Faculté des Sciences
Aix*Marseille Université

M1 Informatique
Réseaux

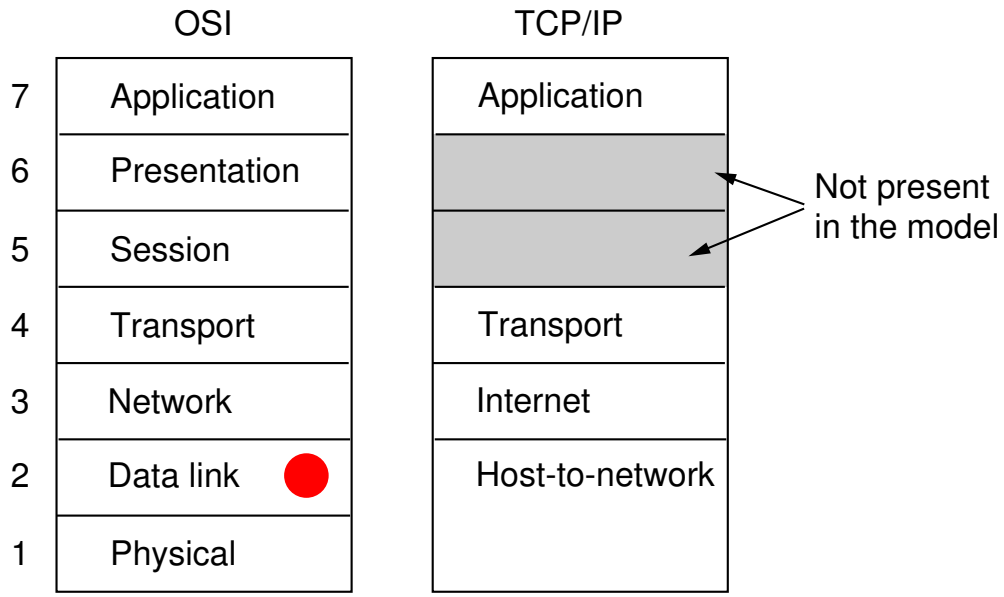
Cours 2 – Couche Liaison de Données

Notes de Cours

LA COUCHE LIAISON DE DONNEES A POUR OBJECTIF PRINCIPAL de proposer à la couche Réseau supérieure une abstraction pour la connexion locale.

S'appuyant sur la couche physique, elle doit donc, en outre, en gérer les conséquences (erreurs, pertes, ...).

1 "Vous êtes ici"



2 La Couche OSI Liaison de Données

2.a Objectifs de la Couche Liaison

En s'appuyant sur la couche physique, la couche Liaison de Données doit offrir une **connexion locale** à la couche Réseau

- connexion
  - un-vers-un (*unicast*)
  - un-vers-plusieurs (*multicast*)
  - un-vers-tous = diffusion (*broadcast*)
- fiable ou non
- utilisant un *espace de nom* ***local***

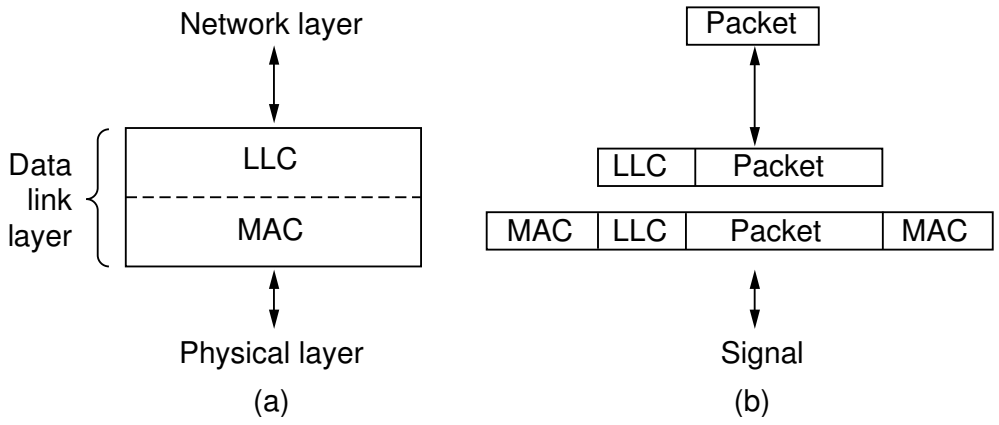
Comme cette couche s'appuie sur la couche physique, elle doit gérer les *conséquences* des imperfections de la couche physique.

2.b Exemples de Protocoles Couche 2

- Ethernet
- MPLS (Multiprotocol Label Switching)
- HDLC (High-Level Data Link Control)
- FDDI (Fiber Distributed Data Interface)
- PPP (Point-to-Point Protocol)
- ...

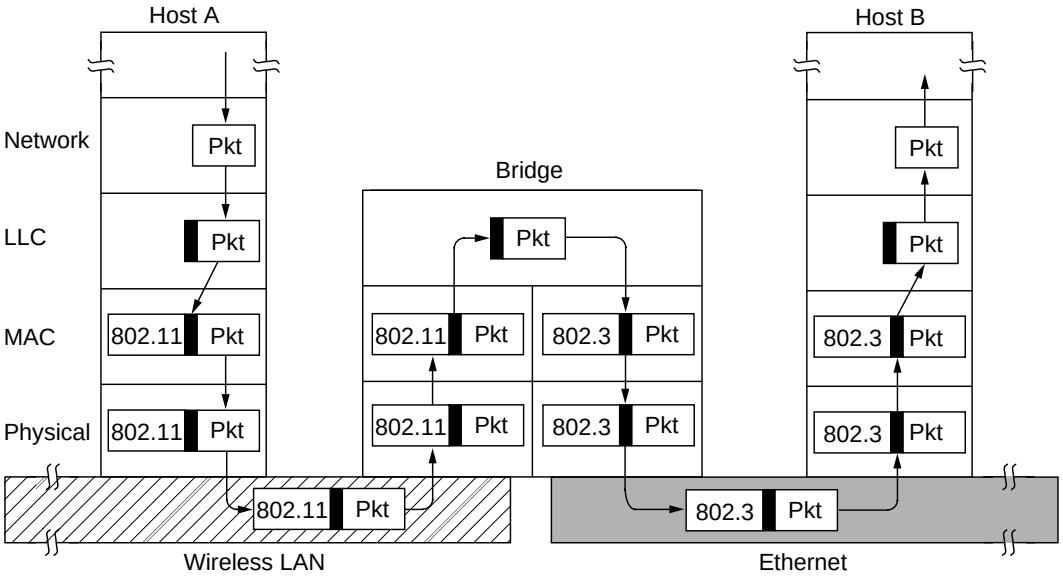
2.c Des Sous-Couches pour la Couche Liaison

- LLC : Contrôle Logique de la Liaison
- MAC : Contrôle d'Accès au Médium : des protocoles adaptés aux spécificités des couches physiques sous-jacentes



2.d Ponts

La subdivision en deux sous-couches, même si elle contrevient en un certain sens à la la norme OSI, permet de constituer des segments de Niveau 2 s'appuyant sur des segments de Niveau 1 de nature *physique* différente.



Un pont 802.11 vers 802.3

2.e Aparté : Matériel Réseau

Couche OSI	Matériel
Application	passerelle applicative
Transport	passerelle transport
Réseau	routeur
Liaison	commutateur, **pont**
Physique	concentrateur,répéteur

2.f Matériel Réseau : Définitions

répéteur réémet, amplifie un signal physique ;
concentrateur permet de raccorder différents segments dans un réseau en reproduisant le signal dans tous les segments ;
commutateur permet de raccorder plus efficacement différents segment en ne reproduisant le signal que dans le seul segment raccordé destinataire de la trame ;
pont raccorde en un même segment des segments de couche physique de nature différente ;
routeur appareil effectuant le routage (cf le cours "Routage").

2.g Types de Connexion

Service sans connexion et sans acquittement
- couche physique très fiable
- ou erreurs corrigées par les couches supérieures
- ou données supportant ces erreurs

Ex : LAN, flots temps réels, voix

Service sans connexion et avec acquittement
- émetteur sait si le message est arrivé
- réémission possible

Service avec connexion => ***service fiable*** – établissement de la connexion
- numérotation des messages
- chaque message est envoyé et reçu une seule fois
- l'ordre des messages est respecté

2.h Services Détaillés de la Couche Physique

- organisation des données (=> *trames*)
- Synchronisation
- services de la sous-couche LLC
- services de la sous-couche MAC

2.i Types d'Erreurs

1. Erreur de **modification** : la séquence de bits reçus est différente de celle émise.
2. Erreur d'**omission** : la séquence de bits n'est pas reçue
3. Erreur d'**addition** : une séquence de bits est reçus alors qu'aucune n'avait été émise. (également **duplication**)

2.j La Sous-Couche LLC

Rôle
1. contrôle des erreurs (*omissions*)
2. contrôle de flux

2.k La Sous-Couche d'Accès au Médium

Rôle
- Adressage physique (*adresse MAC*)
- Détection/Correction d'erreurs (*modification*)
- Adaptation au canal
  - gestion des *collisions*
  - taille maximale de trame
  - => optimisation de l'utilisation du canal

3 Mise en Trame

3.a Trames

On appelle *trames* les messages de la couche liaison de données.
La délimitation des trames n'est pas triviale
- Des "espaces" ne suffisent pas
- Longueurs fixes ou variables ?
  => *bourrage* éventuel
- Délimitations *explicites* :

- comptage de caractères
- caractères de début/fin (=> *transparence*)
- utilisation de séquences physiques non-codante.

3.b Transparence

Si l'on utilise un caractère (suite de bits) particulier pour indiquer la **fin d'une trame**, il ne faut pas que ce caractère apparaisse **à l'intérieur** des données encapsulées.

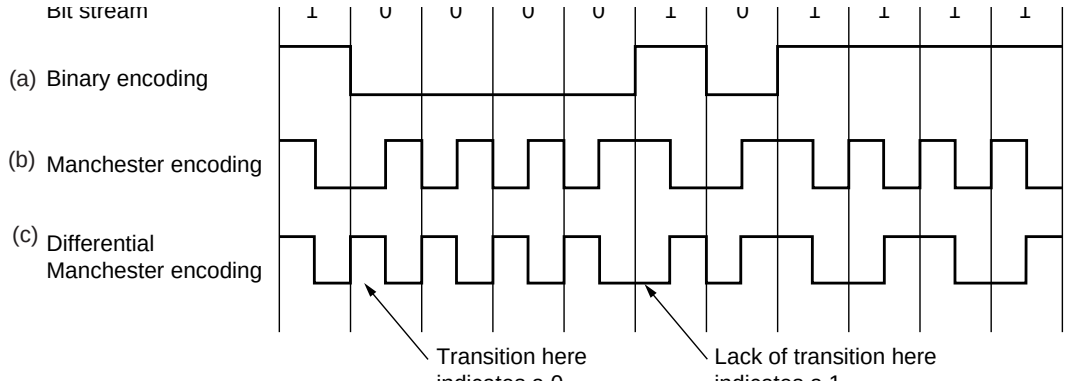
1. Aucune garantie sur le contenu
2. => si ce caractère apparait, il est *modifié* pour qu'il n'y ait pas de confusion possible
3. cette modification doit être *inversée* à la réception

Exemple : Si le caractère est `01111110`, on insère *systématiquement* un bit `0` après 5 `1` consécutifs dans le contenu. A la réception, le `0` qui succède à 5 `1` consécutifs est supprimé.

- `01111111` => `011111011` => `01111111`
- `01111101` => `011111001` => `01111101`

3.c Séquence Physique Non-Codante : Codage Manchester

Pour distinguer, un `0` d'une absence de message, on code `0` par l'*alternance* de deux tensions et `1` par l'*alternance inverse*.



Ce qui signifie que l'absence de signal indique bien l'absence de message.

4 Gestion des Erreurs

4.a Problématique

Aucun système physique ne peut être parfait et il est indispensable de prendre en compte les erreurs potentielles.

Rappel : si le signal est fiable à 99,9999999%, il se produira *une erreur par seconde* si la vitesse d'émission est de 1Gb/s.

4.b Détections/Corrections des Erreurs

Principe : redondance d'information
La redondance , c'est-à-dire la construction du code, peut se faire par
- concaténation d'une valeur de contrôle
- insertion
- transformation

4.c Codes

- *Code*
  - ***ensemble*** *de mots pouvant être émis*
  - efficacité algorithmique vs robustesse : respect d'une règle simple permettant de donner une structure robuste
- Inconvénients :
  - réduction du *débit utile quand tout va bien*
  - on ne peut détecter **toutes** les erreurs : erreurs résiduelles

4.d Détection : Bit de Parité

- Code : $Code =$ mots binaires de 8 bits dont le nombre de bits à 1 est pair.
- Fonctionnement :
  - mot à émettre :`1010111` (**7** bits)
  - émission de $M =$ `10101111`
  - réception de M'
  - si $M' \in Code$ => OK
  - sinon => erreur
  - **NB** erreur résiduelle possible si *plus d'une erreur* lors de la transmission.

4.e Correction des Erreurs

Principe : redondance d'information *supplémentaire* permettant de détecter et corriger les erreurs *sans retransmission*.
- Même avantages et inconvénients que précédemment
- **limite théorique** *Théorème de Shannon* (cf Cours 1bis)
  presqu'atteinte par les **turbocodes.**

Voir aussi sur la page web du cours.

4.f Principe de la Correction d'Erreur

- Emission du mot $M \in Code$
- Réception du mot M'
  - Si $M' \in Code$ => OK
  - Sinon corriger : trouver un mot M'' *proche* de M'
  - Erreurs résiduelles :
    - ne pas détecter l'erreur ($M \neq M' \in Code$)

– mauvaise correction ($M'' \neq M$)

4.g Comment Construire un Code Correcteur

1. Un bon code est un ensemble qui *remplit régulièrement* l'espace de tous les mots
2. *distance de Hamming* entre deux mots binaires x et y (de même taille) est le nombre de `0` à changer en `1` et inversement pour passer de x à y.
3. structure régulière avec des propriétés fines => structures mathématiques.

4.h Exemples de Codes Détecteurs et Correcteurs

Exemples :
– Code de Hamming
– Codes linéaires
– Codes polynomiaux
– Turbocodes

4.i Exemple des Codes Polynomiaux

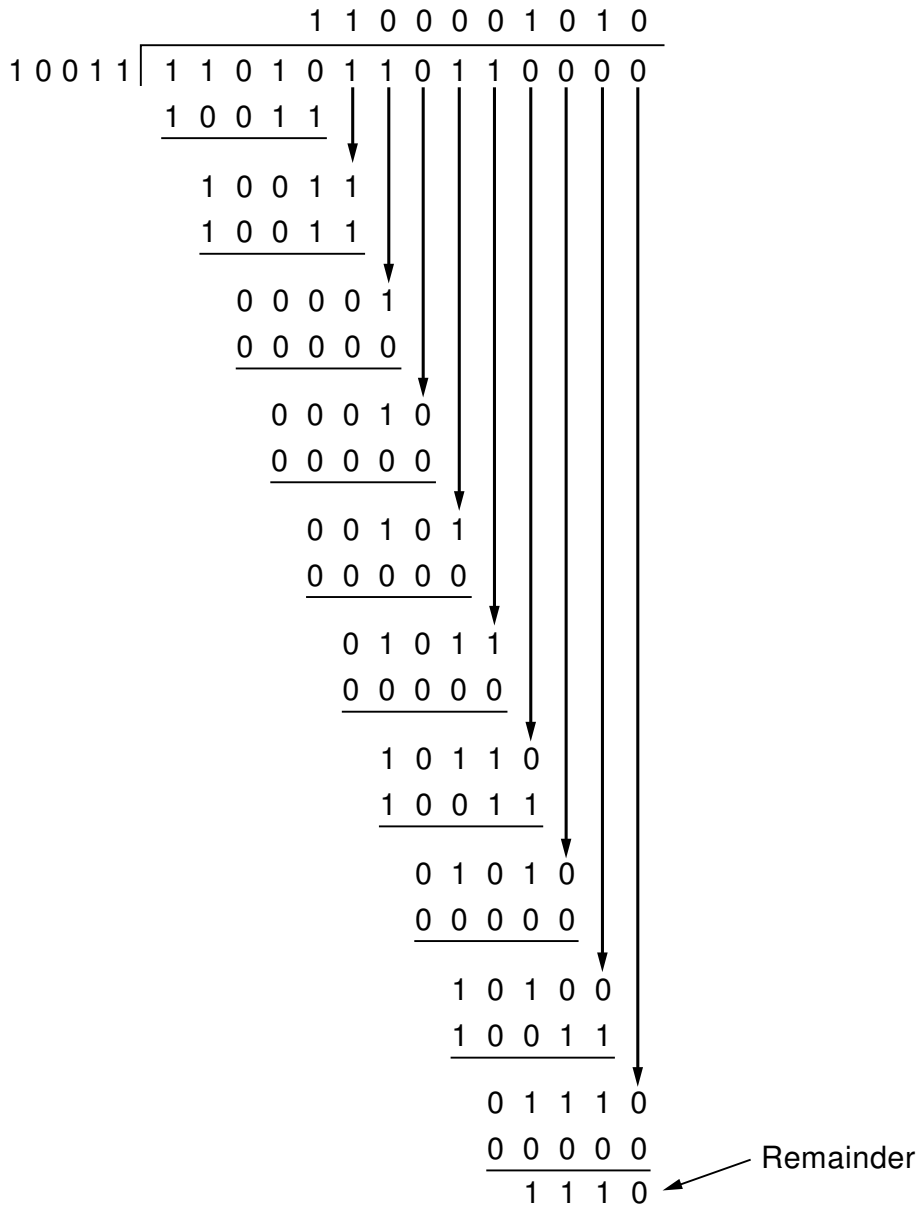
On identifie une suite de bits avec un *polynôme* $P(X)$ de $\mathbb{F}_2[X]$.

La séquence de contrôle correspond au reste de la division euclidienne de $X^k P(X)$ par un polynôme (particulièrement) bien choisi $Q(X)$ de degré k. *Ainsi les mots du codes sont exactement les multiples de* $Q(X)$.

On a $X^k P(X) = U(X) \times Q(X) + R(X)$ donc $X^k P(X) + R(X) = U(X) \times Q(X)$. On envoie la séquence binaire correspondant à $X^k P(X) + R(x)$.

– pour $Q(X) = X + 1$ on retrouve le code de parité !
– `CRC-32` : $Q(X) = X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$

Frame : 1 1 0 1 0 1 1 0 1 1
Generator: 1 0 0 1 1
Message after 4 zero bits are appended: 1 1 0 1 0 1 1 0 1 1 0 0 0 0



Transmitted frame: 1 1 0 1 0 1 1 0 1 1 1 1 1 0

4.j Pour Résumer

- Détection :
  - moins de bits de contrôle

  - *retransmission du message entier* en cas de détection d'erreurs
- Correction :
  - plus de bits de contrôle
  - pas de retransmission

Le choix s'effectue donc en fonction d'un compromis débit utile / coût retransmission

5 Gestion des Pertes

5.a Problématique

1. La trame est complètement **perdue**
2. => `ACK` accusé de réception
3. Mais si l'accusé de réception se perd ?
4. On les numérote
5. mais a-t-on assez de "numéros" ?
6. ? ... ?

5.b Protocole du Bit Alterné : Principes

1. Communication unidirectionnelle
2. "Envoyer et attendre" (... un accusé de réception)
   - `envoyer(M,seq)`
   - `declencherTemporisation()`
   - `si recevoir() == ACK(M) seq++;` *(* gerer suivant(M) *)*
     `sinon` *`envoyer(M,seq)`* (* et recommencer ...*)
3. Si le message de `ACK(M)` est perdu, on va retransmettre `M` alors qu'il a été correctement reçu
   => *duplication* => numéro de séquence `seq`
4. Combien de bits pour coder `seq` ?
5. Ce problème se pose seulement entre un message et le suivant, pas entre le prédécesseur et le suivant
   => il suffit d'avoir $seq \in \{0,1\}$, => `ACK0`, `ACK1`

5.c Protocole du Bit Alterné : Détails

Source : A. Tanenbaum Réseaux

```
/* Protocol 3 (par) allows unidirectional data flow over an unreliable channel. */

#define MAX_SEQ 1                       /* must be 1 for protocol 3 */
typedef enum  {frame_arrival, cksum_err, timeout} event_type;
#include "protocol.h"

void sender3(void)
{
  seq_nr next_frame_to_send;            /* seq number of next outgoing frame */
  frame s;                              /* scratch variable */
  packet buffer;                        /* buffer for an outbound packet */
  event_type event;

  next_frame_to_send = 0;               /* initialize outbound sequence numbers */
  from_network_layer(&buffer);          /* fetch first packet */
  while (true) {
      s.info = buffer;                  /* construct a frame for transmission */
      s.seq = next_frame_to_send;       /* insert sequence number in frame */
      to_physical_layer(&s);            /* send it on its way */
      start_timer(s.seq);               /* if answer takes too long, time out */
      wait_for_event(&event);           /* frame_arrival, cksum_err, timeout */
      if (event == frame_arrival) {
          from_physical_layer(&s);      /* get the acknowledgement */
          if (s.ack == next_frame_to_send) {
              stop_timer(s.ack);        /* turn the timer off */
              from_network_layer(&buffer);     /* get the next one to send */
              inc(next_frame_to_send);     /* invert next_frame_to_send */
          }
      }
  }
}

void receiver3(void)
{
  seq_nr frame_expected;
  frame r, s;
  event_type event;

  frame_expected = 0;
  while (true) {
      wait_for_event(&event);           /* possibilities: frame_arrival, cksum_err */
      if (event == frame_arrival) {     /* a valid frame has arrived. */
          from_physical_layer(&r);      /* go get the newly arrived frame */
          if (r.seq == frame_expected) {/* this is what we have been waiting for. */
              to_network_layer(&r.info);    /* pass the data to the network layer */
              inc(frame_expected); /* next time expect the other sequence nr */
          }
          s.ack = 1 – frame_expected;   /* tell which frame is being acked */
          to_physical_layer(&s);        /* send acknowledgement */
      }
  }
}
```

6 Contrôle de Flux

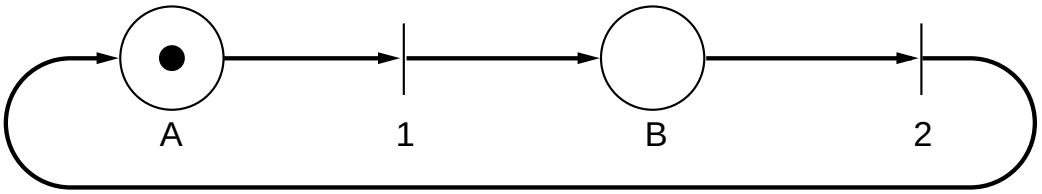
6.a Objectifs du Contrôle de Flux

Principe : *Asservir* la vitesse de l'émission aux capacités de réception.

- idée : utilisation d'une *fenêtre d'émission glissante*
  - messages et acquittements sont numérotés
  - un certain nombre de messages peuvent être émis avant attente acquittement
  - un acquittement positif de valeur k acquitte tous les messages de numérotation inférieure ou égale à k
  - sinon : `NACK` =>
    - demande de retransmission d'un message donné,
    - fenêtre de retransmission
- Inconvénients :
  - simpliste
  - risque de *duplication*
  - risque de pertes si les fenêtres ne sont pas bien calculées *dynamiquement*.

6.b Méthode de Vérification

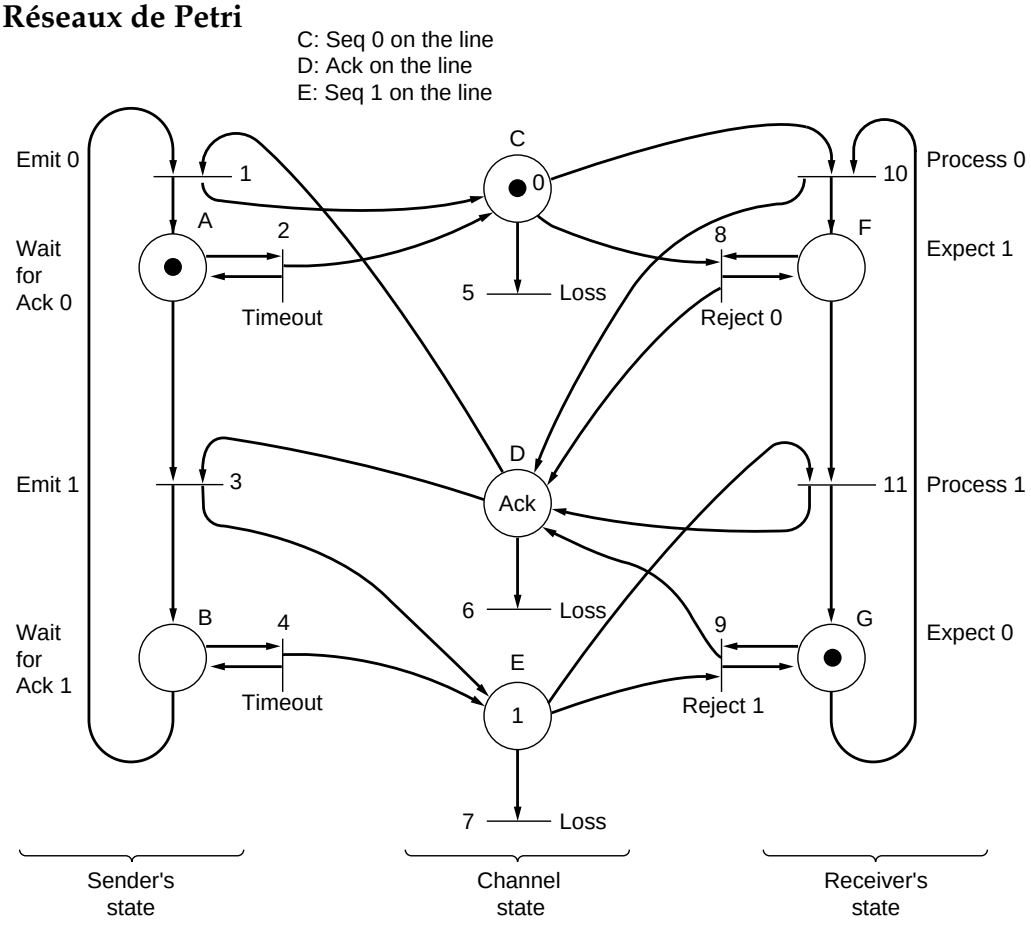
- Batterie de tests
- Preuve mathématique
- Preuve assistée
- Vérification formelle Ex : Réseaux de Petri



Un *réseau de Petri* est composé de places (les ronds) et de transitions (les traits). Des jetons (petits ronds noirs) se déplacent dans ce système en respectant la règle suivante :

- une transition T ne peut être réalisée que si chacune des places origines d'un arc entrant à la transition T contient au moins un jeton.
- ces jetons sont alors détruits
- un jeton est créé dans chacune des places cibles des arcs sortants de T

6.c Réseaux de Petri

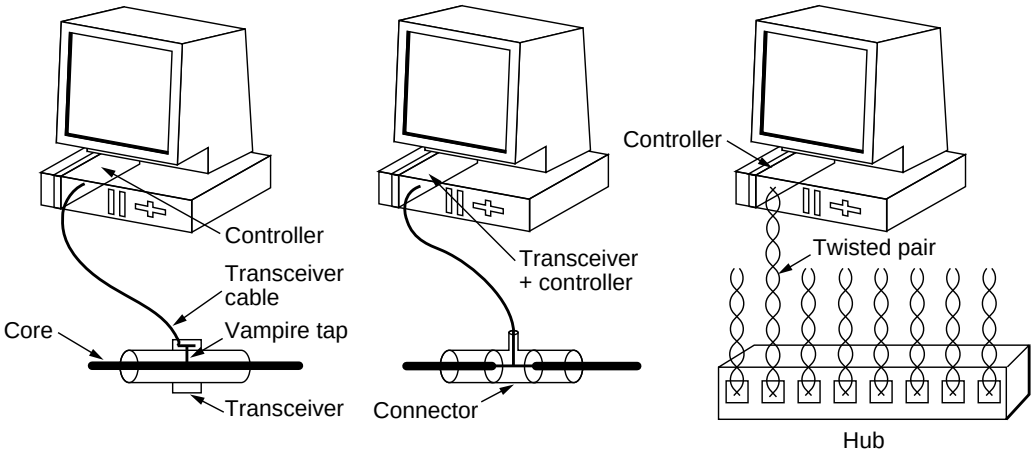


Exemple pour le protocole du bit alterné.

7 Accès au Canal de Communication

7.a Collisions Electriques

On parle d'une *collision* lorsque deux émetteurs tentent d'accéder simultanément au canal de communication.



Dans un LAN utilisant les signaux électriques sur paire torsadée, les interfaces se partage un même circuit élecrique *en parallèle*.

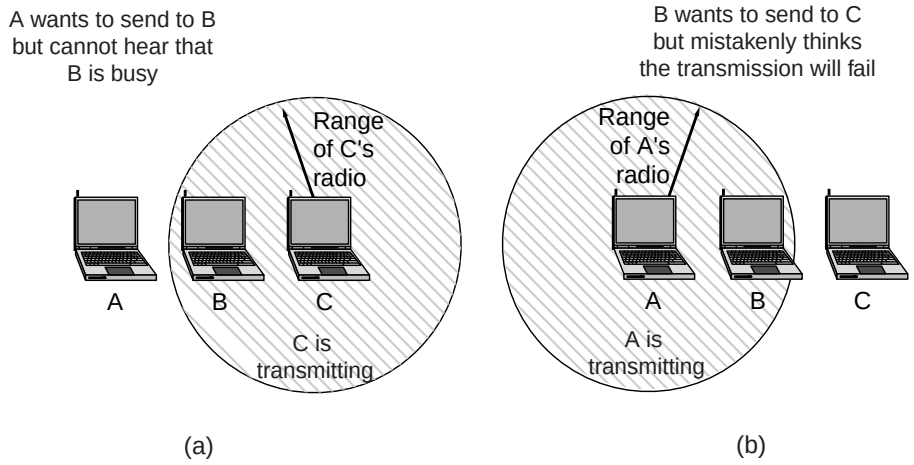
Pour transmettre un message, un émetteur va modifier la tension et les récepteurs vont détecter le changement de tension et l'interpréter.

Si deux stations émettent en même temps, le signal électrique devient une composition des deux signaux émis et devient **indécodable**.

7.b Accès Aléatoire avec Ecoute

- CSMA : Carrier Sense Multiple Access
  - écouter le canal avant d'émettre
  - si occupé, différer l'émission
  - problème : il peut subsister des collisions en cours d'émission
- CSMA avec détection de collision : CSMA/CD
  - à l'écoute préalable on ajoute l'écoute pendant la transmission
  - Réémission au bout d'un temps aléatoire
  - Utilisé par Ethernet, normalisation ISO 802.3
- Algorithme :
  - les stations écoutent le canal
  - si le canal est libre, elles commencent à émettre
  - Quand une collison est détectée : envoi de signaux spéciaux appelés bits de bourrage (jam32)
  - Réémission après un temps aléatoire
- La tranche canal :
  - Durée s'écoulant entre l'instant d'émission des premiers bits et le moment où l'émetteur est sûr que son message est complètement transmis
  - $T_c = 2 \times$ délai de propagation
- Round trip delay :
  - temps de détection de la collision
  - tps aller + tps retour + tps jam

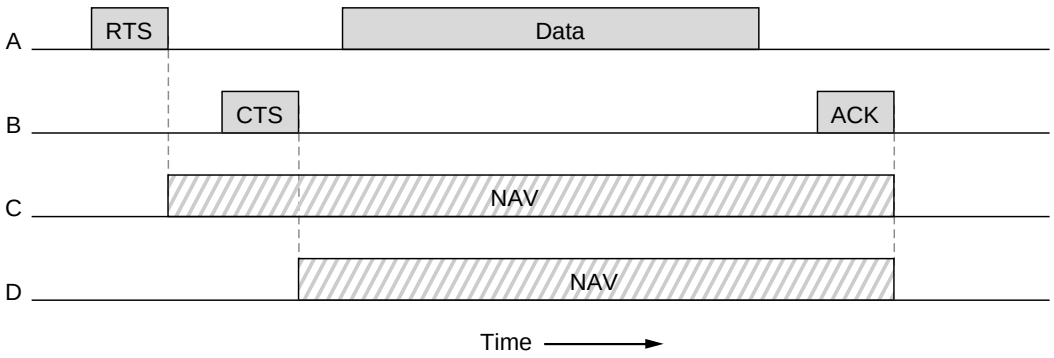
7.c Collisions Electromagnétiques



Contrairement au cas précédent, il n'est pas possible de détecter toutes les collisions à l'émetteur. De plus, l'émission est en général incompatible avec la réception, c'est-à-dire qu'il n'est pas possible de détecter les collisions.

7.d Esquive des Collisions (CSMA/CA)

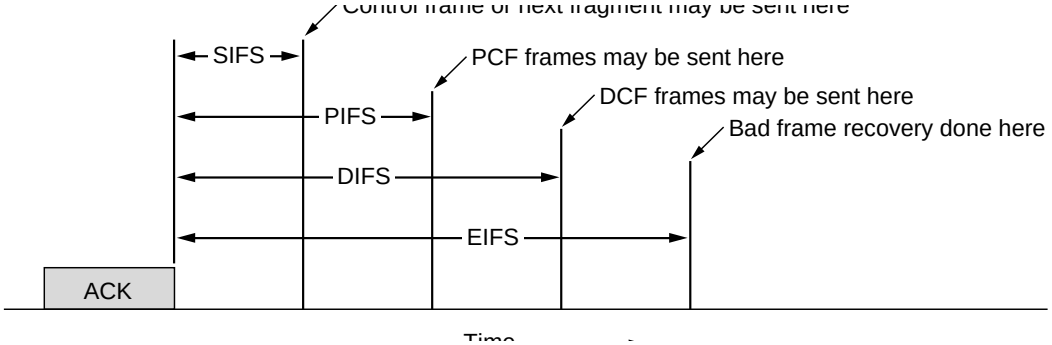
Dans le cas d'une communication sans-fil, on utilise un protocole permettant de prévenir les collisions. Cela revient à demander et d'obtenir *explicitement* la parole avant d'émettre.



- RTS (Ready To Send) : déclaration d'intention de l'émetteur
- CTS (Clear to Send) : le récepteur est disponible
- NAV : non disponible pour émettre

7.e Attente Variable

Le temps d'attente après occupation du canal est variable suivant le mode utilisé :
- DIFS : **Distributed** Inter Frame Space
- PIFS : **Point** Inter Frame Space



Cela permet une meilleure gestion d'utilisation de la capacité du médium, même avec des collisions.

8 Protocoles de Liaison de Données

8.a Ethernet

- famille de protocoles compatibles définis par IEEE
- transmission de paquets de *taille variable* dans des réseaux filaires et non filaires

Rappel : *IEEE* : Institute of Electrical and Electronics Engineers est une association professionnelle constituée d'ingénieurs électriciens, d'informaticiens, de professionnels du domaine des télécommunications, etc.

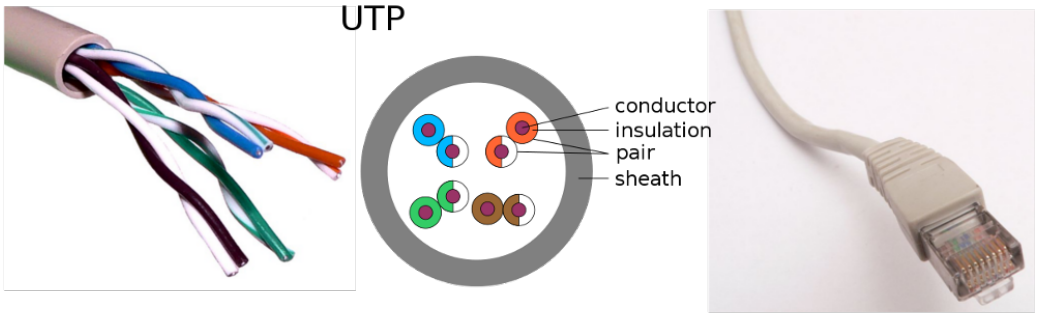
8.b Ethernet filaire

- Norme 802.3 : réseau local bande de base avec méthode d'accès CSMA/CD (détaillé précédemment)
- variantes :

	Câble	Longueur	Nbre stations
10base2	coaxial fin	200m	30
100baseT	paire torsadée	100m	1024
100baseFX	fibre optique	2000m	1024

8.c Ethernet : Caractéristiques

- Débit Nominal : 10/100Mbits/s
- Transmission en bande de base avec codage Manchester ($\pm 2,5V$ en 10BASE-T)
- sur câble *catégorie 5* : deux paires utilisées



8.d Ethernet Gigabit

fibre optique – 1000BASE-LX en mode single
  – 5 km
paire torsadée – 1000BASE-T
  – 4 paires utilisées sur un câble catégorie 5 et supérieure
  – 100 m
vers l'infini et au-delà norme IEEE 802.3ba : 40 Gb/s et *100 Gb/s* normalisé en 2010.

8.e Ethernet : niveau MAC

Rappel :
- Fonctions sous-niveau MAC
  - mise en trame
  - adressage
  - détection erreur
  - réaction aux signaux d'occupations du canal/collisions
- Format de la trame :

7	1	2 ou 6	2 ou 6	2	46-1500	4
préamb.	dél.	Adresse	Adresse	type ou	Données + remplissage	CRC
		dest.	source	longueur		

8.f Ethernet : les Champs

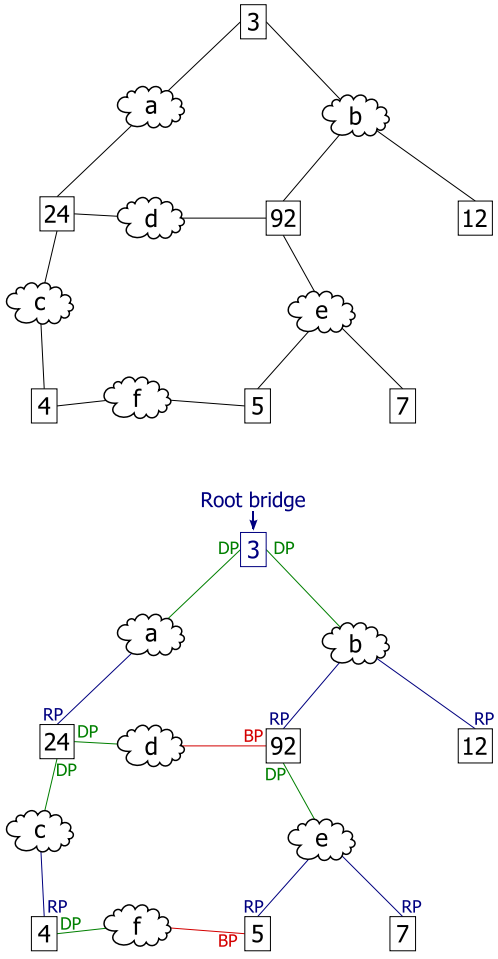
- Préambule : 7×`10101010` => synchro bit
- Délimiteur : 1×`10101011` => synchro octet/trame
- Adresse destination : sur 6 octets en général, si tous les bits sont à 1 => diffusion
- Longueur : au minimum 64 octets
- Bourrage : si la longueur des données est insuffisante
- Contrôle : CRC-32

8.g Adresses Ethernet

- Adresses uniques sur 48 bits (attribuées à la fabrication)
- 3 types d'adresse reconnue par le coupleur
  - adresse physique d'un coupleur
    - 24 bits fabricant (OUI)
    - 24 bits n° de série
  - diffusion générale (broadcast)
    - `FF:FF:FF:FF:FF:FF`
  - diffusion multidestinataires (multicast) (bit de point faible du premier octet à 1)
    - `01:80:C2:00:00:00`

8.h Arbre Couvrant Ethernet

Pour des raisons d'efficacité, on organise un segment en arbre grâce à des *commutateurs* qui filtre le traffic.



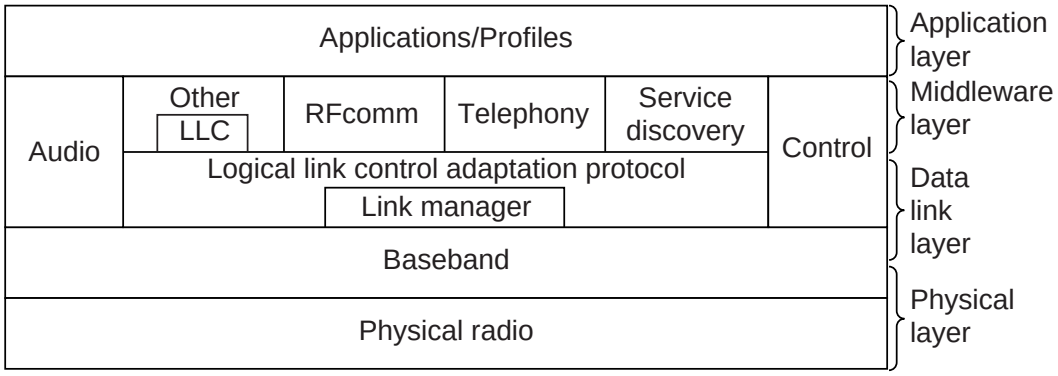
RP port racine – **DP** port désigné – **BP** port bloqué

8.i Récapitulatif : Ethernet Filaire

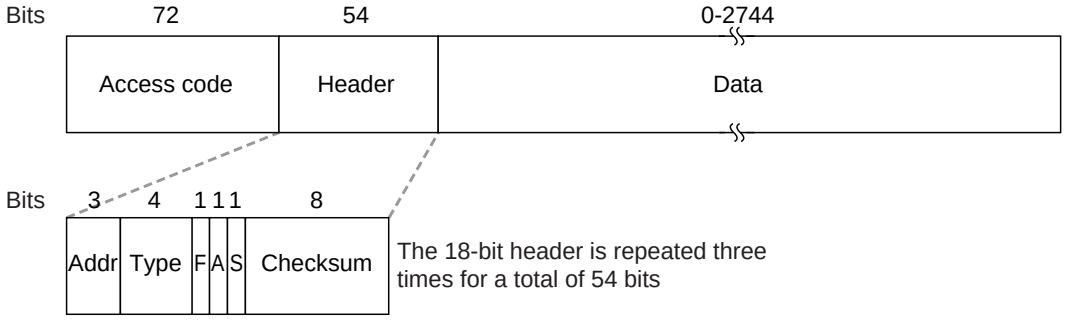
Processus Ponts, Commutateurs, Stations
Communication message
Nommage *adresse MAC*
Synchronisation protocoles de gestion du flux
Cache et Réplication N/A
Tolérance aux Défaillances Codes détecteurs et correcteurs - Protocoles de gestions des omissions
Sécurité aucune (physique)

8.j Transmissions sans fil

Pragmatisme vs modèle OSI :

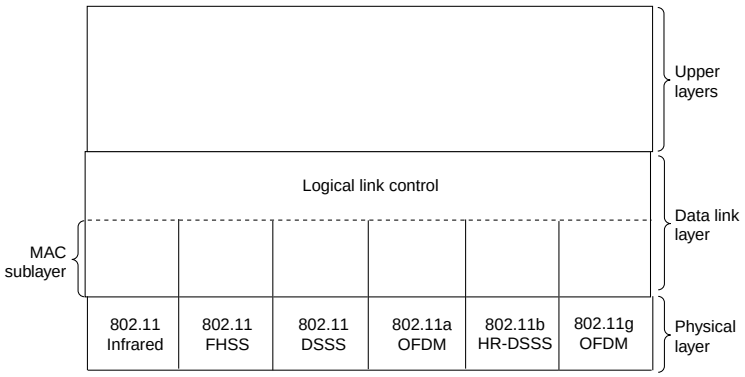


8.k Bluetooth

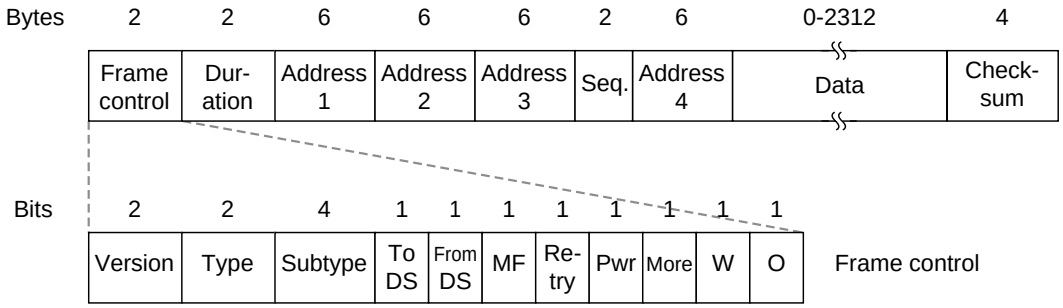


8.l 802.11 : communication sans fil

Sous-couche MAC



Trame



8.m Déclinaisons du 802.11

La norme principale est déclinée en améliorations :

802.11a	wifi 5GHz	haut débit 30Mbits/s
802.11b	wifi	débit 11Mbits/s, large base installée
802.11d	i18n	gestion de l'*allocation légale* des fréquences
802.11e	QoS	gestion de la qualié de service
802.11f	itinérance	utiliser plusieurs point d'accès successivement
802.11g	wifi	débit 54Mbits/s, majoritaire
802.11h	Europe	standard européen (hiperLAN 2)
802.11i	sécurité	gestion cryptographique complète
802.11j	Japon	standard japonais
802.11n	WWiSE	très haut débit : 300Mbits/s

Sans parler des "améliorations" propriétaires...

8.n Modes

Un réseau sans fil (WLAN) peut fonctionner
- en mode décentralisé : *ad hoc*
- en mode infrastructure : *points d'accès*

8.o Récapitulatif : Ethernet Sans-fil

˜Idem Ethernet filaire +

Processus répéteurs WDS

Sécurité L'interception passive étant *très facile*, il faut rajouter une couche de sécurité :

- WEP : cassé (2001)
- => protocoles plus sûrs (?)
- WPA/WPA2 *cassé partiellement (2008-2010)*
  =>cf Cours ultérieurs et *option Cryptographie*

9 Crédits

- Figures A. Tanenbaum. Libre d'utilisation pour l'enseignement
- Wikimedia CC-BY-SA