
    - Continu
    - Discrétisé
      - intervalles suffisantes pour transmission de trame (Time slot)
  - Écoute avant transmission
    - Avec
    - Sans

Les questions ...

Quelques exemples

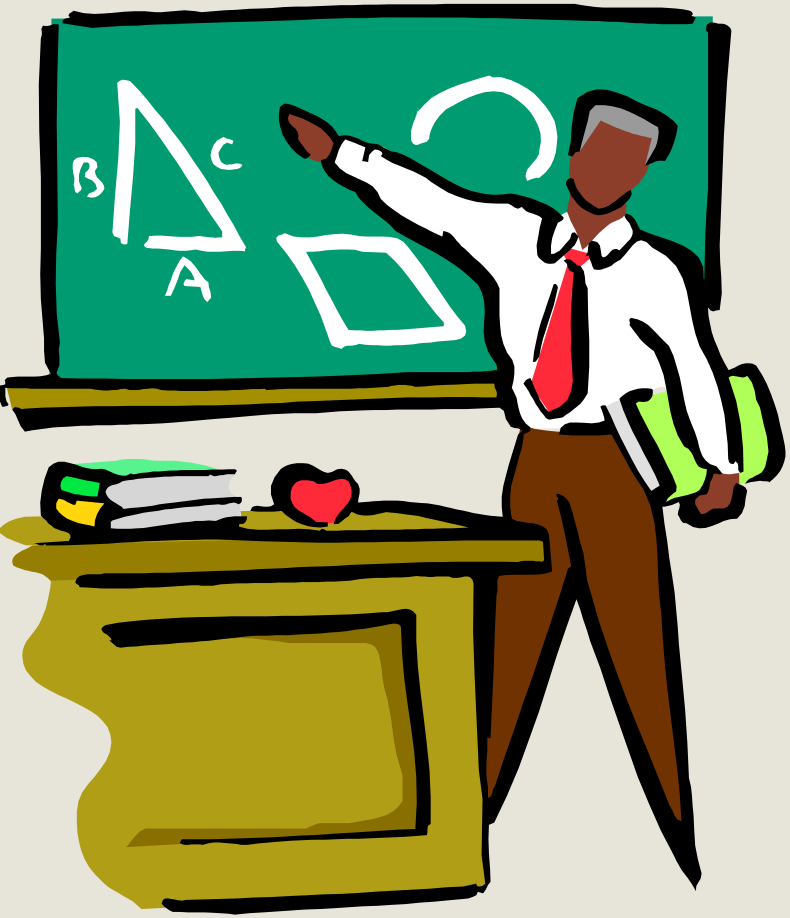
- Quand est ce qu une station peut accéder au cannal pour transmettre?
- Que fait la station quand le cannal est occupé?
- Comment savoir si la transmission a été couronnée de succès?
- Que fait la station si elle sait qu une autre station désire aussi transmettre?

Les réponses ...

Les protocoles de contrôle et d accès au support physique

- Généralement implémentés en hardware / firmware
  - Protocoles avec collision
    - Plus connues / déployées
  - Protocoles sans collision

Les protocoles avec collision …



1. ALOHA
2. Accès par écoute de porteuse
3. Le meilleur des deux mondes

ALOHA ...

Début des années 70

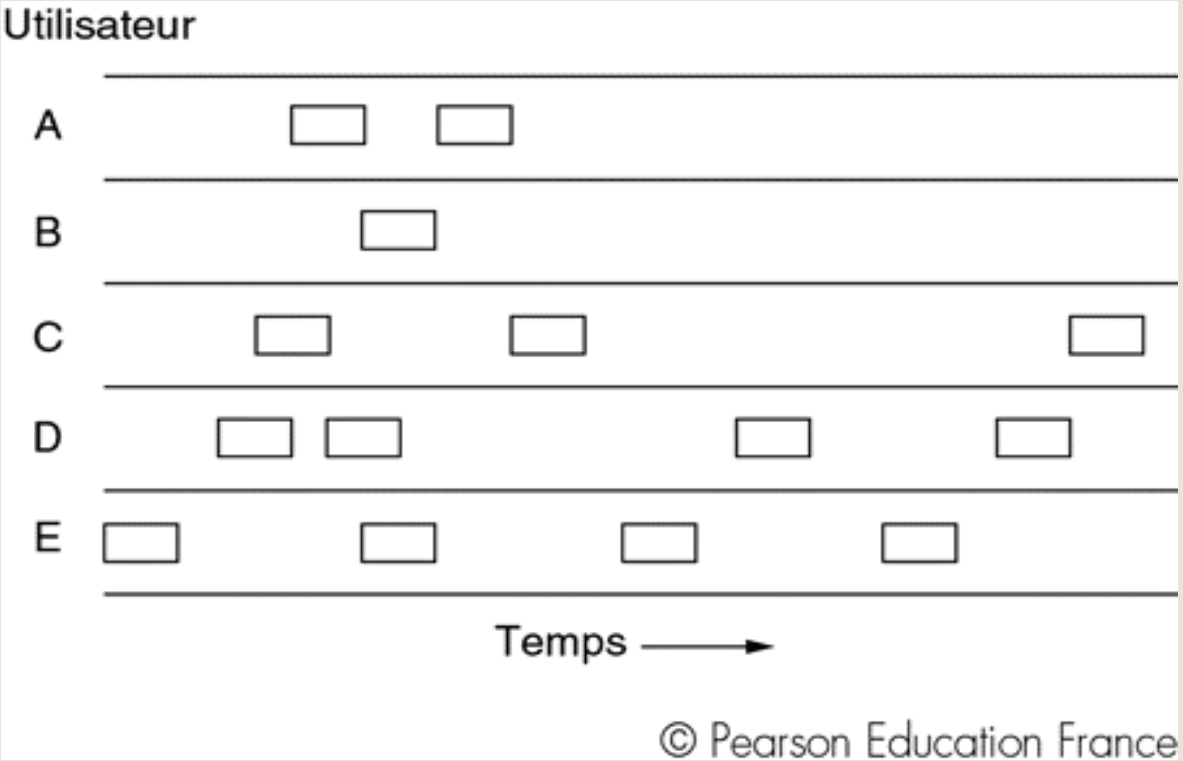
- Dévelopé dans un environement de radio terrestre pour connecter des ordinateurs a l université d Hawai
- Hypothèse principale
  - Les collisions sont détectées (Collision Detection – CD)
- Deus formes:
  - ALOHA pur
  - ALOHA discrétisé

ALOHA ...

- ALOHA pur
- Procedure dans chaque station
  - Station transmet dès que prête
  - S il y a collision
    - Station attend un temps aléatoire
    - Station re-transmet
- Evaluation intuitive
  - Période de haute charge
  - Période de basse charge

ALOHA ...

ALOHA pur

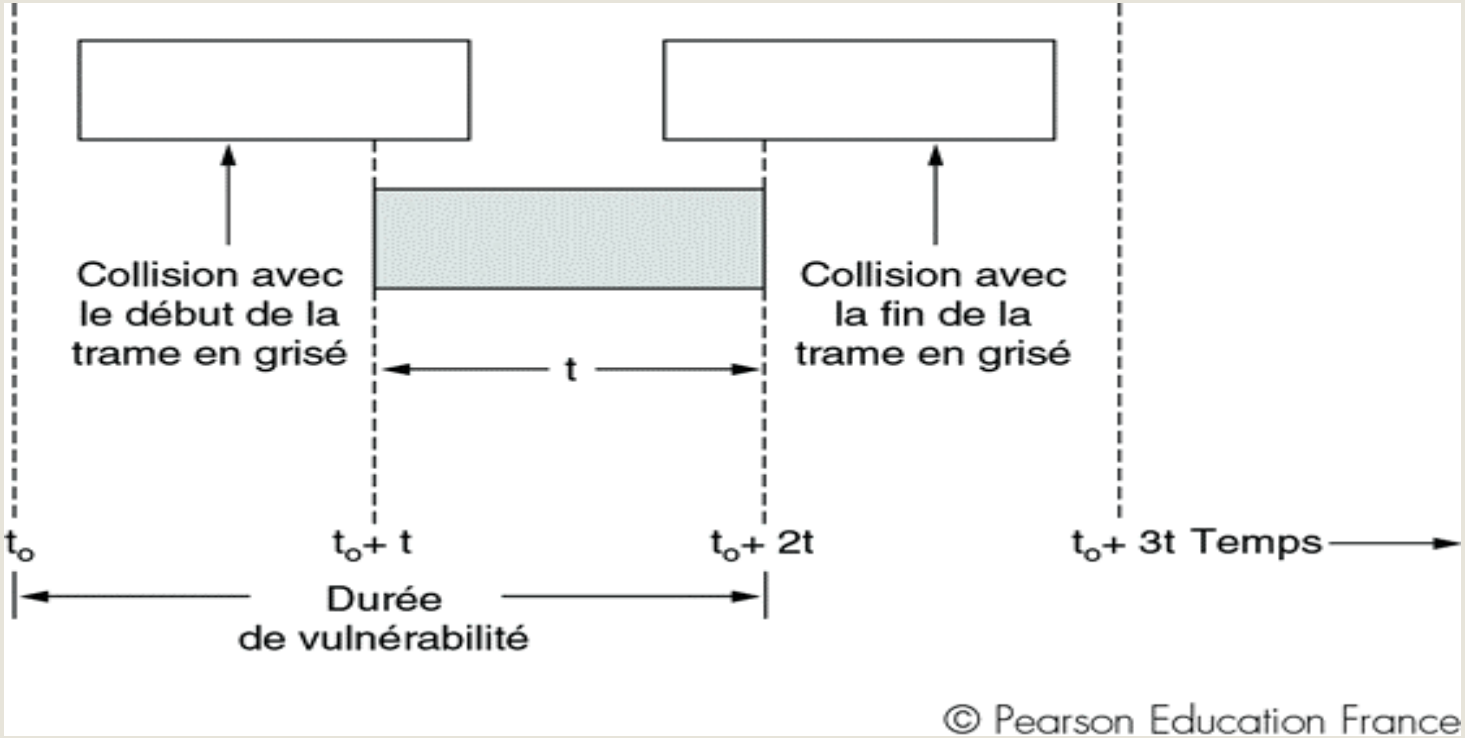


ALOHA ...

- ALOHA discrétisé
- Version amiliorée d ALOHA pur
  - Temps divisé en intervalles “slots” sur lesquelles toutes les stations s entendent
    - Slot: Temps nécessaire pour la transmission d une trame
  - Procédure
    - S il y a collision
      - Attendre le début du prochain slot
      - Re-transmettre
- Evaluation intuitive
  - Permet de réduire le nombre de collisions

ALOHA ...

ALOHA discrétisé



Accès par écoute de porteuse ...

- Collision Sensing Multiple Access (CSMA)
- Écoute et detection des actions (transmissions) sur le cannal
  - Une autre base d Ethernet
  - Amilioration importante par rapport a ALOHA
    - Éviter les collisions avec les stations qui sont déjà entrain de transmettre
      - Ne permet pas d éviter toutes les collisions
    - Plusieurs formes
      - Persistant
      - Non persistant
      - Avec détection de collision

Accès par écoute de porteuse ...

CSMA persistant

Procédure

1. Écoute
2. Transmet si cannal libre
3. S il y a collision retourne a 1.

Amilioration par rapport a ALOHA

Réduction du nombre de collisions

Accès par écoute de porteuse ...

CSMA non persistant

Procédure

1. Écoute
2. Transmet si cannal libre
3. S il y a collision

   Attendre temps aléatoire

   Retourner a 1.

Amilioration par rapport au CSMA persistant

Réduction du nombre de collisions

Le meilleur des 2 mondes ...

CSMA/CD

Carrier Sense Multiple Access with Collision Detection)

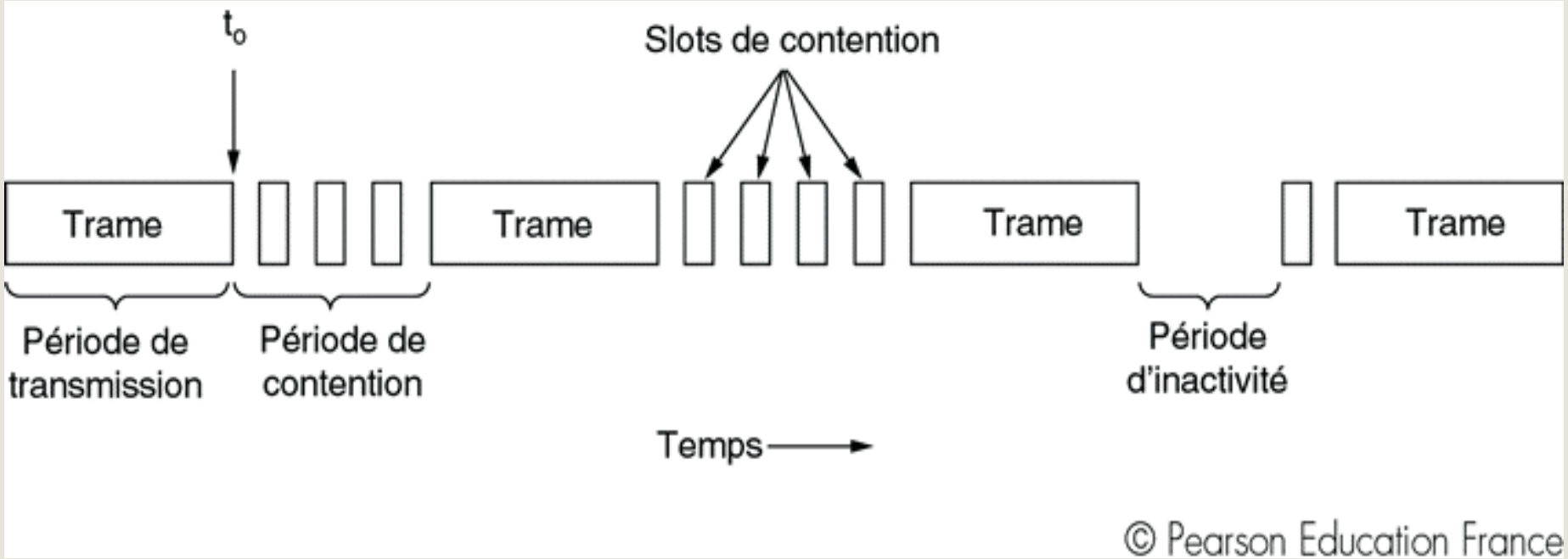
Acces par écoute de porteuse avec détection de collision

Base d Ethernet

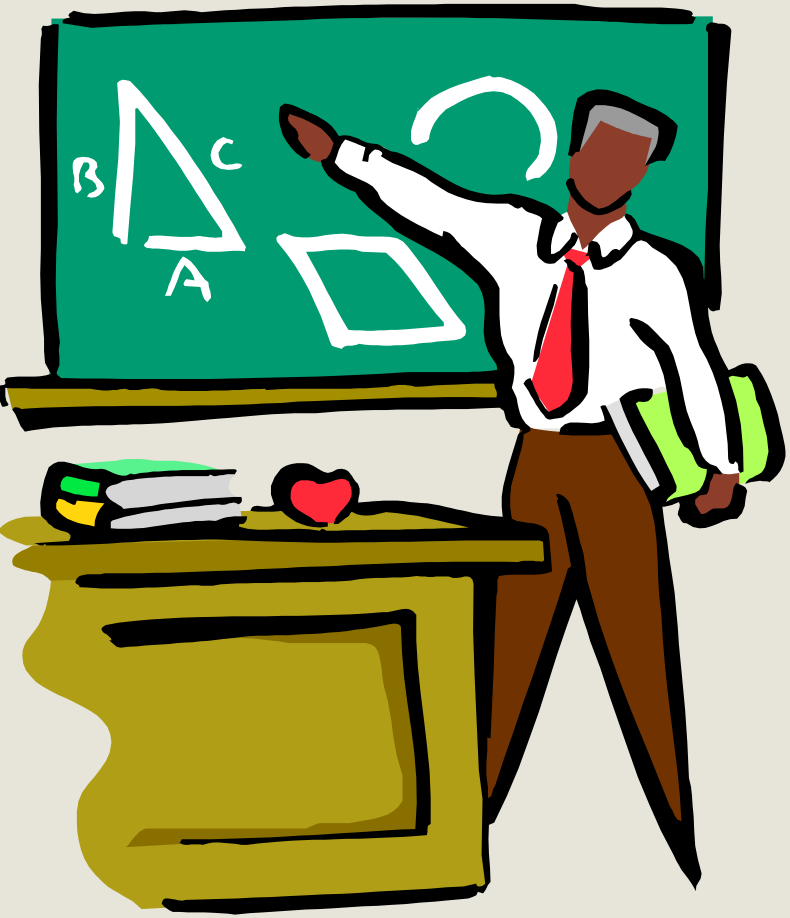
– Possibilité de détecter les collisions (avec une probabilité élevée) quand deux stations comment une transmission au même moment
  - Interruption
  - Délai aléatoire
  - Nouvelle tentative

Le meilleur des 2 mondes ...

CSMA/CD



Les protocoles sans collision …



1. Anneau a jeton (Token ring)
2. Protocole Bit-Map
3. Protocole a comptage binaire

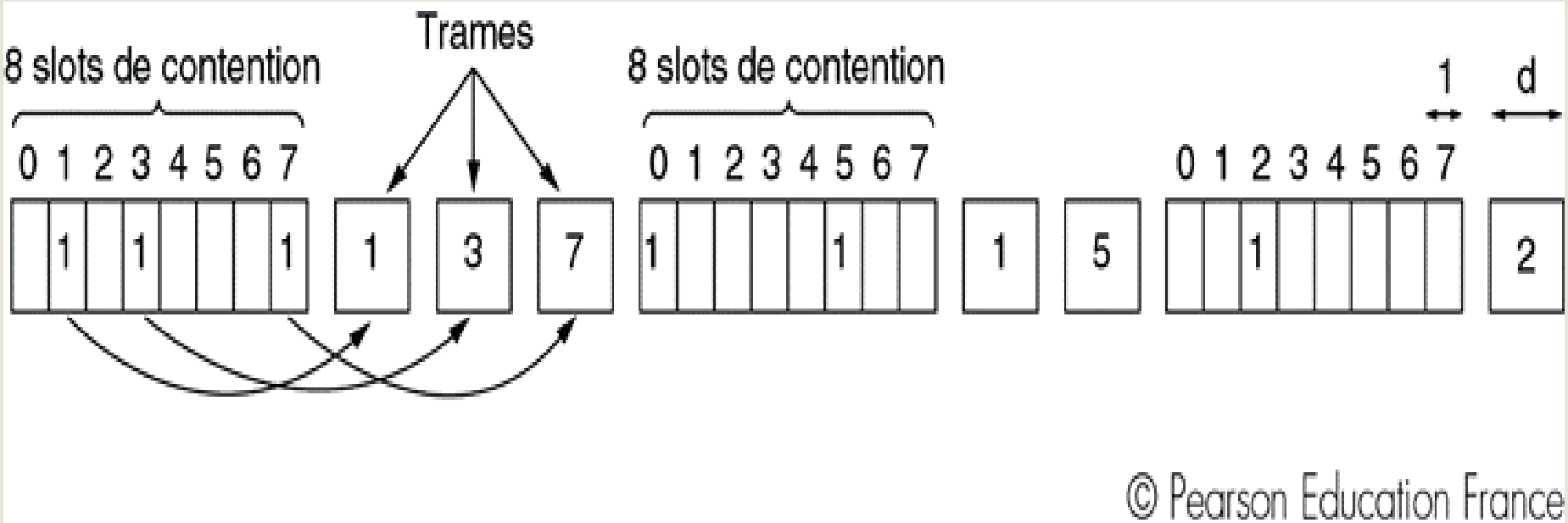
Anneau a jeton ...

- Dévelopé et poussé par IBM dans les années 70s
  - Peu deployé au tout début
  - A pratiquement disparu
  - Principles de base
  - Pas de collision: Une seule station transmet a la fois
  1. Jeton circule sur anneau et peut être pris par n importe quelle station
  2. Seule la station en possession du jeton a le droit de transmettre
  - Gestion très complexe
    - Perte de jeton
    - “Confiscation” de jeton par station

Protocole Bit-Map ...

- Deux phases
  - Première phase (Réservation)
    - Expression d interet pour transmission
      - Station j émet 1 durant le time slot j si elle désire transmettre
      - Et 0 si elle ne le désire pas
  - Deuxième phase (Transmission)
    - Après N slots
      - Tansmission selon l ordre des numérique des adresses
  - Problème
      - Pas adéquat pour les réseaux avec un nombre élevé de stations

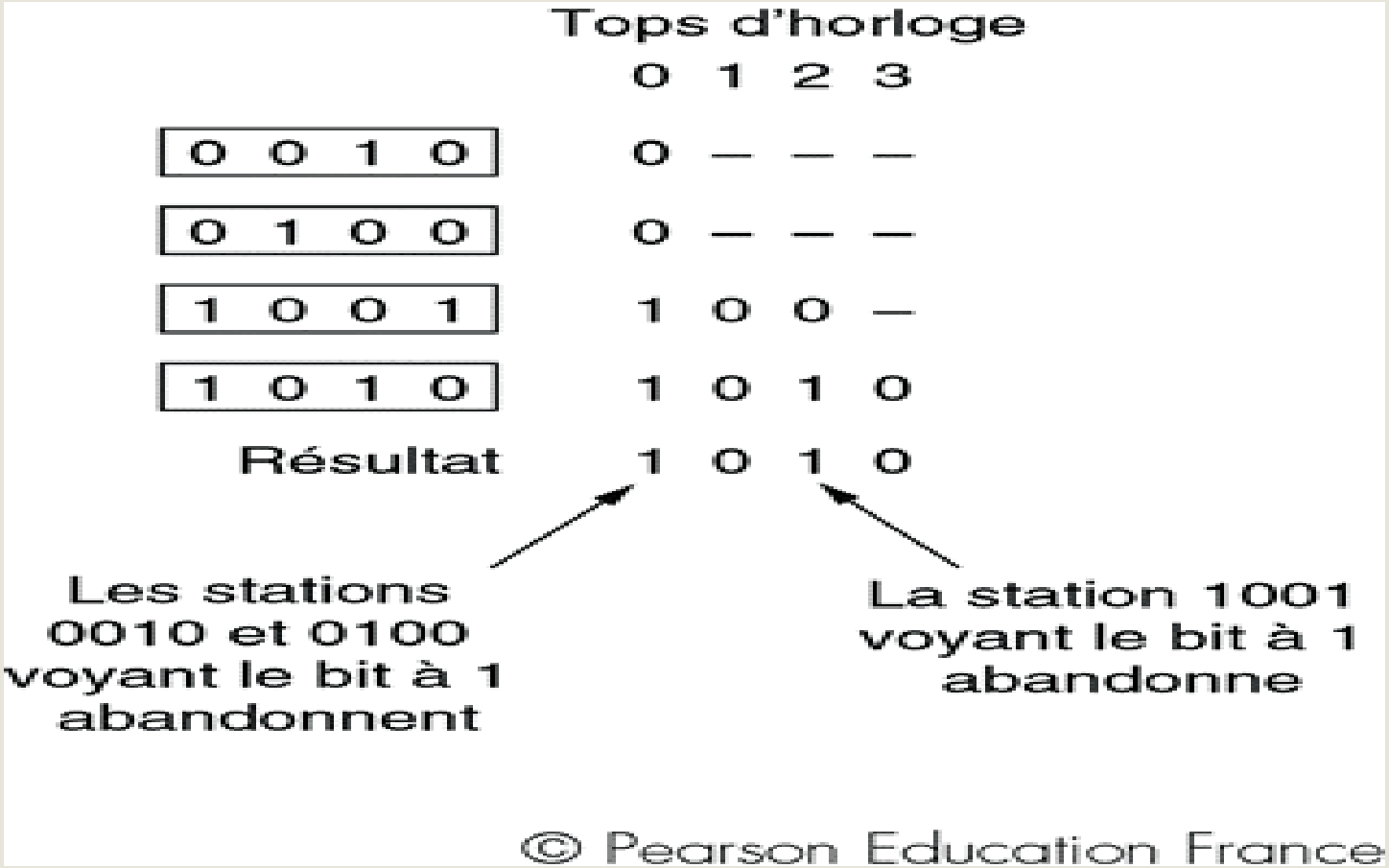
Protocole Bit-Map ...



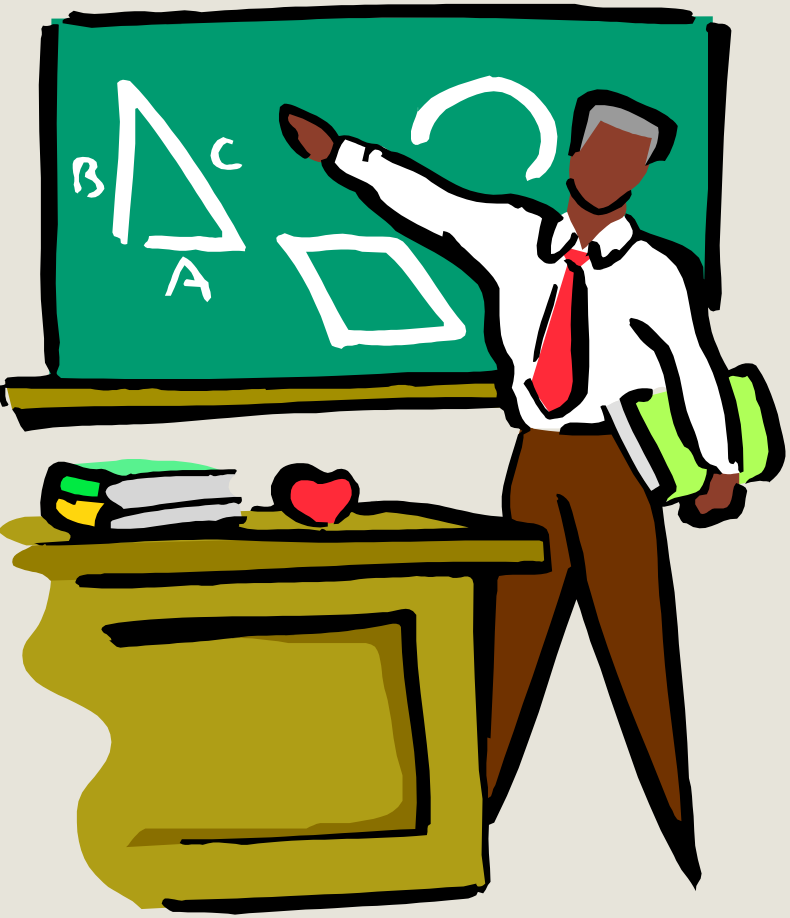
Protocole a comptage binaire ...

- Extension au protocole Bit-Map
  - Résolution du problème des réseaux avec nombre élevé de stations
    - Assignation d adresses binaires aux stations
    - Expression d interet par transmission bit par bit de l adresse
    - OU logique opérée par cannal
    - Station en concurrence avec station de plus haut rang se retire
      - Détectée quand l opération OU donne un 1

Protocole a comptage binaire ...



Les protocoles à contention limitée …



1. Principes généraux
2. Parcours arborescent adaptatif

Principes généraux ...

- Hybride entre protocoles avec collision et protocoles sans collision
  - Permettre les collisions
  - Reduire leur nombre en réservant des slots pour des groupes de station
    - Répartition des stations en groupe 1 à M
    - Seules les stations du groupe j peuvent transmettre durant le slot j
  - Cas extrêmes
    - 1 station par groupe
    - Toutes les stations dans un seul groupe

Parcours arborescent adaptatif ...

- Analogie:
- Test de dépistage de maladie conduits par l armée américaine durant la deuxième guerre mondiale
  - Prélèvement sanguin chez N soldats
  - 1 seul test sur le mélange de sang
    - Si test positif, deux nouveaux tests
      - 1 sur chaque groupe de N/2 soldats
  - Et ainsi de suite jusqu a ce que les malades soient détectés
- Protocole a contention limitée
  - Même processus chaque fois qu il y a collision

Parcours arborescent adaptatif ...

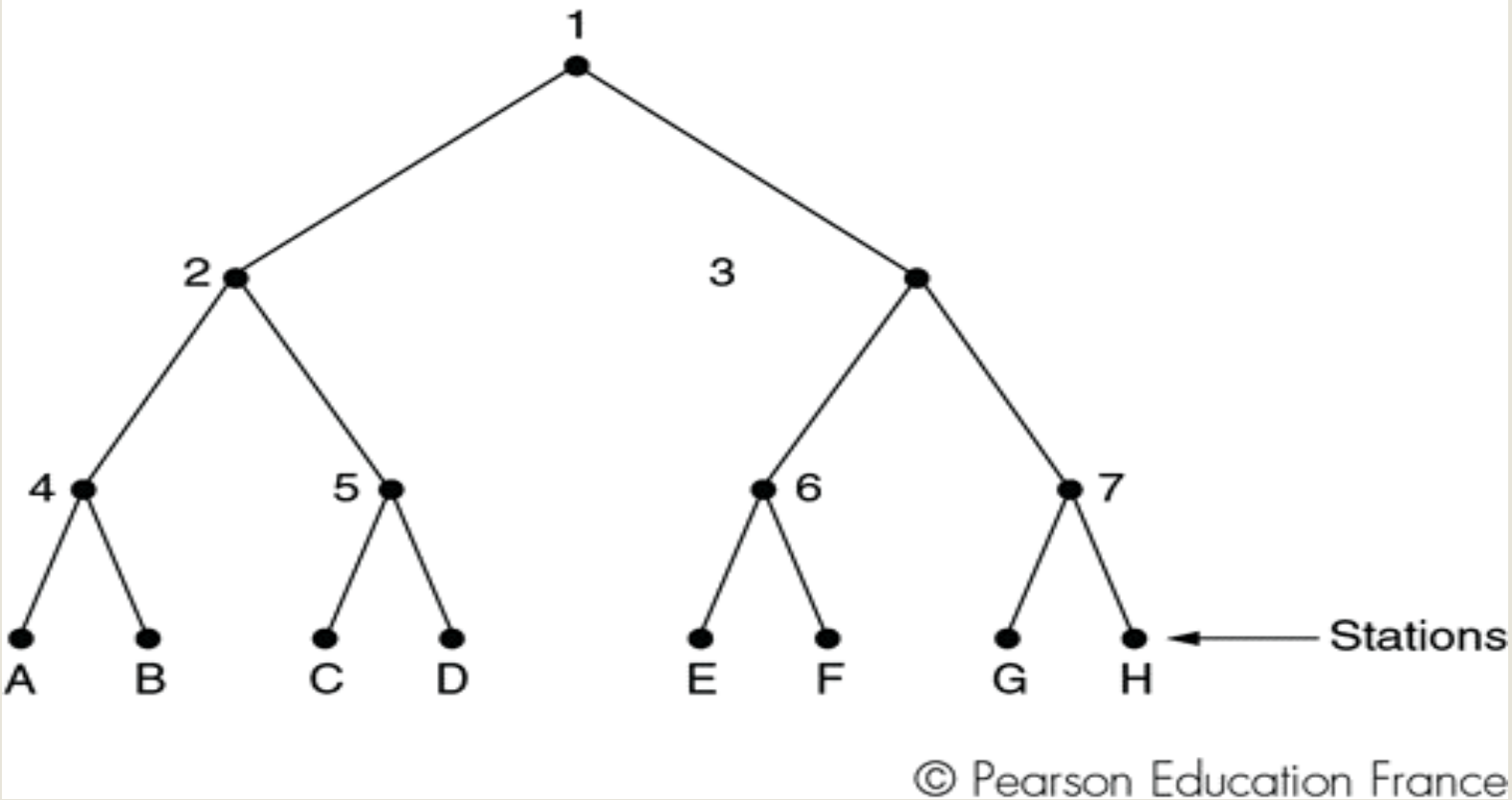
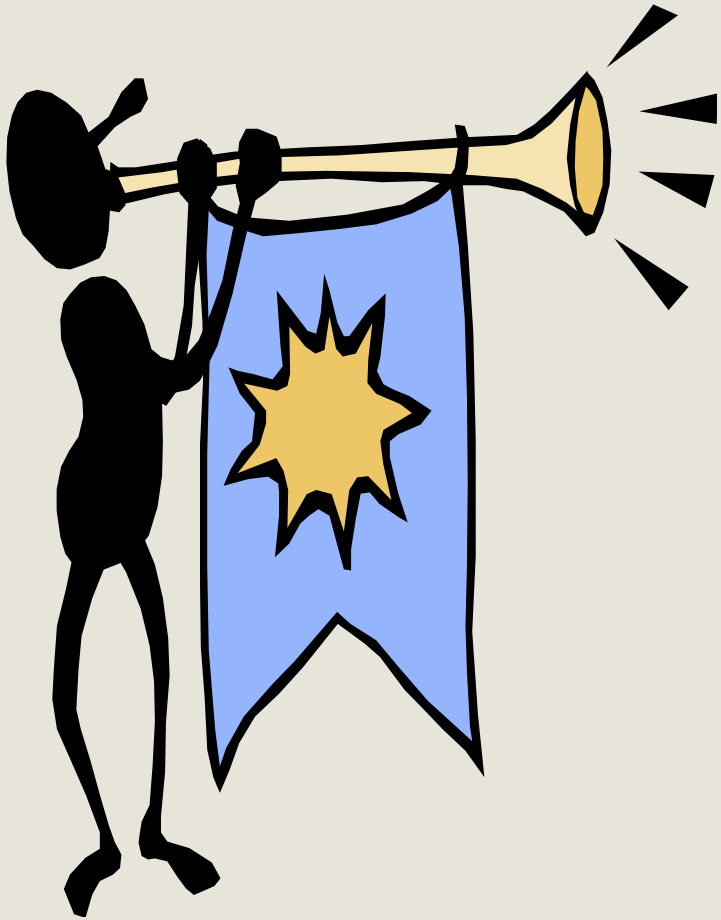
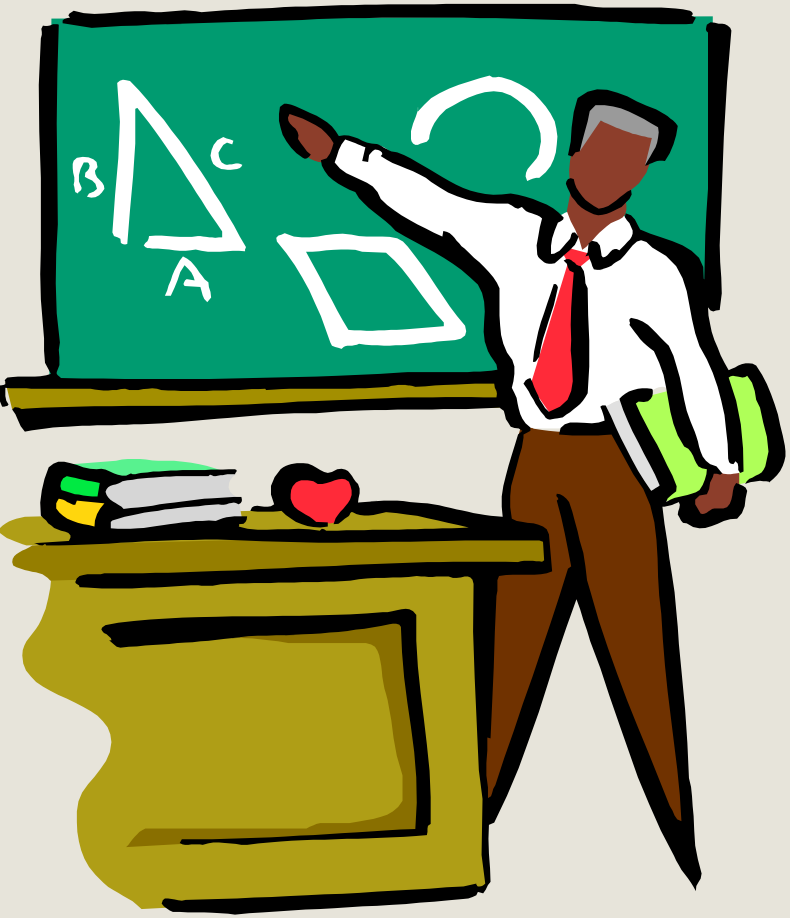


Table des matières: Au delà du contrôle et de la gestion de l accès au support physique



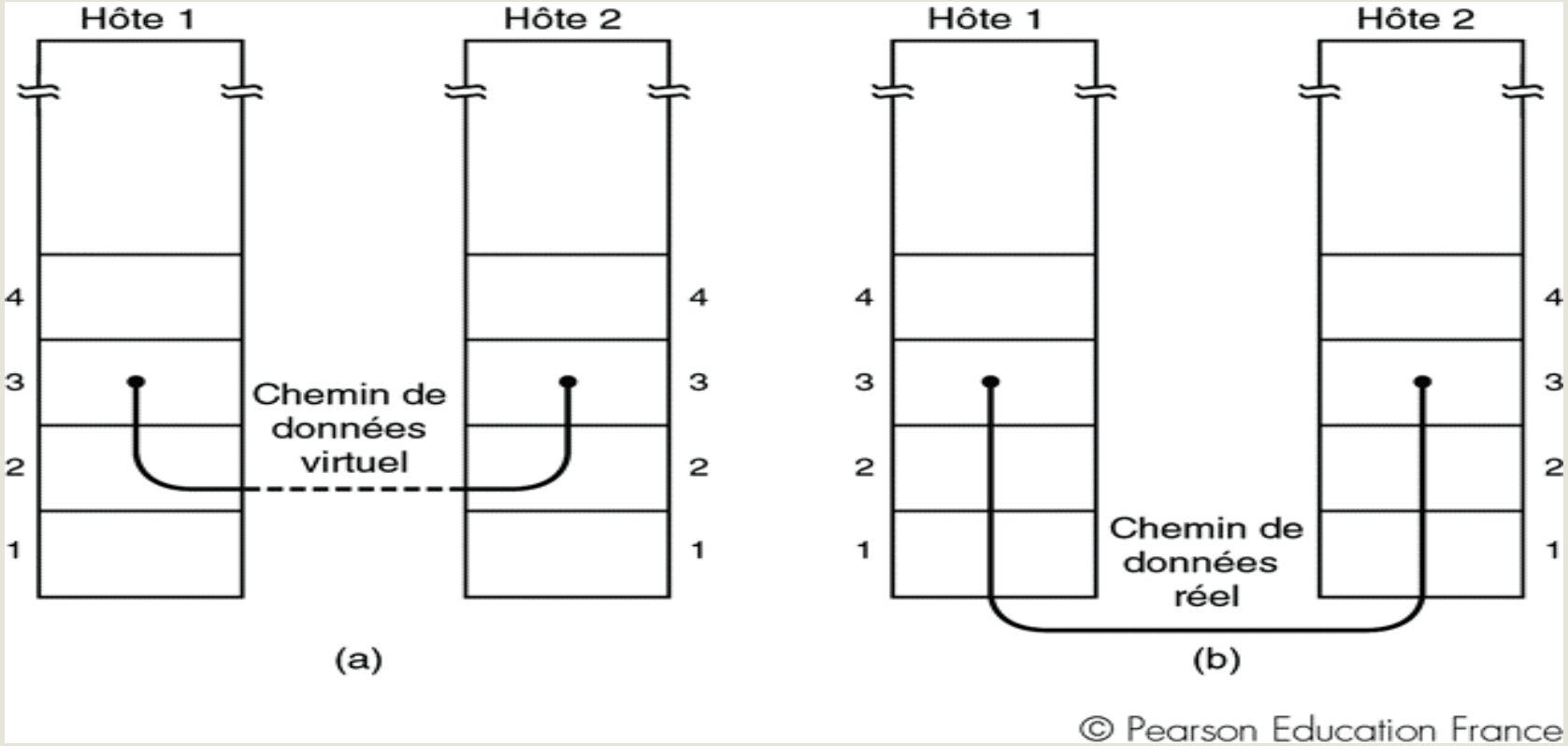
- Les fonctions
- Codes correcteurs et détecteurs d erreurs
- Protocoles élémentaires

Les fonctions …



1. Services fournis a la couche réseau
2. Trames
3. Contrôle d erreur
4. Contrôle de flux

Services fournis a la couche réseau ...



Services fournis a la couche réseau ...

Service 1: Sans connexion et sans accusé de réception

- Support physique avec faible taux d erreurs
- Correction prévue dans les couches supérieures
- Traffic temps réel (ex: voix)

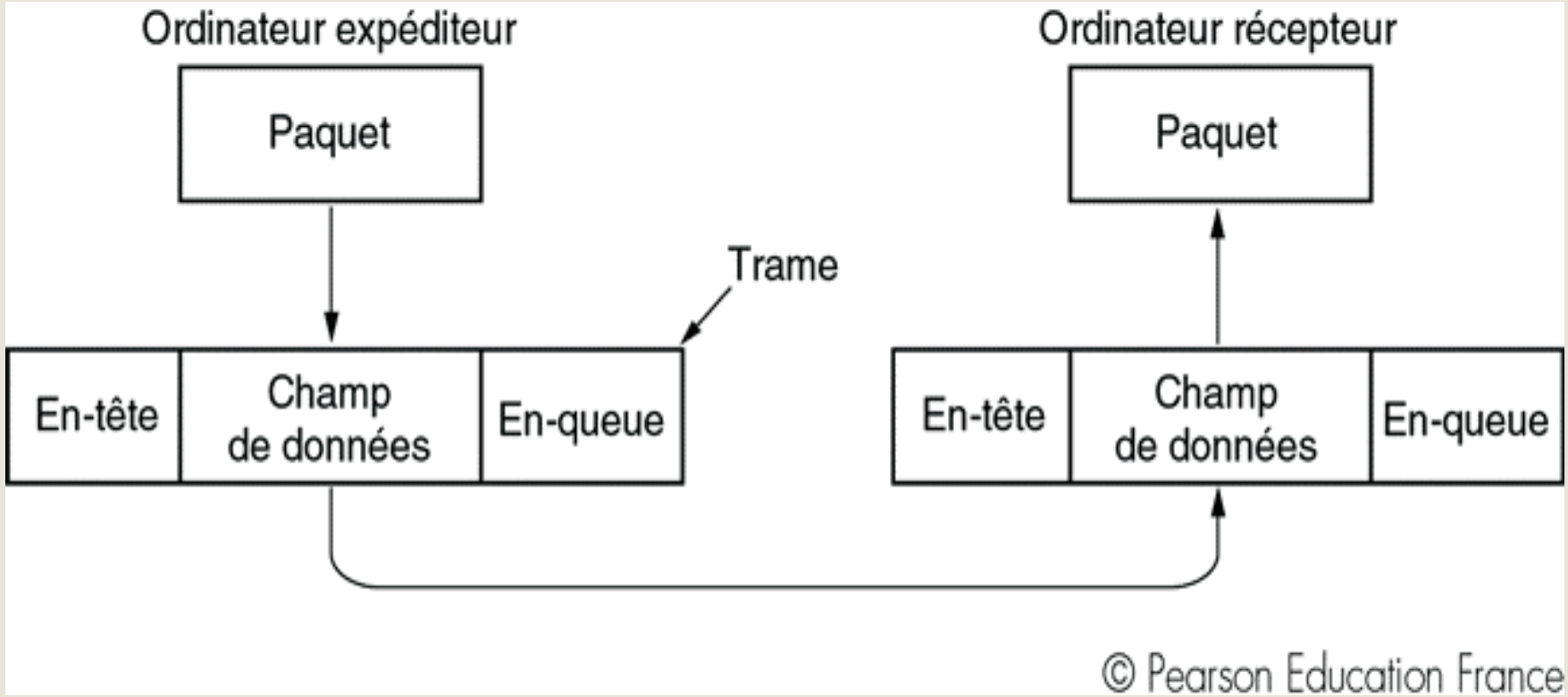
Service 2: Sans connexion et avec accusé de réception

- Plus fiable que le service 1
- Mais rallonge le temps de transmission des messages

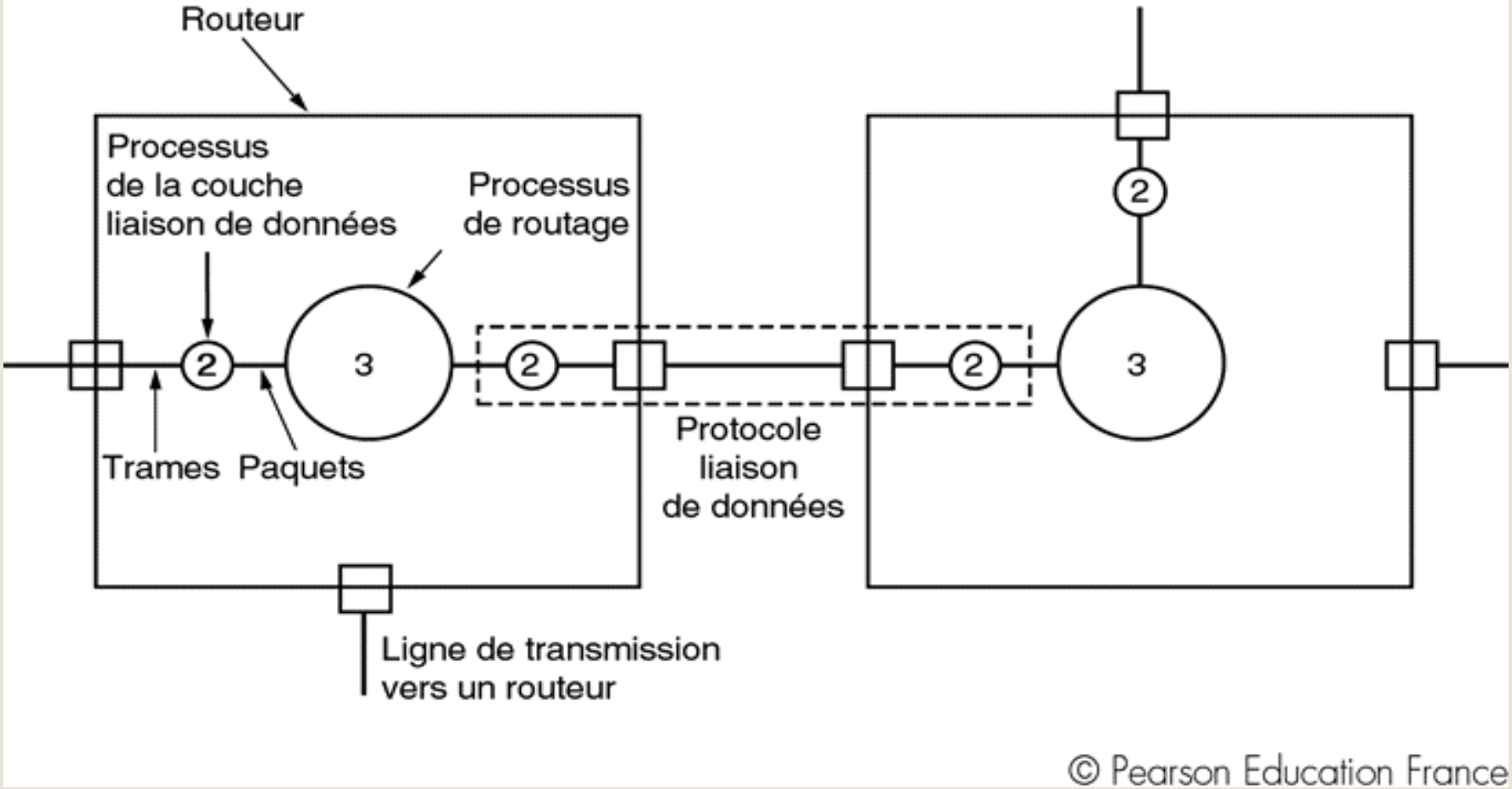
Service 3: Avec connexion et avec accusé de réception

- Plus fiable que le service 2
  - Établissement de la connexion
  - Transmission de données
  - Libération de la connexion

Trames ...


Ordinateur expéditeur
Paquet
Trame
En-tête
Champ de données
En-queue
Ordinateur récepteur
Paquet
En-tête
Champ de données
En-queue
© Pearson Education France

Trames ...


Routeur
Processus
de la couche
liaison de données
Processus
de routage
2
3
2
2
3
2
Trames
Paquets
Protocole
liaison
de données
Ligne de transmission
vers un routeur
© Pearson Education France

Trames ...

Délimitation des trames

1 Compter les caractères

- Problème:
  - Erreur sur le caractère indiquant le total

2 Fanion de signalisation (Flag) avec remplissage d octets

- Problème:
  - Profile binaire du fanion se retrouvant dans les données
- Solution:
  - Technique de remplissage d octets
    - Octet spécial d échapement avant chaque fanion accidentel
- Problème additionnel: Basée sur les octets

Trames ...

Délimitation des trames:

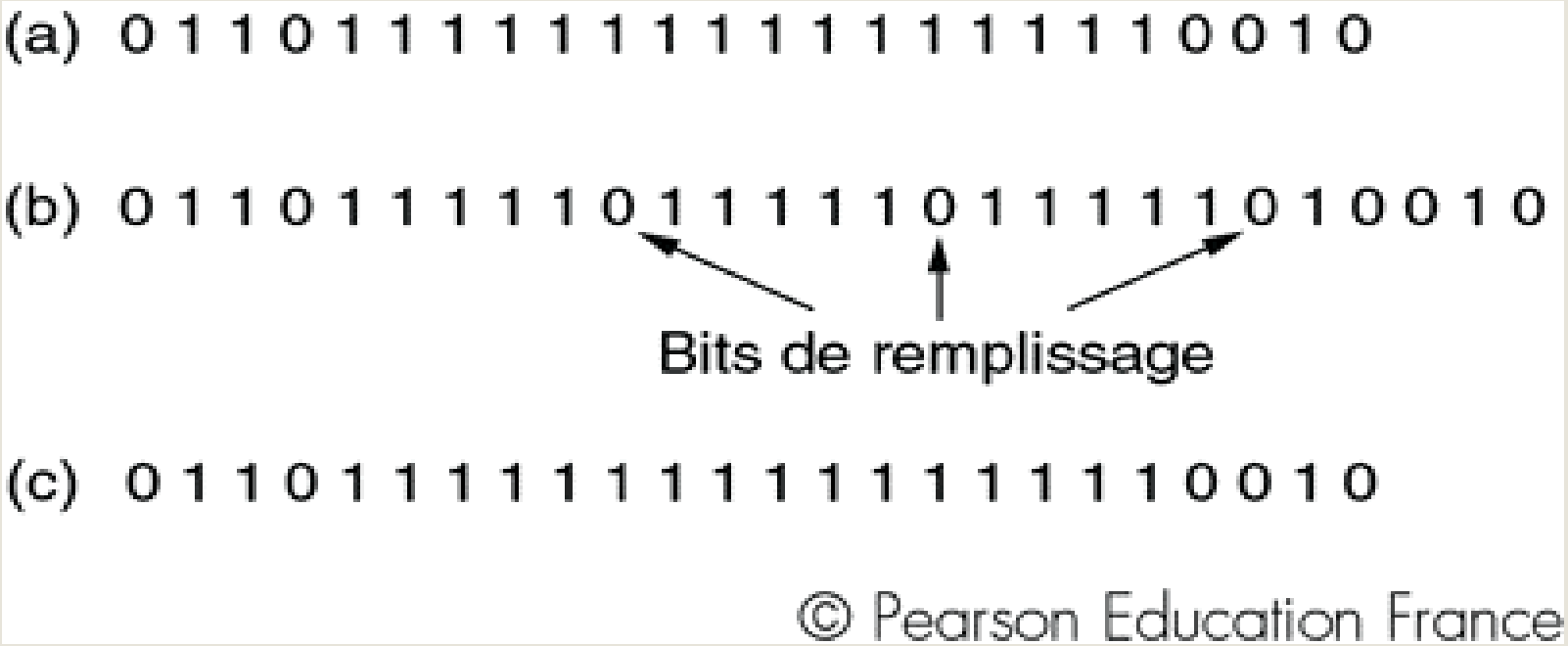
3 - Indicateur de début et de fin de trame avec remplissage de bits

- Début et fin de trame indiquée par configuration spéciale de bit
- Exemple:
  - 01111110
  - Émetteur évite d envoyer ce flux comme donnée en ajoutant 0 s il y a 5 bits consécutifs a 1.
  - Récepteur enlève automatiquement le 0 après chaque séquence de 5 bits

Trames ...

Délimitation des trames:

3 - Indicateur de début et de fin de trame avec remplissage de bits



Contrôle d erreur ...

Problème:

- Trame non recue
- Trame recue, mais avec erreur

Solutions

- Numérotation de paquets
- Timers (ou temporisateurs)

Contrôle de flux ...

Problème:

- Émetteur envoie plus que le récepteur ne peut traiter
- Trame recue, mais avec erreur

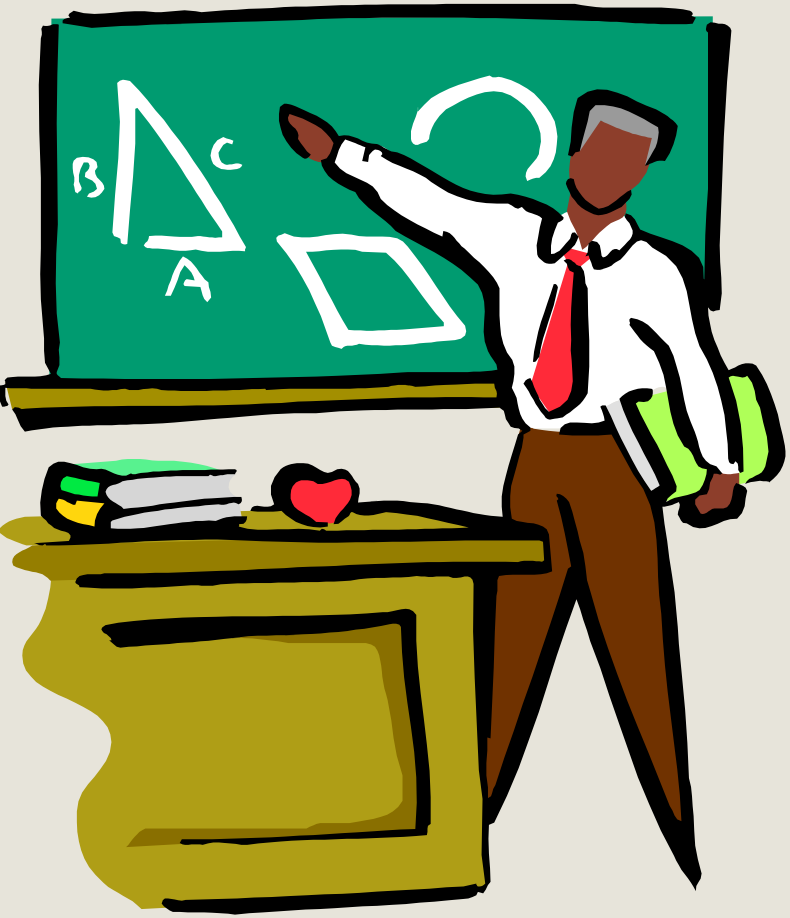
Solutions

- Contrôle de flux avec retour d informations
  - Feeback based flow control

    Pas de trames avant l autorisation explicite du récepteur

    Nombre de trames a envoyer décidée par le recepteur

Codes correcteurs et détecteurs …



1. Codes correcteurs
2. Codes détecteurs

Codes correcteurs (Error Correcting Code)...

- Assez de redondance dans les données afin de reconstituer les données correctes quand il y a erreur
  - Adéquat pour les cannaux peu fiables
  - Concepts clés
    - Mot de code = bits de données + bits redondants
    - Différence entre mots de code
      - OU exclusif
      - Distance de Hamming = nombre de bits de différence
        - » Si la distance de Hamming entre deux mots est d il faut d erreurs simples pour transformer l un en l autre
        - » Pour détecter d erreur il faut que le code ait une distance de Hamming de d+1
        - » Pour corriger d erreurs il faut un code de distance 2d+1

Codes correcteurs (Error Correcting Code)...

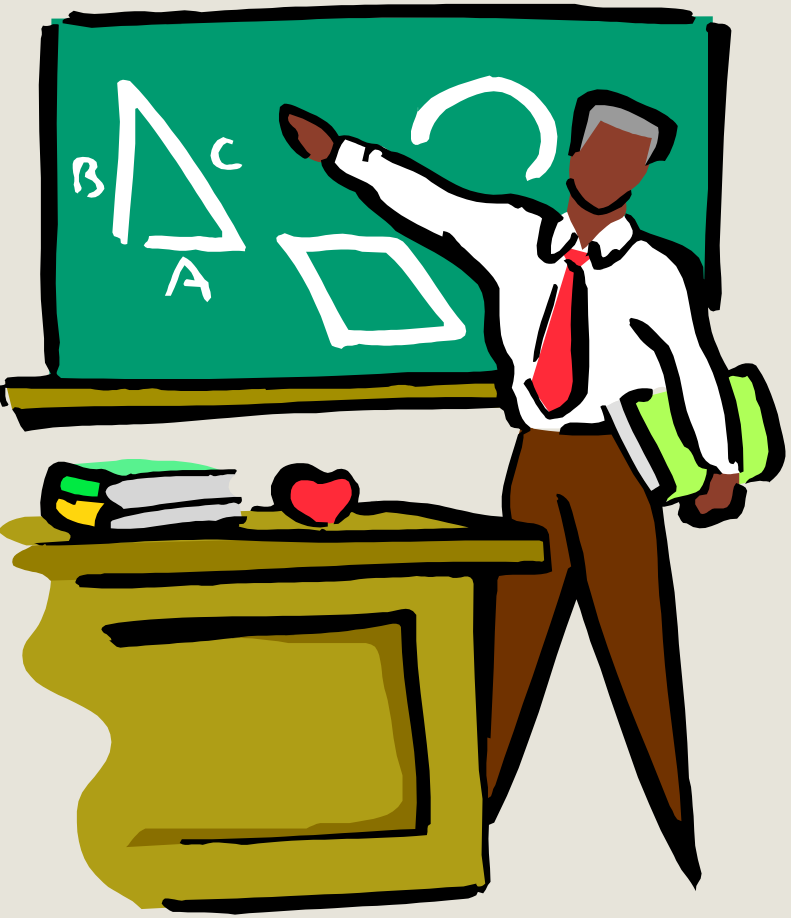
Exemples

- Bit de parité
  - Bit redondant choisi par émetteur tel que le nombre de bits a 1 soit pair (ou impair)
  - Erreur si mot impair (ou pair) recu par récepteur
  - Distance = 2
  - Détection d erreurs simples
- Code a 4 mots
  - Détection d erreurs doubles
  - 0000000000
  - 0000011111
  - 1111100000
  - 1111111111

Codes détecteurs (Error Detecting Code)...

- Assez de redondance dans les données afin de détecter les erreurs
  - Adéquat pour les cannaux fiables
  - Méthode la plus utilisée:
    - Code polynomial ou contrôle de redondance cyclic (CRC)
      - Polynôme générateur G(x) sur lequel s entendent émetteur et récepteur
        - » Bit de poid le plus fort et le plus faible sont egaux a 1
      - Bits redondants sont ajoutés par émetteur afin que le polynôme M(x) soit divisible par G(x)
      - Division faite par récepteur

Protocoles élémentaires …



1. Considérations générales
2. Protocole simplex
3. Protocole a fenêtre d anticipation

Considérations générales...

Hypothèses

- Couche physique, liaision de données et reseaux
  - Processus indépendants
  - Communications par échange de messages

- Communication entre machines A et B
  - Service fiable orienté connexion
  - Pas de panne de machine

Considérations générales...

Quelques définitions communes aux protocoles qui suivent.

```
#define MAX_PKT 1024                    /* taille du paquet
   en octets */
typedef enum {false, true} boolean ;    /* type booléen */
typedef unsigned int num_seq ;          /* numéros de séquences
   ou d'acquittement. */
typedef struct {unsigned char data[MAX_PKT] ;} paquet ;   /*
   définition du paquet */
typedef enum {data, ack, nak} type_trame ;                /*
   définition type_trame */
typedef struct {          /* les trames sont transportées dans
   cette couche */
   type_trame type ;      /* quel type de trame est-ce ? */
   num_séquence séq ;     /* numéro en séquence */
   num_séquence ack ;     /* numéro de l'accusé de réception
   */
   paquet info ;          /* paquet de la couche réseau */
} trame ;
```

Considérations générales...

Quelques définitions communes aux protocoles qui suivent.

```
/* Attente d'un événement ; renvoi de son type. */
void attendre_événement (type_événement *événement);
/* Chargement d'un paquet depuis la couche réseau pour transmission sur
   le canal */
void origine_couche_réseau (paquet *p);
/* Remise de l'information d'une trame entrante à la couche réseau */
void vers_couche_réseau (paquet *p);
/* Prise en compte d'une trame arrivant de la couche physique et copie
   dans r*/
void origine_couche_physique (trame *r);
/* Passage d'une trame à la couche physique pour transmission */
void vers_couche_physique (trame *s);
/* Armement du timer et activation de l'événement timeout*/
void start_timer (num_séquence k);
/* Arrêt du timer et désactivation de timeout */
void stop_timer (num_séquence k);
```

Considérations générales...

Quelques définitions communes aux protocoles qui suivent.

```
/* Armement d'un timer auxiliaire et activation de ack_timeout. */
void start_ack_timer (void);
/* Arrêt du timer auxiliaire et désactivation de ack_timeout */
void stop_ack_timer (void);
/* Autorisation donnée à la couche réseau de créer un événement
   couche_réseau_prête */
void activer_couche_réseau (void);
/* Interdiction à la couche réseau de créer un événement
   couche_réseau_prête */
void désactiver_couche_réseau (void);
/* L'expansion de la macro inc se fait en ligne : incrémentation
   circulaire de k */
#define inc (k) if (k < MAX_SEQ) k = k + 1; else k = 0
```

Protocole simplex non restreint ...

```
/* Le protocole 1 (utopique) ne permet la circulation
   de données que dans un seul sens, de l'émetteur vers
   le récepteur. On suppose que le canal de
   communication est parfait et que le récepteur est
   capable de recueillir toutes les données
   instantanément. L'émetteur, quant à lui, se contente
   d'émettre les données aussi vite qu'il le peut. */
```

Protocole simplex non restreint ...

```
typedef enum {arrivée_trame} type_événement;
#include "protocol.h"
void émetteur2 (void)
{
   trame s;                                  /* tampon trame en
   sortie */
   paquet tampon;                            /* tampon paquet en
   sortie */
   type_événement  événement;                /* arrivée_trame est
   la seule possibilité */
   while (true) {
      origine_couche_réseau (&tampon);       /* prendre ce qu'il y
   a à envoyer */
      s.info = tampon;                       /* le copier dans s
   pour transmission */
      vers_couche_physique (&s);             /* envoi de la trame
   */
      attendre_événement (&événement);       /* attendre le signal
   */
   }
}
```

Protocole simplex non restreint ...

```
void récepteur2 (void)
{
   trame r, s;                                  /* tampons pour
   trames */
   type_événement  événement;                   /* arrivée_trame est
   la seule possibilité */
   while (true) {
      attendre_événement (&événement);          /* seule possibilité
   arrivée_trame */
      origine_couche_physique (&r);             /* acquisition de la
   trame */
      vers_couche_réseau (&r.info);             /* passage du paquet
   à la couche réseau */
      vers_couche_physique (&s);                /* envoi d'une trame
   factice pour réveiller l'émetteur */
   }
}
```

Protocole simplex de type arrêt et attente ...

```
/* Le protocole 2 (protocole de type arrêt et attente)
   ne permet lui aussi la circulation des données que
   dans un seul sens, de l'émetteur vers le récepteur.
   Le canal de communication est supposé parfait, comme
   dans le protocole 1. Ici, toutefois, le récepteur a
   une capacité finie de mémoire tampon et une vitesse
   de traitement également finie ; le protocole doit
   donc explicitement empêcher l'émetteur de noyer le
   récepteur sous un flux de données arrivant plus vite
   que le récepteur ne peut les traiter. */
```

Protocole simplex de type arrêt et attente ...

```
void émetteur2 (void)
{
   trame s;                                  /* tampon trame en
   sortie */
   paquet tampon;                            /* tampon paquet en
   sortie */
   type_événement  événement;                /* arrivée_trame est
   la seule possibilité */
   while (true) {
      origine_couche_réseau (&tampon);       /* prendre ce qu'il y
   a à envoyer */
      s.info = tampon;                       /* le copier dans s
   pour transmission */
      vers_couche_physique (&s);             /* envoi de la trame
   */
      attendre_événement (&événement);       /* attendre le signal
   */
   }
```

Protocole simplex de type arrêt et attente ...

```
void récepteur2 (void)
{
   trame r, s;                                  /* tampons pour
   trames */
   type_événement  événement;                   /* arrivée_trame est
   la seule possibilité */
   while (true) {
      attendre_événement (&événement);          /* seule possibilité
   arrivée_trame */
      origine_couche_physique (&r);             /* acquisition de la
   trame */
      vers_couche_réseau (&r.info);             /* passage du paquet
   à la couche réseau */
      vers_couche_physique (&s);                /* envoi d'une trame
   factice pour réveiller l'émetteur */
   }
}
```

Protocole simplex pour un cannal bruité ...

```
/* Le protocole 3 (par) autorise le flux de données
   unidirectionnel sur un canal non fiable. */
#define MAX_SEQ 1   /* doit être égal à 1 pour le
   protocole 3 */
typedef enum {arrivée_trame, cksum_err, timeout}
   event_type;
#include "protocol.h"
```

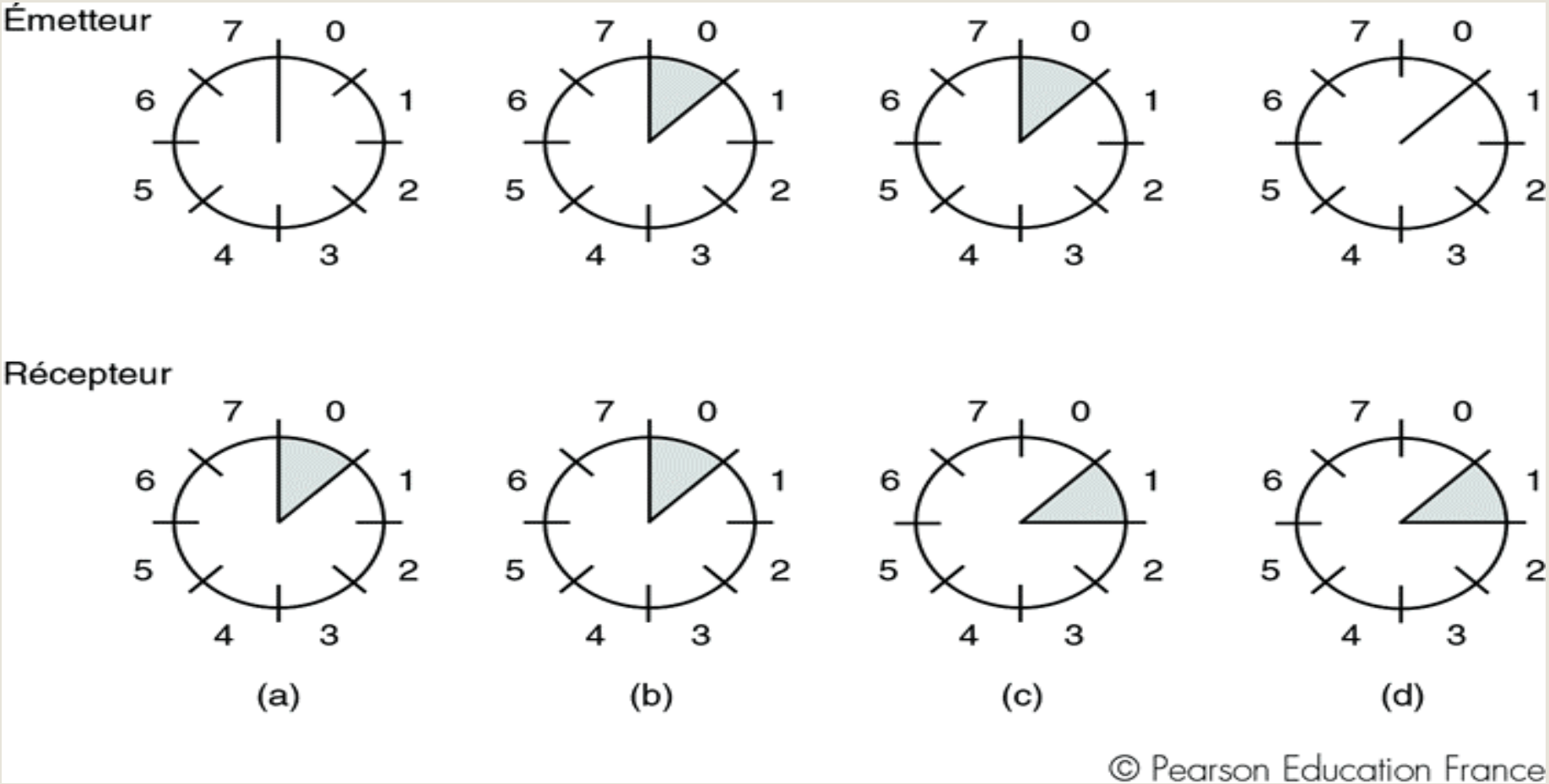
Protocole simplex pour un cannal bruité ...

```
{
    seq_nr prochaine_trame_à_envoyer;        /* numéro de
    séquence de la prochaine trame sortante */
    frame s;                                 /* variable
    scratch */
    packet buffer;                           /* buffer du
    paquet émis */
    event_type event;
    prochaine_trame_à_envoyer = 0;           /*
    initialise les numéros de séquences émises */
    origine_couche_réseau(&buffer);          /* récupère
    le premier paquet */
```

Protocole simplex pour un cannal bruité ...

```
while (true) {
      s.info = buffer;                          /* construit une trame à
   transmettre */
      s.seq = prochaine_trame_à_envoyer;        /* insère un numéro de
   séquence dans la trame*/
      vers_couche_physique(&s);                 /* l'envoie */
      start_timer(s.seq);                       /* si la réponse est trop
   longue à venir, délai dépassé */
      attendre_événement(&event);               /* arrivée_trame, cksum_err,
   timeout */
      if (event == arrivée_trame) {
         origine_couche_physique(&s);            /* récupère l'acquittement
   */
         if (s.ack == prochaine_trame_à_envoyer) {
            stop_timer(s.ack);                    /* désactive le timer */
            origine_couche_réseau(&buffer);       /* récupère la prochaine à
   envoyer */
            inc(prochaine_trame_à_envoyer);       /* inverse
   prochaine_trame_à_envoyer */
         }
      }
   }
}
```

Protocole avec fenêtre d anticipation ...



Protocole avec fenêtre d anticipation ...

```
/* Le protocole 4 (à fenêtre) est bidirectionnel. */
#define MAX_SEQ 1   /* doit être égal à 1 dans le protocole 4 */
typedef enum {arrivée_trame, cksum_err, timeout} event_type;
#include "protocol.h"
```

Protocole avec fenêtre d anticipation ...

```
{
   seq_nr prochaine_trame_à_envoyer; /* uniquement 0 ou 1 */
   seq_nr trame_attendue;            /* uniquement 0 ou 1 */
   frame r, s;                       /* variables scratch */
   packet buffer;                    /* paquet en cous d'envoi */
   event_type event;
   prochaine_trame_à_envoyer = 0;    /* prochaine trame dans le flux
    d'émission */
   trame_attendue = 0;               /* prochaine trame attendue */
   origine_couche_réseau(&buffer);   /* récupère un paquet de la couche
    réseau */
   s.info = buffer;                          /* se prépare à envoyer
    la trame initiale */
   s.seq = prochaine_trame_à_envoyer;        /* insère le numéro de
    séquence dans la trame */
   s.ack = 1 - trame_attendue;               /* superposition de
    l'acquittement */
   vers_couche_physique(&s);                 /* transmission de la
    trame */
```

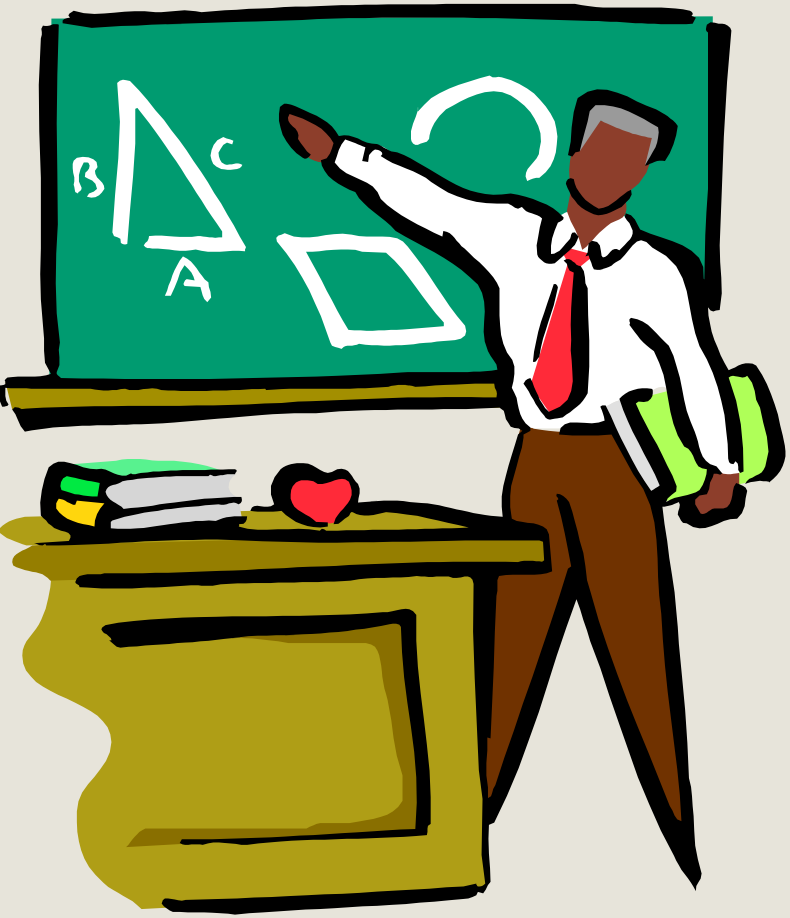
Protocole avec fenêtre d anticipation ...

```
start_timer(s.seq);                          /* démarre le timer */
   while (true) {
      attendre_événement(&event);            /* arrivée_trame,
    cksum_err ou timeout */
      if (event == arrivée_trame) {          /* une trame correcte est
    arrivée */
         origine_couche_physique(&r);        /* la récupère */
         if (r.seq == trame_attendue) {      /* gère le flux de trames
    entrantes */
            vers_couche_réseau(&r.info);     /* passe le paquet à la
    couche réseau */
            inc(trame_attendue);             /* inverse le numéro de
    séquence attendu après */
         }
         if (r.ack == prochaine_trame_à_envoyer){   /* gère le flux de
    trames sortantes */
            stop_timer(r.ack);                /* désactive le timer */
            origine_couche_réseau(&buffer);   /* récupère le nouveau
    paquet dans la couche réseau*/
            inc(prochaine_trame_à_envoyer);   /* inverse le numéro de
    séquence de l'émetteur*/
         }
```

Protocole avec fenêtre d anticipation ...

```
s.info = buffer;                          /* construit la trame émise */
    s.seq = prochaine_trame_à_envoyer;    /* y insère le numéro de
  séquence */
    s.ack = 1 - trame_attendue;           /* numéro de séquence de la
  dernière trame reçue */
    vers_couche_physique(&s);             /* transmet une trame */
    start_timer(s.seq);                   /* lance l'exécution du
  timer */
  }
}
```

Ethernet …

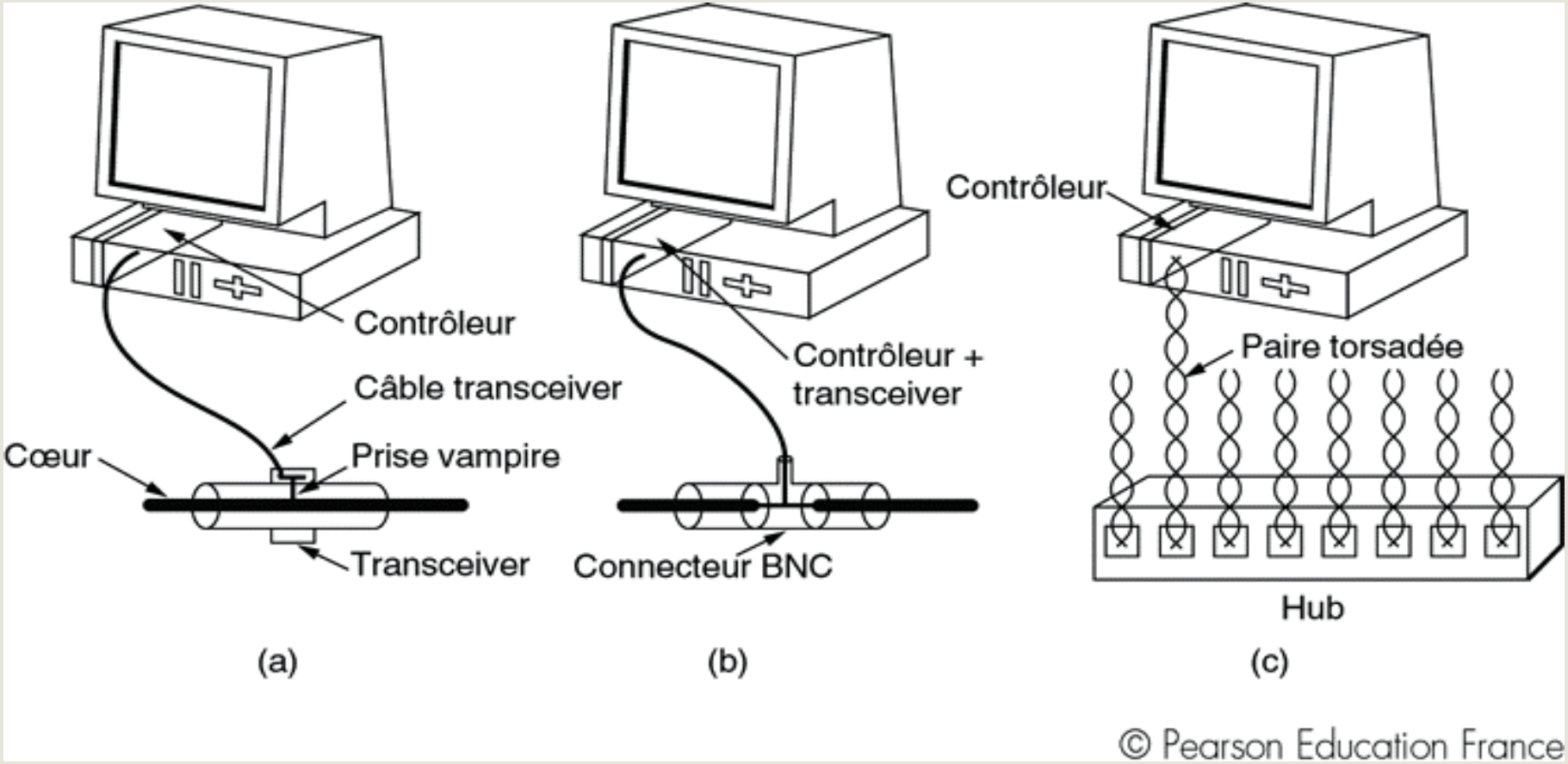


1. Couche physique
2. Couche liaison de données

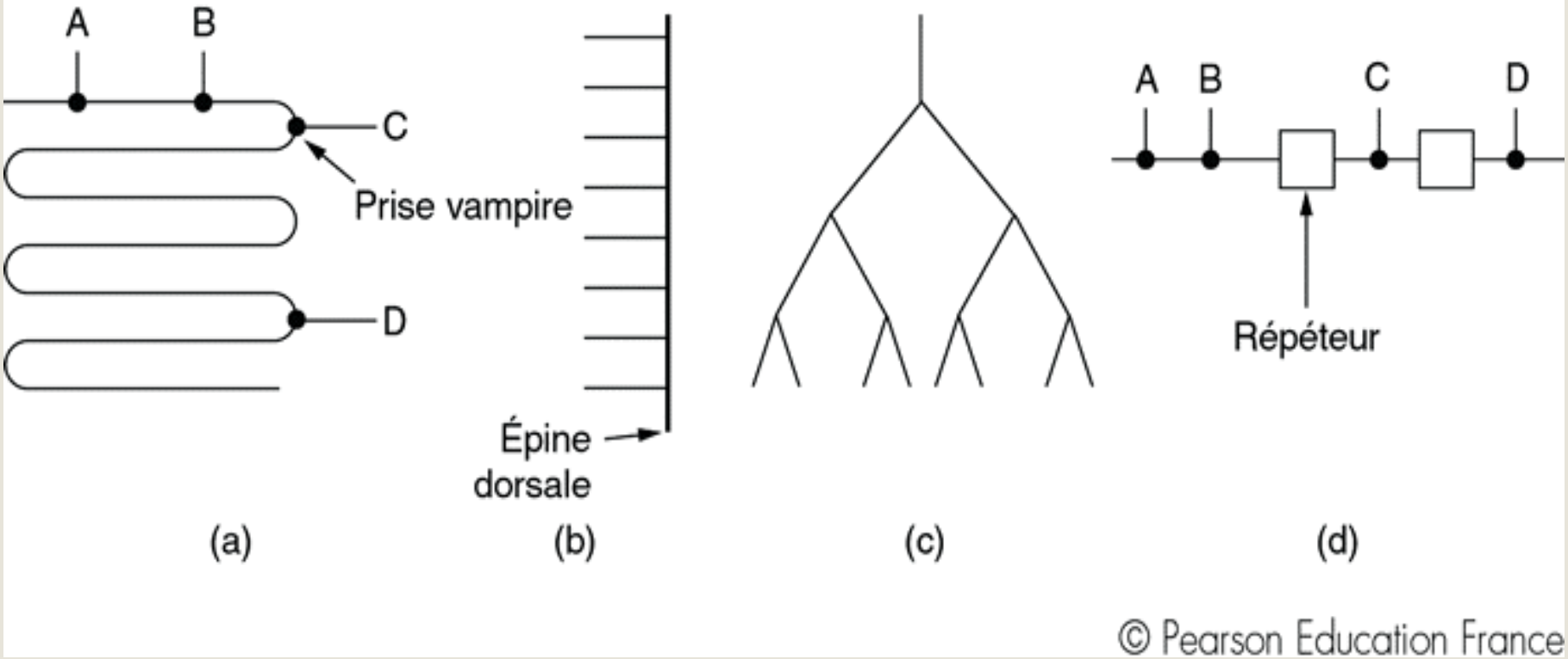
Couche physique

10Base5	Coaxial épais	500m	100 noeuds	Obsolète
10 Base-1	Coaxial fin	185m	39 noeuds	Pas de hub
10Base-T	Paire torsadée	100m	1024	Le – cher
10Base-F	Fibre optique	2000m	1024	Connexion d immeubles

Couche physique



Couche physique

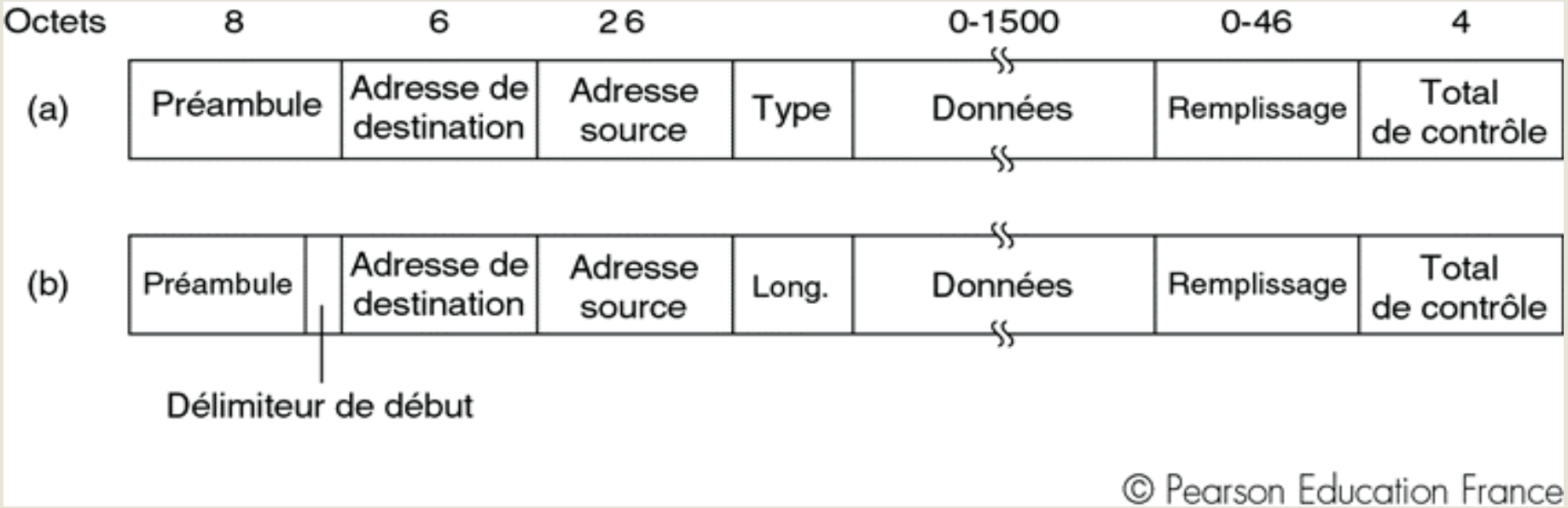


Couche liaison de données ...

- Medium Access Control (MAC): Protocole de gestion et de contrôle du cannal:
- CSMA/CD
- Trame
  - Trame originale (DIX)
  - Trame standardisée (IEEE 802.3)
  - Addressage (au niveau liaison de données)
    - Un seul noeud
    - Quelques noeuds (diffusion restreinte - multicast)
    - Tous les noeuds (diffusion générale – broadcast)

Couche liaison de données ...

MAC



Références ...

1. A. Tanembaum, Réseaux, 4eme édition, Pearson Education 2003 (Chapitres 3

.